

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/274734142>

Seri Belajar ASP.NET : Membangun Aplikasi Web Mudah dan Cepat

Book · November 2008

DOI: 10.13140/RG.2.1.3832.2964

CITATION

1

READS

29,174

1 author:



M Reza Faisal

Universitas Lambung Mangkurat

59 PUBLICATIONS 133 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



IT Asset Management [View project](#)



DNA Sequence Classification [View project](#)



INDONESIA
.NET DEVELOPER COMMUNITY



Seri Belajar
ASP.NET
Membangun Aplikasi Web Mudah & Cepat
M Reza Faisal, Microsoft MVP ASP.NET/IIS

Daftar Isi

DAFTAR ISI	I
1 PENDAHULUAN	1-1
KATA PENGANTAR	1-1
GARIS BESAR	1-1
2 .NET 3.5 & VISUAL STUDIO 2008	2-3
.NET FRAMEWORK 3.5	2-3
<i>.NET Framework sebagai platform</i>	2-3
<i>.NET 3.0 sebagai jalan menuju Longhorn</i>	2-4
<i>.NET di masa depan</i>	2-5
VISUAL STUDIO 2008	2-5
<i>Membuat Solution</i>	2-6
<i>Menambah Project pada Solution</i>	2-9
<i>Toolbox</i>	2-13
<i>Editor</i>	2-17
<i>Build, Execute & Debug</i>	2-21
<i>Menambah Reference pada Project</i>	2-21
3 PENGENALAN C#	3-24
HELLO WORLD WITH C#	3-24
VARIABEL.....	3-26
<i>Aturan Penamaan Variabel</i>	3-27
<i>Kategori Variabel</i>	3-27
EKSPRESI.....	3-29
PERCABANGAN.....	3-30
<i>if dan if-else</i>	3-30
<i>switch</i>	3-30
PENGULANGAN	3-31
<i>while</i>	3-31
<i>for</i>	3-31
<i>foreach</i>	3-31
<i>do-while</i>	3-31
JUMP STATEMENT	3-32
PENANGANAN ERROR.....	3-33
<i>Statement try</i>	3-33
PEMROGRAMAN OBJECT-ORIENTED.....	3-34
<i>Class dan Objek</i>	3-34
<i>Method dan Field</i>	3-36
<i>Static Member</i>	3-39
<i>Inheritance dan Polymorphism</i>	3-40
<i>Method Overloading</i>	3-42
COLLECTION	3-43
<i>Array</i>	3-43
<i>Mendeklarasikan Array</i>	3-43
<i>Inisialisasi Array</i>	3-43
<i>Collection</i>	3-44
<i>ArrayList</i>	3-45
<i>Hashtable</i>	3-45
REFERENSI.....	3-46
4 ASP.NET	4-47

WEB SITE DAN WEB APPLICATION	4-47
STRUKTUR WEB SITE.....	4-49
<i>File Konfigurasi</i>	4-49
Web.config.....	4-49
<i>Tipe File ASP.NET</i>	4-50
<i>Folder ASP.NET</i>	4-50
HELLO WORLD WITH ASP.NET.....	4-51
STANDARD SERVER CONTROL.....	4-58
<i>Label</i>	4-58
<i>TextBox</i>	4-60
<i>Button</i>	4-61
<i>HyperLink</i>	4-64
<i>DropDownList</i>	4-64
<i>ListBox</i>	4-66
<i>CheckBox</i>	4-67
<i>CheckBoxList</i>	4-68
<i>RadioButton</i>	4-69
<i>RadioButtonList</i>	4-71
<i>Image</i>	4-72
<i>Calendar</i>	4-73
<i>Panel</i>	4-79
<i>Multiview dan View</i>	4-82
<i>Wizard</i>	4-85
<i>FileUpload</i>	4-87
<i>PlaceHolder</i>	4-88
5 ASP.NET DAN DATABASE.....	5-92
ADO.NET	5-92
<i>Koneksi</i>	5-92
<i>Operasi Data</i>	5-97
Query tanpa mengembalikan result set.....	5-97
Query dengan pengembalian single value.....	5-99
Query dengan pengembalian berupa result set	5-100
LINQ.....	5-106
<i>C# 3.0</i>	5-106
<i>LINQ</i>	5-112
Sorting	5-113
Filter.....	5-114
<i>Standar Query Operator</i>	5-114
Select	5-114
Order	5-115
Filter.....	5-115
LINQ TO SQL.....	5-116
<i>A Glance with LINQ to SQL</i>	5-118
Menampilkan Data	5-121
Menambah Data	5-122
Mengupdate Data	5-123
Menghapus data	5-123
DATA BINDING PADA ASP.NET	5-124
<i>Standard Control</i>	5-124
<i>DropDownList</i>	5-124
<i>RadioButtonList</i>	5-126
<i>CheckBoxList</i>	5-128
<i>Data Control</i>	5-130
Cara Otomatis.....	5-131
Repeater	5-136
DataList	5-138
ListView.....	5-139

GridView	5-143
6 DESIGN ANTARMUKA.....	6-147
MASTER PAGE	6-147
<i>Membuat Master Page.....</i>	6-147
<i>Menggunakan Master Page</i>	6-149
<i>Mengakses Server Control Pada Master Page.....</i>	6-150
<i>Mengakses Server Control Pada File ASPX dari Master Page.....</i>	6-151
<i>Nesting Master Page</i>	6-152
THEME DAN SKIN.....	6-154
<i>Implementasi Theme</i>	6-154
<i>Implementasi Skin.....</i>	6-156
WEB USER CONTROL	6-157
<i>Membuat dan Menggunakan Web User Control</i>	6-158
7 RESOURCE & GLOBALIZATION.....	7-162
GLOBAL RESOURCE	7-162
LOCAL RESOURCE	7-165
8 VALIDASI	8-167
REQUIREDFIELDVALIDATOR	8-169
RANGEVALIDATOR.....	8-169
REQULAREXPRESSIONVALIDATOR	8-170
COMPAREVALIDATOR.....	8-170
CUSTOMVALIDATOR	8-172
<i>Validasi Client-Side</i>	8-172
<i>Validasi Server-Side</i>	8-173
9 ASP.NET AJAX DAN AJAX CONTROL TOOLKIT	9-175
ASP.NET AJAX.....	9-175
<i>Implementasi ASP.NET Ajax.....</i>	9-175
<i>Pengenal UpdateMode pada UpdatePanel</i>	9-176
<i>Trigger pada UpdatePanel.....</i>	9-178
<i>UpdateProgress</i>	9-180
<i>Timer.....</i>	9-181
AJAX CONTROL TOOLKIT	9-182
<i>Mendaftarkan Library Ajax Control Toolkit pada Web.config.....</i>	9-183
<i>Control Penggunaan.....</i>	9-183
<i>Calendar.....</i>	9-183
<i>FilteredTextBox.....</i>	9-184
10 MEMBERSHIP, PROFILE DAN NAVIGASI.....	10-185
MEMBERSHIP	10-185
<i>Menyiapkan Database untuk Membership</i>	10-185
<i>Menyiapkan Web.config.....</i>	10-188
<i>Mengelola Role</i>	10-190
<i>Mengelola User</i>	10-190
<i>Mengelola Access Rule</i>	10-191
<i>Login Control</i>	10-192
<i>Authentifikasi User</i>	10-192
<i>Mengamankan Halaman Member</i>	10-195
<i>Mengatur Akses Control Pada Halaman</i>	10-196
PROFILE.....	10-199
<i>Menyiapkan Web.config.....</i>	10-199
<i>Operasi Profile</i>	10-200
NAVIGASI.....	10-202
<i>Sitemap.....</i>	10-202

<i>Menu dan TreeView</i>	10-204
Menu	10-204
TreeView.....	10-204
11 REPORTING	11-206
MICROSOFT REPORTVIEWER	11-206
<i>Report Data Source</i>	11-214
<i>List</i>	11-222
<i>Tables</i>	11-230
<i>Chart</i>	11-237
<i>Matrix (Cross Table)</i>	11-242

1

Pendahuluan

Kata Pengantar

Terima kasih, Anda telah mengunduh dan mulai membaca ebook ini. Pada edisi pertama ebook ini telah dibahas tentang fitur-fitur yang dimiliki oleh ASP.NET 3.5 dan Visual Studio 2008. Pada edisi kedua ini, selain terjadi perubahan pada cover juga ditambahkan pembahasan fitur-fitur yang terdapat pada ASP.NET 4.0 dan penggunaan Visual Studio 2010 sebagai tool pengembang.

Ebook ini diharapkan dapat digunakan oleh seluruh level web developer yang ingin mempelajari pembuatan aplikasi web dengan ASP.NET. Pada ebook ini pembahasan dilakukan dari pengenalan tool Visual Studio 2010, pengenalan bahasa C# untuk pemrograman dan pengenalan kontrol-kontrol pada ASP.NET. Selain itu ditambah beberapa penggunaan template project yang dapat membantu produktifitas dalam melakukan pembangunan aplikasi web dengan ASP.NET.

Buku digital ini ditujukan untuk para web developer yang telah terbiasa dengan HTML, sudah mengerti konsep-konsep OOP dan juga sudah pernah melakukan pembangunan aplikasi dengan teknologi server side seperti PHP atau Java, sehingga pada buku ini tidak akan membahas konsep-konsep terlalu dalam dan hanya memaparkan praktek pembangunan aplikasi web dengan ASP.NET.

Bagi pembaca yang masih pemula dunia pemrograman terutama pemrograman aplikasi web maka ada baiknya sembari membaca buku digital ini juga membaca buku-buku yang menjelaskan lebih detail tentang HTML, CSS, Javascript dan juga konsep-konsep algoritma dan pemrograman berorientasi objek.

Garis Besar

Secara garis besar buku digital ini akan memaparkan tentang :

1. Pengenalan Visual Studio 2010 yang meliputi cara-cara pembuatan solution, project, pengenalan fasilitas yang telah tersedia.
2. Paparan singkat bahasa pemrograman C#.
3. Pengenalan struktur project website atau web application, yang meliputi penjelasan fungsi direktori dan file yang ada pada project website.
4. Pembuatan website atau web application dan pemanfaatan server control yang telah tersedia pada ASP.NET dan fasilitas lain yang membantu dalam pembangunan aplikasi web, seperti :
 - a. Pengenalan penggunaan server control yang terdapat pada Standard Toolbox.
 - b. Design antarmuka website dengan memanfaatkan MasterPage dan Theme.

- c. Membership, Profile dan penggunaan Login Toolbox. Pembuatan proses authentifikasi suatu halaman yang hanya boleh diakses oleh user yang terdaftar dan juga contoh kasus pembuatan role dan user management.
 - d. Navigasi dan menu dengan memanfaatkan Sitemap dan server control yang ada pada Navigation Toolbox. Pembuatan menu disini ada kaitannya dengan role yang ada pada Membership.
 - e. Validation dengan memanfaatkan control pada Navigation Toolbox.
 - f. Personalization dan Webpart dengan memanfaatkan Webpart Toolbox.
5. ASP.NET dan database, pada bagian ini akan diterangkan bagaimana melakukan koneksi ke database server dan melakukan operasi database dengan ADO.NET, LINQ, menampilkan data dengan memanfaatkan control yang ada pada Standard Toolbox dan Data Toolbox.
 6. ASP.NET Ajax dan Ajax Control Toolkit.
 7. Reporting dengan Report Viewer.

2

.NET 4.0 & Visual Studio 2010

.NET Framework 4.0

Microsoft .NET Framework (dibaca Microsoft Dot Net Framework) adalah sebuah komponen yang dapat ditambahkan ke sistem operasi Microsoft Windows atau telah terintegrasi ke dalam Windows (mulai dari Windows Server 2003 dan versi-versi Windows terbaru). Kerangka kerja ini menyediakan sejumlah besar solusi-solusi program untuk memenuhi kebutuhan-kebutuhan umum suatu program baru, dan mengatur eksekusi program-program yang ditulis secara khusus untuk framework ini. .NET Framework adalah kunci penawaran utama dari Microsoft, dan dimaksudkan untuk digunakan oleh sebagian besar aplikasi-aplikasi baru yang dibuat untuk platform Windows.

Pada dasarnya, .NET Framework memiliki 2 komponen utama: CLR dan .NET Framework Class Library.

Program - program yang ditulis untuk .NET Framework dijalankan pada suatu lingkungan software yang mengatur persyaratan-persyaratan runtime program. Runtime environment ini, yang juga merupakan suatu bagian dari .NET Framework, dikenal sebagai Common Language Runtime (CLR). CLR menyediakan penampilan dari application virtual machine, sehingga para programmer tidak perlu mengetahui kemampuan CPU tertentu yang akan menjalankan program. CLR juga menyediakan layanan-layanan penting lainnya seperti jaminan keamanan, pengaturan memori, garbage collection dan exception handling / penanganan kesalahan pada saat runtime. Class library dan CLR ini merupakan komponen inti dari .NET Framework. Kerangka kerja itu pun dibuat sedemikian rupa agar para programmer dapat mengembangkan program komputer dengan jauh lebih mudah, dan juga untuk mengurangi kerawanan aplikasi dan juga komputer dari beberapa ancaman keamanan.

CLR adalah turunan dari CLI (Common Language Infrastructure) yang saat ini merupakan standar ECMA.

Solusi-solusi program pembentuk class library dari .NET Framework mengcover area yang luas dari kebutuhan program pada bidang user interface, pengaksesan data, koneksi basis data, kriptografi, pembuatan aplikasi berbasis web, algoritma numerik, dan komunikasi jaringan. Fungsi-fungsi yang ada dalam class library dapat digabungkan oleh programmer dengan kodennya sendiri untuk membuat suatu program aplikasi baru.

Pada berbagai literatur dan referensi di Internet, .NET Framework seringkali disingkat menjadi .NET saja.

.NET Framework sebagai platform

.NET seringkali juga dapat diartikan sebagai platform, yang merupakan suatu lingkungan terpadu untuk pengembangan dan eksekusi untuk berbagai macam bahasa pemrograman dan kumpulan library untuk bekerja sama membuat dan menjalankan aplikasi berbasis

Windows yang lebih mudah untuk dibuat, diatur, didistribusikan, dan diintegrasikan dengan sistem jaringan lain.

Dalam perkembangannya, .NET seringkali dikaitkan pula dengan versi Visual Studio yang sesuai dengan dukungan versi yang bersangkutan untuk pengembangan aplikasi. Berikut ini versi .NET dan versi Visual Studio yang terkait:

1. .NET 1.0 dan Visual Studio .NET (atau seringkali disebut juga dengan Visual Studio .NET 2002).
2. .NET 1.1 dan Visual Studio .NET 2003.
3. .NET 2.0 dan Visual Studio 2005.
4. .NET 3.0 dan Visual Studio 2005 dengan tambahan addin untuk WPF, WCF dan WF.
5. .NET 3.5 dan Visual Studio 2008.
6. .NET 4.0 dan Visual Studio 2010

.NET 2.0, 3.0 dan 3.5 memiliki CLR yang sama. Dengan demikian, struktur IL juga sama. Adapun fasilitas penambahan kata kunci pemrograman seperti pada LINQ yang sebenarnya lebih mengarah sebagai fitur bahasa pemrograman (programming language feature) sehingga bukan merupakan fitur CLR. Sedangkan pada .NET 4.0 terjadi perubahan pada CLR dan Base Class Library (BCL) serta pengembangan di beberapa bagian seperti pengembangan untuk parallel computing, improvement pada file IO dan masih banyak lagi.

.NET 3.0 sebagai jalan menuju Longhorn

.NET 3.0 pada hakikatnya adalah .NET 2.0 ditambah pilar-pilar teknologi untuk Windows Longhorn Client/Server atau sekarang bernama Windows Vista untuk Longhorn Client dan Windows Server 2008 untuk Longhorn Server. Dengan demikian, program yang dibuat dengan .NET 2.0 dapat berjalan pada .NET 3.0, kecuali untuk fasilitas atau fitur yang sangat spesifik.

Pilar-pilar teknologi pada .NET 3.0:

1. Windows Presentation Foundation atau WPF.

WPF adalah subsystem pada .NET Framework 3.0 yang berfungsi menangani :

- a. Grafik yang bertipe vector atau raster.
- b. Audio dan video.
- c. Menggambar objek 2D dan 3D.
- d. Animasi.
- e. Databinding.

WPF memisahkan antara antarmuka (User Interface/UI) dengan logika bisnis. Antarmuka pada WPF menggunakan file dengan format eXtensible Application Markup Language (XAML, baca Zammel) yang mengikuti format penulisan XML.

WPF dapat digunakan untuk membuat aplikasi desktop dan juga web. Dalam perkembangannya dikenalkan teknologi dengan nama Silverlight yang merupakan pengembangan WPF untuk aplikasi web.

2. Windows Communication Foundation atau WCF.

WCF merupakan framework untuk membuat aplikasi berdasarkan service-oriented messaging system yang memungkinkan program untuk berkomunikasi baik secara lokal atau remote seperti web service.

3. Windows Workflow Foundation atau WF

4. CardSpace

Walaupun teknologi di atas ditujukan pada Windows Vista dan sesudahnya, .NET 3.0 dapat pula diinstall pada Windows XP SP2 dan Windows Server 2003.

Versi terbaru dari .NET adalah .NET 3.5. Versi ini didukung sepenuhnya oleh Visual Studio 2008, berisi teknologi-teknologi kunci yaitu:

1. LINQ, Language Integrated Query adalah komponen pada .NET Framework 3.5 yang memungkinkan menambahkan kemampuan query data seperti perintah SQL pada bahasa yang digunakan pada .NET Framework (seperti pada C# 3.0 atau VB.NET 9.0).

Contohnya bisa dilihat pada kode di bawah ini :

```
int SomeValue = 5;

var results = from c in SomeCollection
              let x = SomeValue * 2
              where c.SomeProperty < x
              select new {c.SomeProperty, c.OtherProperty};

foreach (var result in results)
    Console.WriteLine(result);
```

2. Extension Methods.
3. Lambda Expression.

Dan juga perbaikan serta pengembangan dari WPF, WCF, dan WF.

.NET 4.0

Microsoft tengah mengembangkan .NET "4". .NET 4 ini akan lebih mengedepankan konsep SOA (Service Oriented Architecture) dan SaaS (Software As A Service). Dengan dukungan library dasar yang sudah ada seperti XML dan WCF, Microsoft berencana untuk menambah dukungan web service seperti REST, JSON, POX. Di samping itu Microsoft juga telah mempersiapkan pengembangan "Oslo" sebagai bagian dari inisiatif pengembangan platform .NET.

Sumber :

1. http://id.wikipedia.org/wiki/.NET_Framework
2. http://en.wikipedia.org/wiki/Windows_Presentation_Foundation
3. http://en.wikipedia.org/wiki/Windows_Communication_Foundation
4. http://en.wikipedia.org/wiki/Windows_Workflow_Foundation
5. http://en.wikipedia.org/wiki/Windows_CardSpace
6. http://en.wikipedia.org/wiki/Language_Integrated_Query
7. http://en.wikipedia.org/wiki/Extension_method

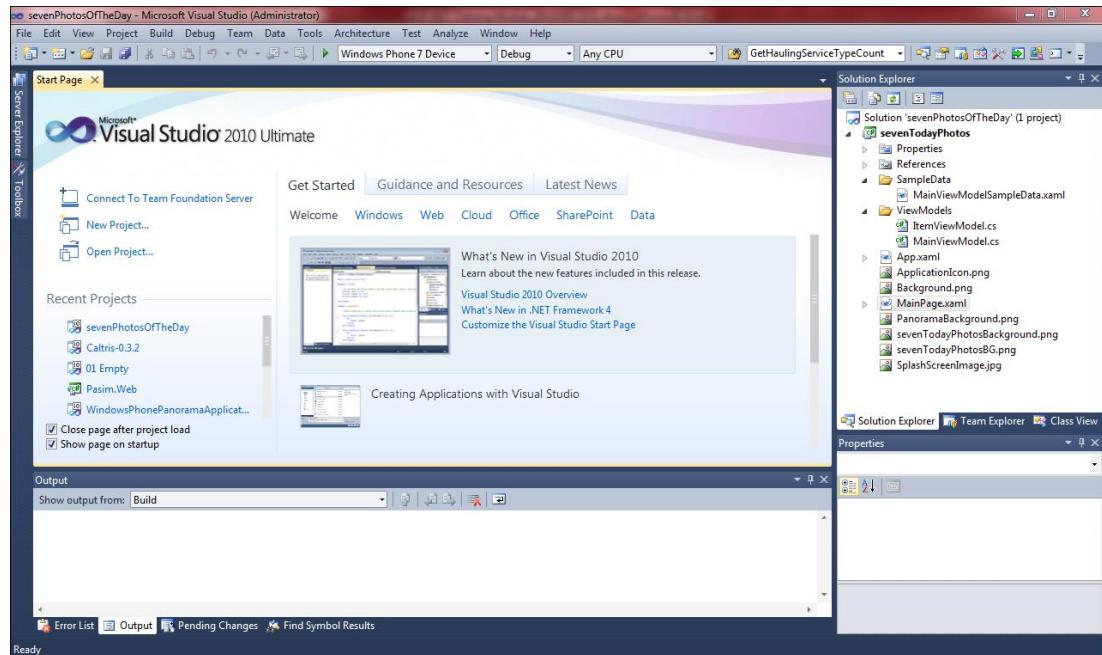
Visual Studio 2010

Visual Studio adalah Integrated Development Environment (IDE) dari untuk membangun aplikasi console dan Graphical user interface (GUI) dengan menggunakan bahasa yang didukung pada .NET Framework. Aplikasi GUI yang dapat dibangun diantaranya adalah Windows Form, Website, Web Application, Windows Mobile.

Visual Studio selain mempunyai feature untuk :

1. Designer antarmuka untuk Winform, WPF dan Web. Selain itu juga dapat digunakan untuk mendesign Class, Data dan Mapping.
2. Code editor dengan dukungan IntelliSense.
3. Debugger.

Visual Studio 2008 adalah IDE dengan multi target .NET Framework, artinya developer dapat membangun aplikasi dengan menggunakan .NET Framework 2.0, 3.0, 3.5 dan 4.0.



Selanjutnya akan dijelaskan penggunaan Visual Studio 2010 untuk keperluan pembangunan aplikasi web. Jadi tidak akan semua feature Visual Studio 2010 yang akan dibahas tetapi hanya sebagian kecil saja yang berhubungan dengan pembangunan aplikasi web.

Membuat Solution

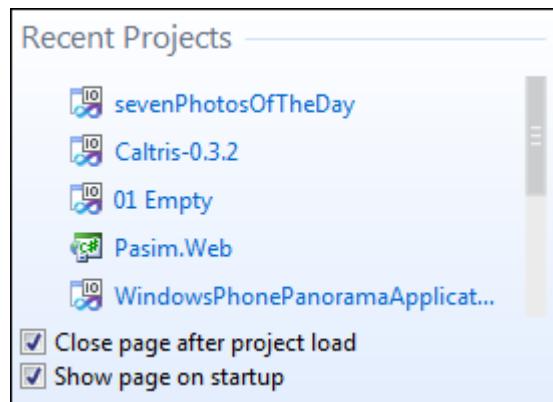
Membuat solution adalah langkah pertama yang dilakukan untuk membangun aplikasi. Dalam satu solution dapat lebih dari satu project. Project yang terdapat pada satu solution dapat terdiri atas bermacam tipe aplikasi dan platform. Artinya dalam satu solution bisa terdapat :

1. Windows.
2. Web.
3. Smart Device.
4. Office.
5. Dan lain-lain.

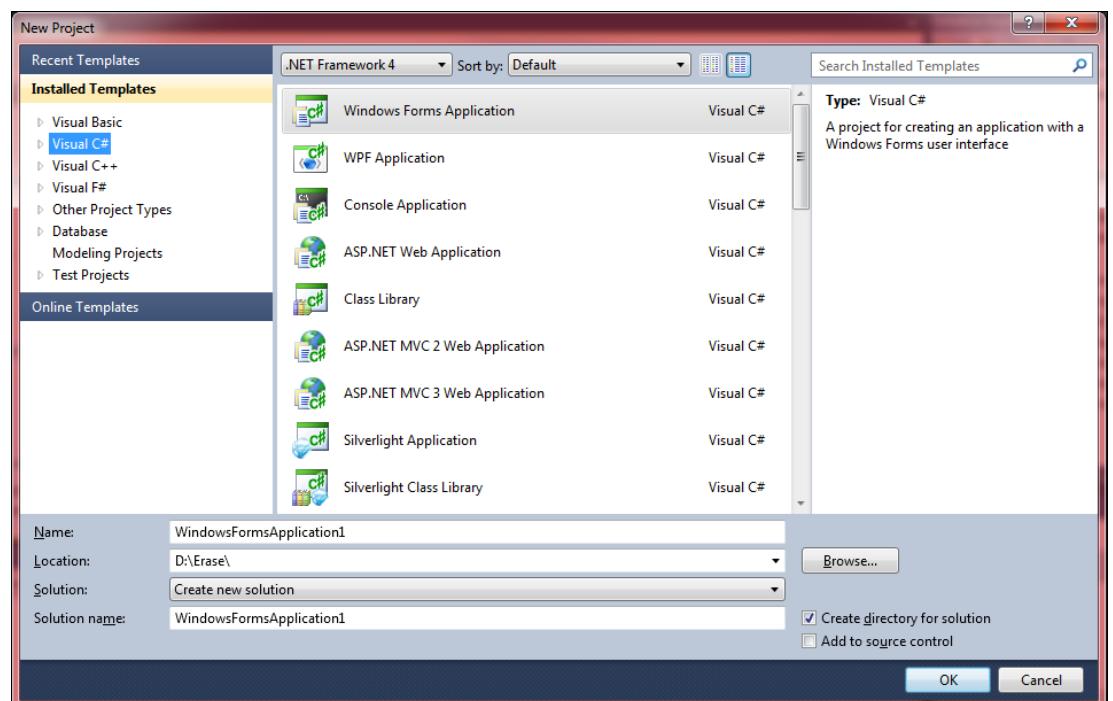
Dengan membuat Solution terlebih dahulu maka developer dapat membuat nama yang dapat mewakili seluruh project yang ada di dalam Solution tersebut.

Berikut ini adalah langkah yang mesti diikuti untuk membuat Solution.

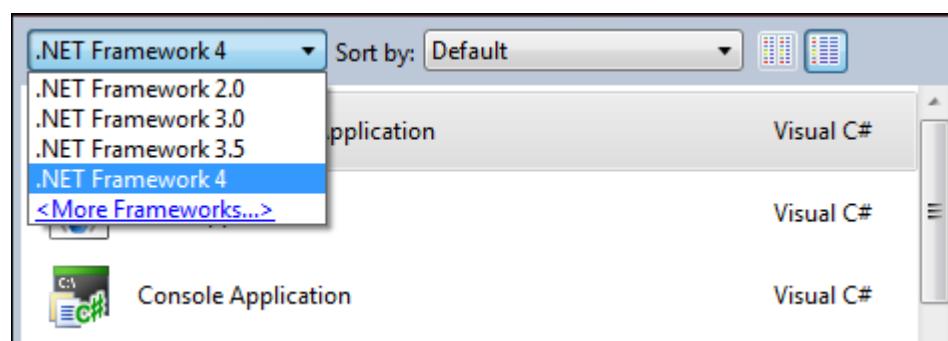
1. Tampilkan window New Project. Untuk menampilkan windows New Project dapat dilakukan dengan beberapa cara yaitu dengan mengklik link Create Project seperti gambar berikut.



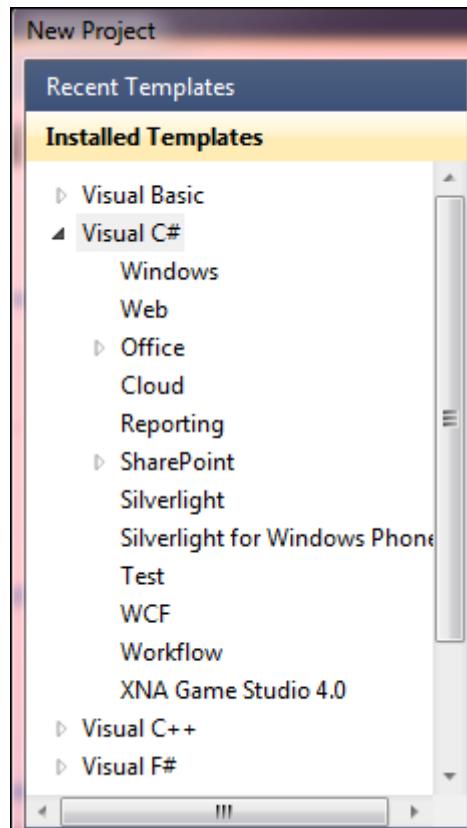
Cara kedua adalah dengan memilih menu File > New > Project atau cukup dengan menekan tombol Ctrl+Shift+N maka dapat dilihat window New Project seperti berikut :



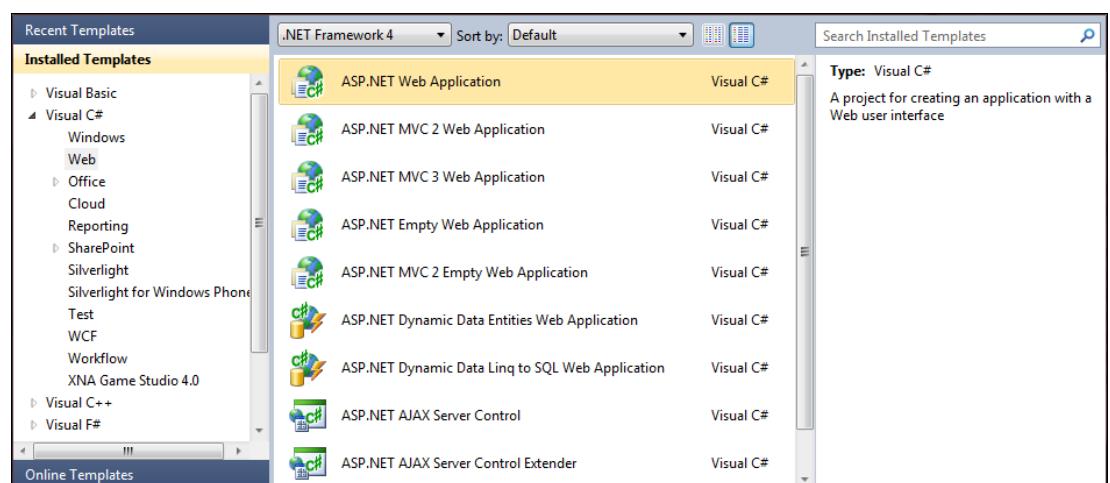
Pada bagian kanan atas dapat dilihat list versi .NET Framework, seperti pada gambar dibawah ini :



Pada sisi kiri dapat dilihat kolom Intalld Templates yang berisi tipe project yang dapat digunakan.



Sedangkan pada sisi kanan dapat dilihat kolom Templates berdasarkan Project types yang dipilih.



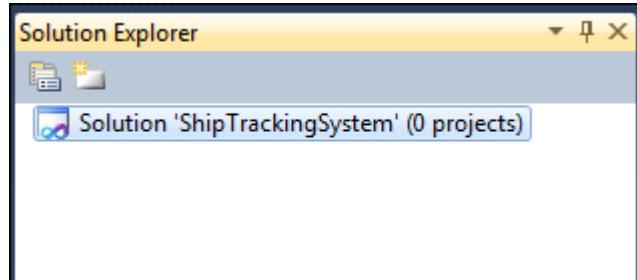
Dapat dilihat pada gambar di atas, ketika memilih Web pada kolom Project types maka kolom Templates dapat dilihat template project yang tergolong dalam tipe project Web.

2. Untuk membuat Solution, pada kolom Project types pilih Other Project Types > Visual Studio Solution. Kemudian pada kolom Templates pilih Blank Solution. Selanjutkan berikan nama Solution yang diinginkan pada kolom Name, lokasi penyimpanan pada kolom Location.



3. Kemudian klik tombol OK, maka dapat ditemui folder ShipTrackingSystem pada folder D:\Erase. Di dalam folder ShipTrackingSystem dapat dilihat file ShipTrackingSystem.sln.

Setelah langkah-langkah di atas telah dilakukan maka dapat dilihat pada Solution Explorer yang ada pada sisi kanan Visual Studio akan terdapat nama Solution yang baru saja dibuat.



Menambah Project pada Solution

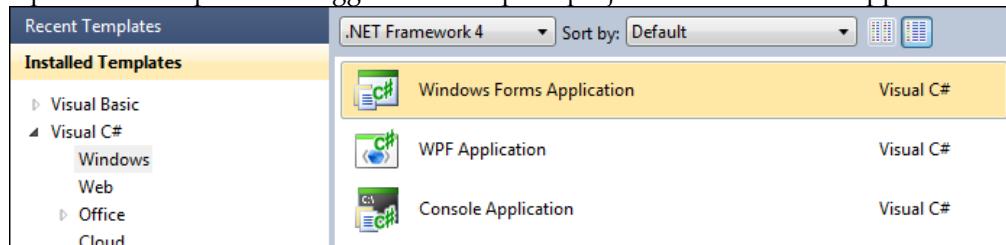
Ketika membangun suatu sistem dapat saja sistem tersebut dibangun dalam satu project saja artinya dalam satu Solution hanya terdiri atas satu Project.

Suatu sistem kadang dapat terdiri atas beberapa aplikasi, sehingga pada satu Solution dapat dibuat beberapa project yang mewakili masing-masing aplikasi tersebut. Sebagai contoh pada sistem perpustakaan yang didalamnya terdapat 2 aplikasi yaitu :

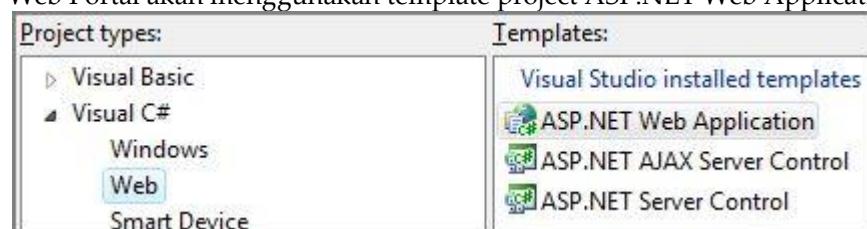
1. Aplikasi desktop yang digunakan untuk pencatatan transaksi simpan pinjam buku di perpustakaan.
2. Web Portal yang berfungsi untuk menampilkan data-data buku yang dimiliki di perpustakaan tersebut beserta status peminjamannya.

Dari kasus tersebut otomatis pada Solution dibuat dua project yaitu :

1. Aplikasi desktop akan menggunakan template project Windows Form Application.

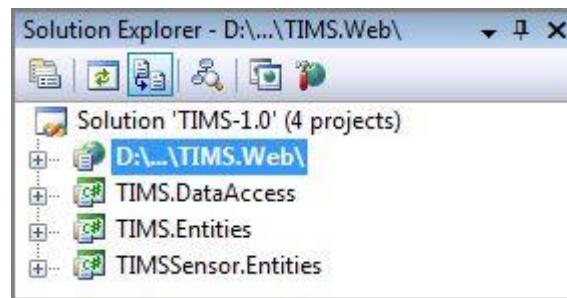


2. Web Portal akan menggunakan template project ASP.NET Web Application.



Selain kedua kasus di atas bisa juga terjadi kasus dimana akan dibuat sebuah sistem dengan aplikasi web tetapi pada Solution terdapat lebih dari satu project. Hal ini biasanya dilakukan untuk memisahkan masing-masing layer yang terdapat pada aplikasi kedalam satu project. Misalnya seperti berikut :

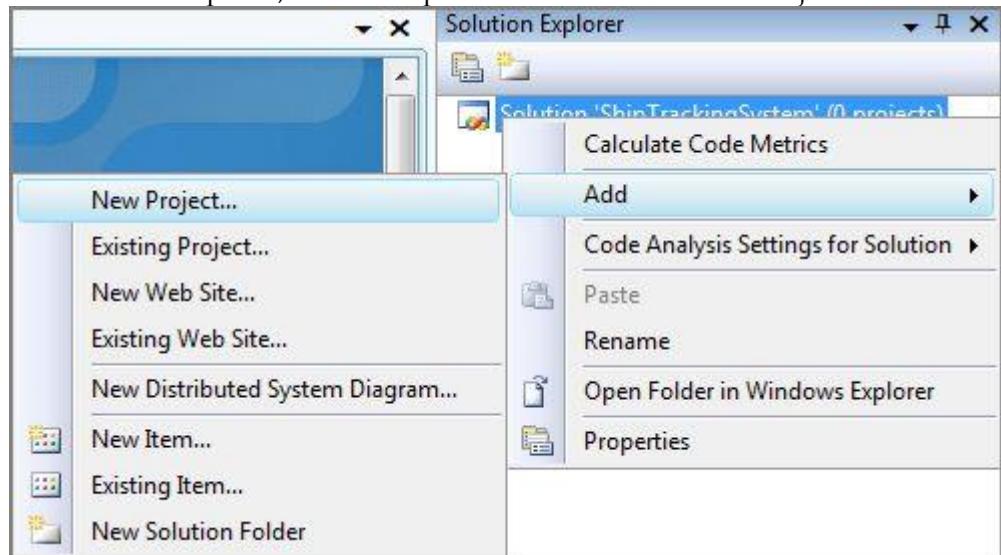
1. Project Web Application.
2. Project Class Library yang berfungsi sebagai Data Access Layer yang akan digunakan oleh aplikasi web.
3. Project Class Library yang berfungsi sebagai penyimpan class-class entities.



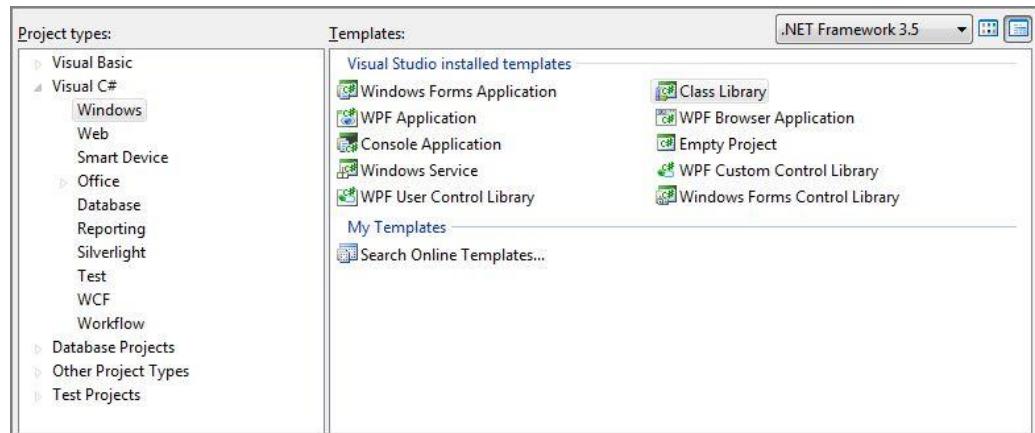
Pada kasus ini, salah satu project dapat menjadi reference dari project lainnya. Sebagai contoh project TIMS.DataAccess digunakan sebagai reference oleh project TIMS.Entities dan TIMSSensor.Entities. Dan WebSite TIMS.Web menggunakan project TIMS.DataAccess, TIMS.Entities dan TIMSSensor.Entities sebagai reference.

Untuk menambahkan project pada Solution langkah yang dilakukan adalah sebagai berikut :

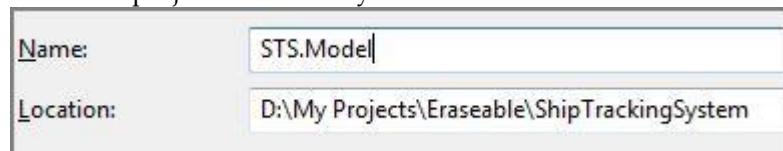
1. Pada Solution Explorer, klik kanan pada Solution > Add > New Project.



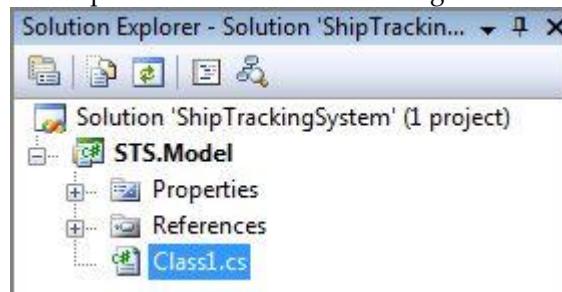
2. Maka akan ditampilkan window New Project. Pilih bahasa yang diinginkan pada kolom Project types, kemudian tipe project yang diinginkan. Setelah itu pilih template project pada kolom Templates. Misal dipilih Visual C# > Windows > Class Library.



3. Beri nama project dan lokasinya.

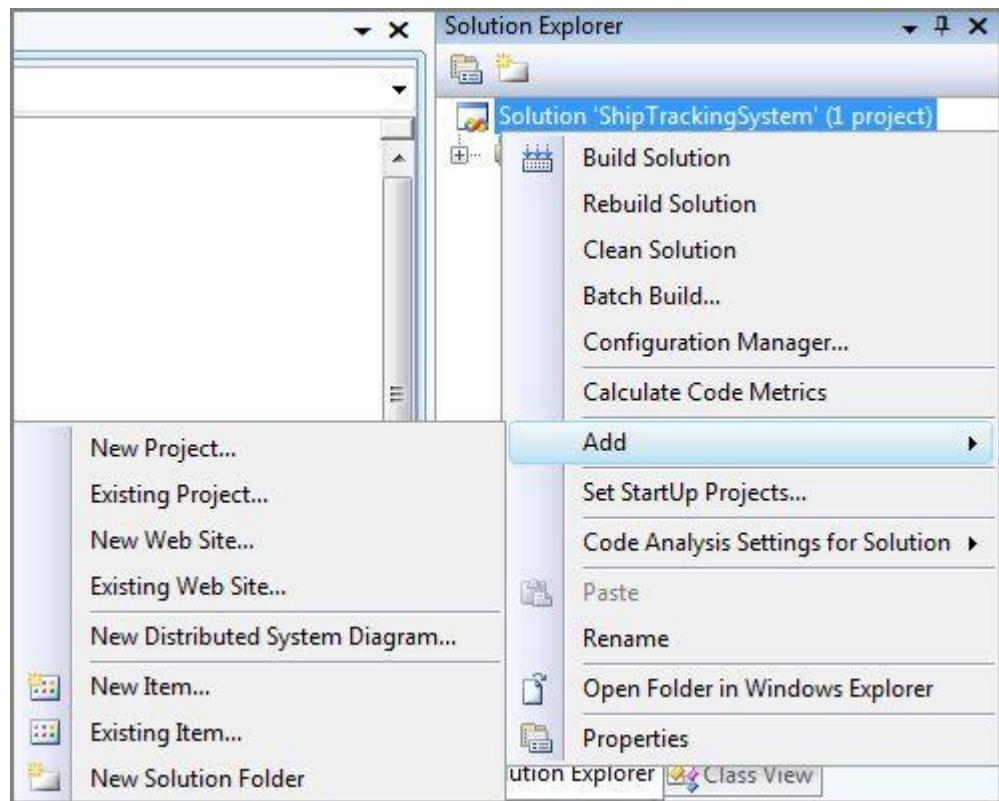


4. Klik OK, maka pada Solution Explorer dapat dilihat terdapat project dengan nama STS.Model. pada direktori D:\My Projects\Eraseable\ShipTrackingSystem dapat dilihat pula tambahan direktori dengan nama STS.Model.

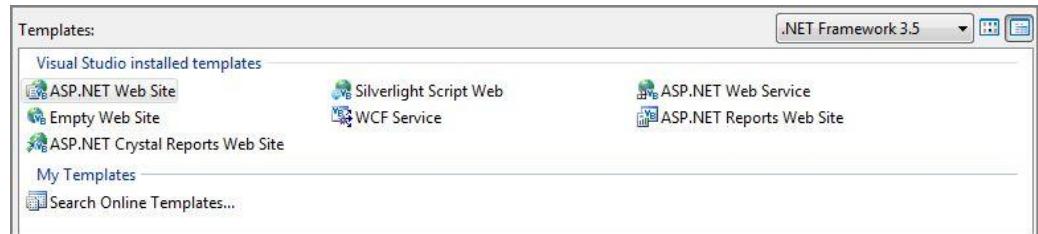


Selain menambahkan project dapat juga ditambahkan Web Site pada Solution. Langkah yang mesti diikuti adalah sebagai berikut :

1. Klik kanan pada Solution yang ada pada Solution Explorer, pilih Add > New Web Site.



2. Pilih template website pada window Add New Web Site seperti pada gambar di bawah ini.



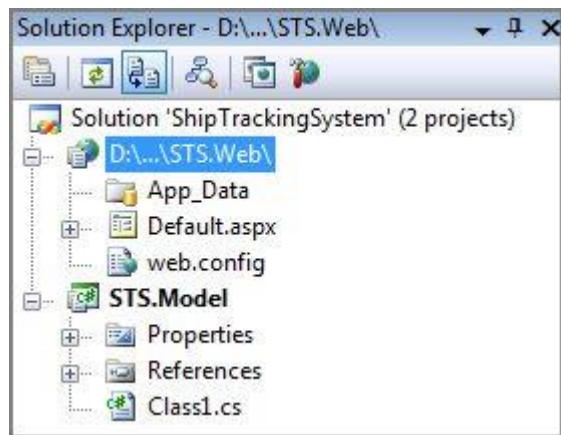
3. Pada dropdownlist Location pilih lokasi yang diinginkan, ada 3 pilihan yaitu :
 - a. File System artinya di lokal komputer.
 - b. HTTP.
 - c. FTP.

Pilih File System dan tulis direktori yang diinginkan untuk menyimpan file-file Web Site.

Pada dropdownlist Language pilih bahasa pemrograman yang ingin digunakan.



4. Klik OK, dan dapat dilihat pada Solution terdapat Web Site.



Toolbox

Pada sisi kiri Visual Studio dapat dilihat tombol Toolbox yang akan memunculkan pane Toolbox ketika tombol ini diklik. Pada Toolbox terdapat control-control yang dapat digunakan pada halaman aplikasi yang akan dibangun. Jenis Toolbox dan control yang terlihat pada Visual Studio tergantung dari jenis file yang aktif saat itu pada editor Visual Studio.

Toolbox dan control hanya dapat dilihat ketika file yang aktif tersebut berhubungan dengan designer atau antarmuka.

Sebagai contoh ketika file Class1.cs yang ada pada project STS.Model dibuka dan aktif pada editor Visual Studio, maka pada tidak dilihat control pada Toolbox. Ini disebabkan file Class1.cs tidak berhubungan dengan designer atau antarmuka. Berbeda ketika file Default.aspx yang ada pada Web Site STS.Web dibuka dan aktif pada editor Visual Studio, dapat dilihat control-control pada Toolbox seperti pada gambar di bawah ini.

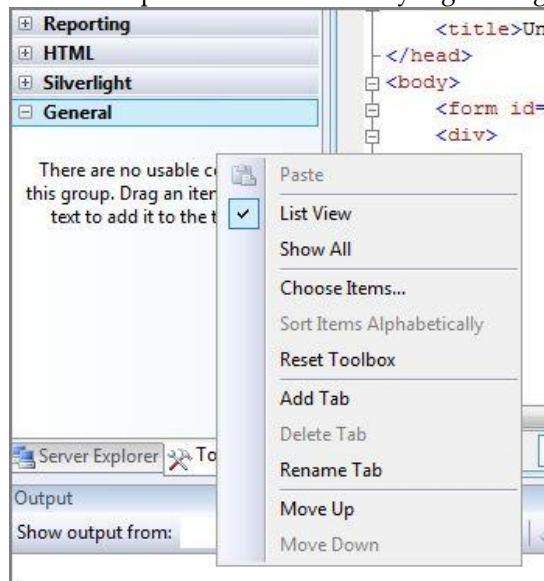


Control-control pada Toolbox akan berbeda sesuai dengan jenis file atau projectnya. Pada file yang tergolong dalam project Windows Form Application akan mempunyai control-control berikut pada Toolbox.



Control atau item pada Toolbox dapat ditambah tetapi mesti diketahui terlebih dahulu tipe project template dari control atau item yang akan ditambahkan itu. Misalkan ingin menambahkan control atau item Ajax Control Toolkit untuk template Web Site atau Web Application maka langkah-langkah yang dilakukan adalah :

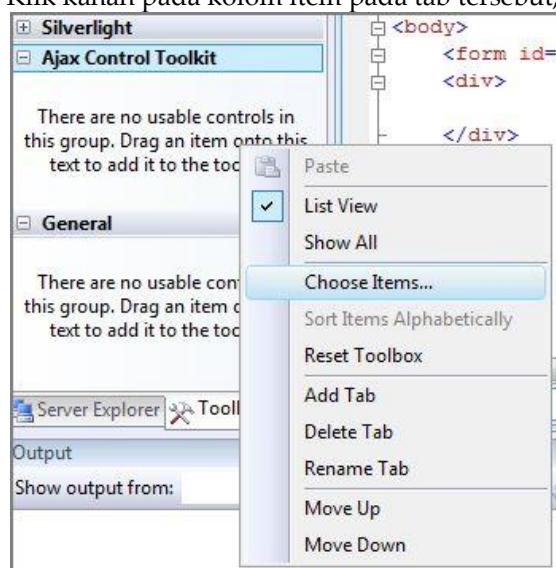
1. Aktifkan file dengan extension ASPX agar pada Toolbox ditampilkan control yang berhubungan dengan project template Web Site atau Web Application.
2. Klik kanan pada daerah Toolbox yang kosong dan pilih Add Tab.



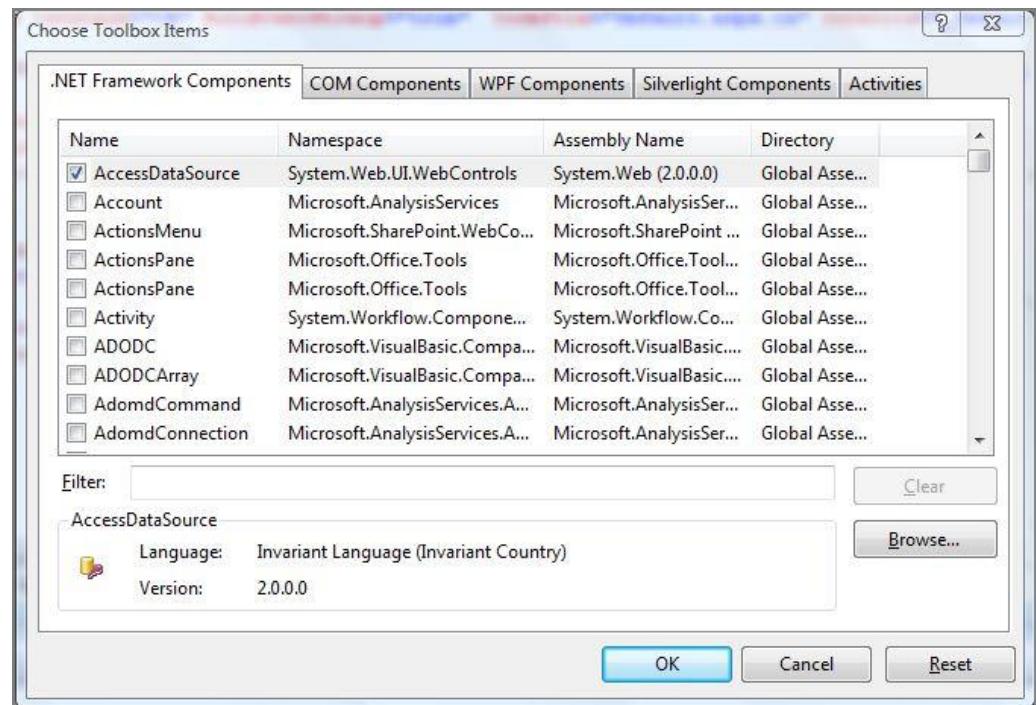
3. Pada tab baru tersebut ketikkan nama tab yang diinginkan, misalkan Ajax Control Toolkit.



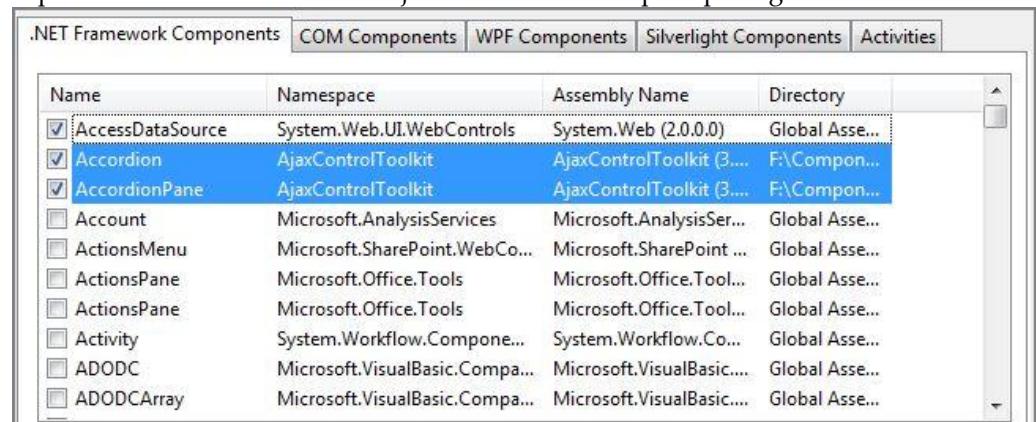
4. Klik kanan pada kolom item pada tab tersebut, dan pilih Choose Items...



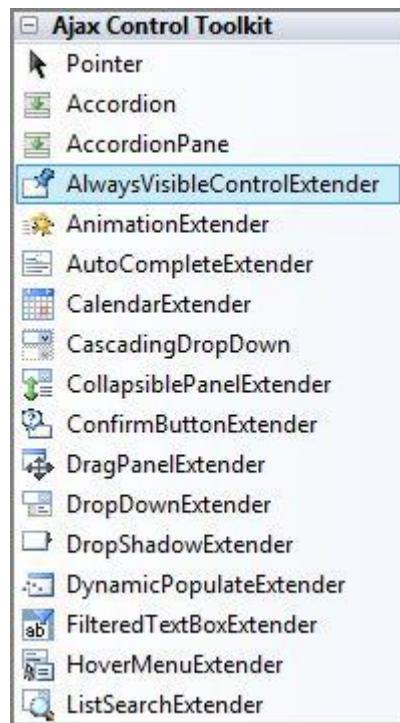
5. Selanjutnya akan ditampilkan window Choose Toolbox Items. Item dapat dipilih pada daftar yang tersedia pada window Choose Toolbox Items. Bila ingin memasukkan control dari library luar (biasanya adalah file dengan extension DLL) maka klik tombol Browse dan pilih DLL yang diinginkan.



- Karena ingin menambahkan control Ajax Control Toolkit maka, klik Browse dan tambahkan AjaxControlToolkit.dll. Makda pada tab .NET Framework Components dapat dilihat tambahan control AjaxControlToolkit seperti pada gambar berikut.



- Klik Tombol OK, maka dapat dilihat pada tab Ajax Control Toolkit pada Toolbox terlihat control-control baru.



Editor

Editor pada Visual Studio dapat dilihat pada gambar berikut :

The screenshot shows the Visual Studio Source Editor for a file named 'Default.aspx'. The editor displays the following ASPX code:

```
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>

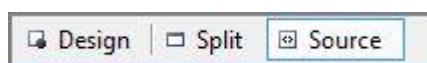
            &nbsp;
            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="Button1" runat="server" Text="Button" />

        </div>
    </form>
</body>
</html>
```

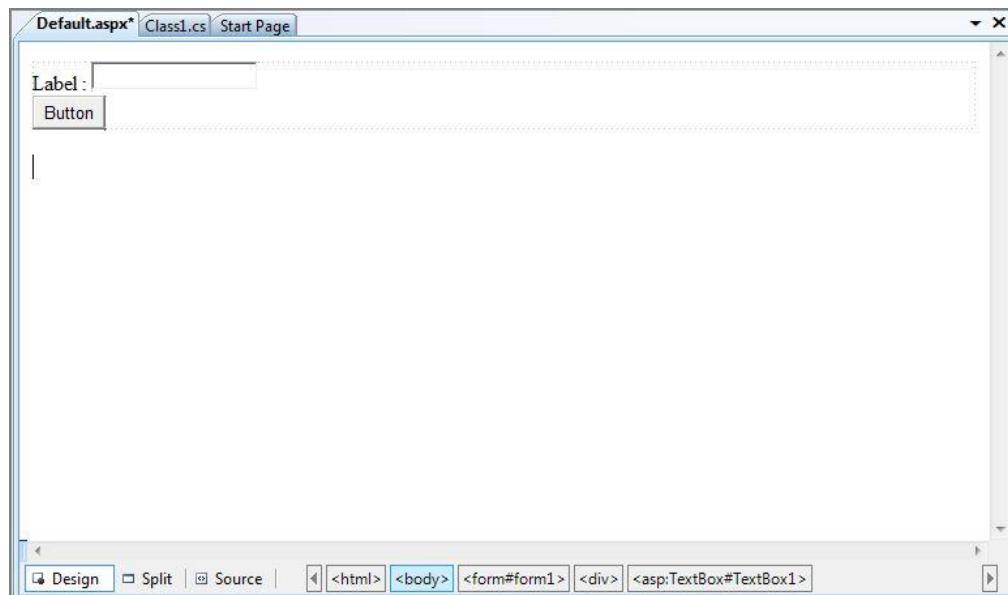
The tabs at the bottom of the editor are labeled 'Design', 'Split', and 'Source', with 'Source' being the active tab.

Pada bagian atas dapat dilihat tab-tab yang menunjukkan fil-file yang telah dibuka pada editor.

Apabila file yang dibuka adalah file yang mempunyai antarmuka maka pada bagian bawah dapat dilihat tombol-tombol berikut ini :



Bila tombol Souce aktif maka editor akan menampilkan source code dari file tersebut. Bila tombol Design yang aktif maka editor akan menampilkan antarmuka dari file yang sedang dibuka. Seperti pada gambar di bawah ini.



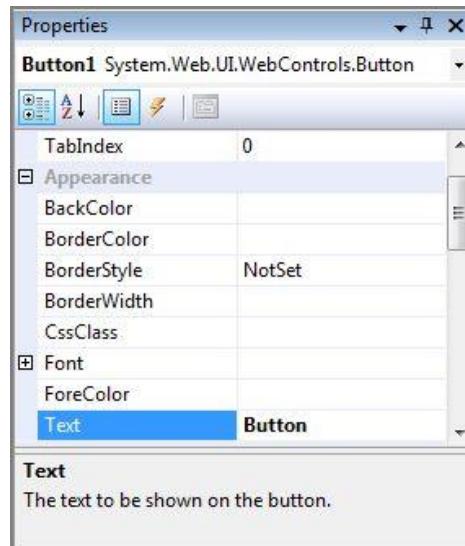
Control-control yang tersedia pada Toolbox dapat di drag-n-drop pada suatu halaman baik pada mode Design ataupun Source.

Pada mode Design, ketika control dari Toolbox yang di letakkan pada suatu file (misalnya file Default.aspx / file dari project Web Application atau Web Site) maka dapat dilihat property dari control tersebut yang terbagi atas dua yaitu :

1. Property yang berhubungan dengan nilai dari control tersebut seperti panjang, lebar, ID, text, warna dan lain-lain.
2. Event.

Untuk melihat property dari suatu control hal yang mesti dilakukan adalah :

1. Pada mode Design pilih control yang diinginkan. Bila pada halaman belum ada control yang tambahan, maka pilih salah satu control pada Toolbox dan drag-n-drop pada halaman tersebut. Misalnya control yang ditambahkan pada halaman adalah Button yang merupakan group Standard pada Toolbox.
2. Ketika control telah dipilih dengan cara mengklik control tersebut maka pada bagian kanan Visual Studio di bawah kolom Solution Explorer akan ditampilkan kolom Properties.

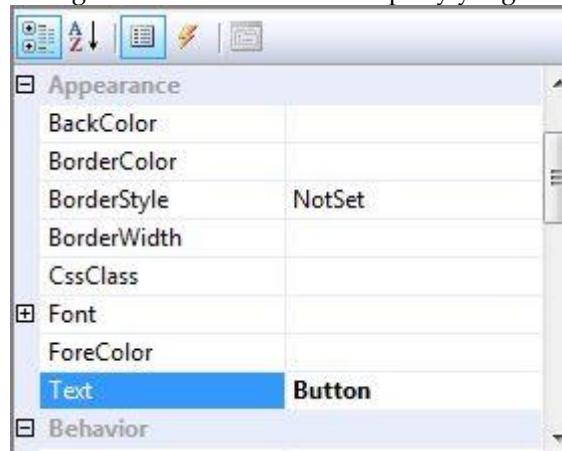


3. Pada kolom Properties terdapat dua tombol yang penting yaitu tombol Properties dan tombol Event.

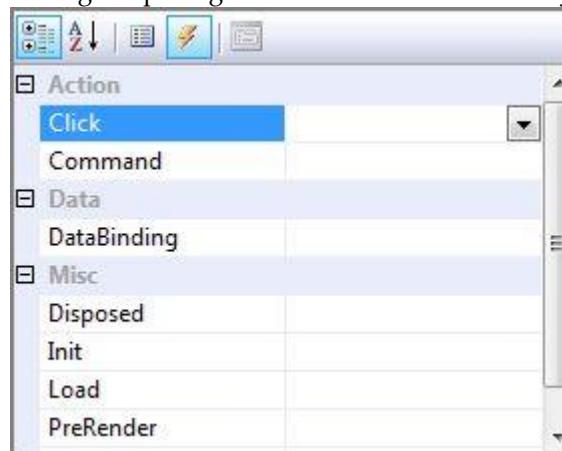


Tombol pada sisi kiri adalah tombol untuk menampilkan Property dari control sedangkan tombol pada sisi kanan adalah tombol untuk menampilkan Event yang dimiliki oleh control.

Pada gambar berikut adalah Property yang dimiliki oleh control Button :



Sedangkan pada gambar berikut adalah Event yang dimiliki oleh control Button :



4. Property Button seperti BackColor, BorderWidth, Font, Text dan lain-lain dapat ditambahkan atau diubah dengan memasukkan nilai yang diinginkan. Perubahan yang terjadi pada nilai-nilai property ini akan dicatat sebagai atribut dari control tersebut.

Misalnya nilai-nilai pada Property diubah seperti berikut :

Appearance	
BackColor	#3399FF
BorderColor	
BorderStyle	NotSet
BorderWidth	1px
CssClass	
Font	
ForeColor	Black
Text	Simpan

Maka pada dapat dilihat kode control Button pada file Default.aspx akan seperti berikut :

```
<asp:Button ID="Button1" runat="server"
    BackColor="#3399FF" BorderWidth="1px"
    ForeColor="Black" Text="Simpan" />
```

5. Pada kelompok Event yang terdapat pada control Button terdapat event Click yang berfungsi untuk melakukan aksi tertentu ketika tombol diklik. Untuk menambahkan event ini pada control Button maka cukup klik dua kali pada kolom kosong disisi text Click.

Action	
Click	

Maka secara otomatis Visual Studio akan membantu membuatkan beberapa hal, yaitu :

- a. Memodifikasi atribut control Button pada file Default.aspx menjadi seperti berikut,

```
<asp:Button ID="Button1" runat="server"
    BackColor="#3399FF" BorderWidth="1px"
    ForeColor="Black" Text="Simpan"
    onclick="Button1_Click" />
```

Dapat dilihat terdapat tambahan event onclick yang akan memanggil method Button1_Click.

- b. Memodifikasi file Default.aspx.cs (merupakan file CodeBehind dari Default.aspx, bila bahasa yang digunakan adalah VB.NET maka file ini akan mempunyai nama Default.aspx.vb). Modifikasi yang dilakukan oleh Visual Studio adalah menambahkan method Button1_Click.

```
protected void Button1_Click(object sender, EventArgs e)
{
}
```

Di dalam method ini dapat ditambahkan baris-baris kode yang ingin dijalankan ketika Button diklik.

- c. Pada kolom Properties dapat dilihat perubahan seperti berikut,



Build, Execute & Debug

Sebelum aplikasi pada suatu project dieksekusi terlebih dahulu project tersebut akan di-build. Proses build ini dapat dilakukan pada suatu project yang diinginkan atau pada seluruh project yang berada pada solution.

Untuk melakukan proses build pada suatu project, cukup klik kanan pada project yang diinginkan dan pada menu pilih Build. Hal yang sama juga dilakukan ketika ingin melakukan proses build pada solution, klik kanan pada solution dan pilih Build pada menu.

Bila pada project terjadi kesalahan maka pada bagian bawah dari Visual Studio dapat dilihat pesan kesalahan seperti berikut :



Pada gambar di atas dapat dilihat kesalahan terjadi pada project dengan nama STS.Model, pada file Class1.cs pada baris ke 10. Dengan melakukan klik dua kali pada baris tersebut maka Visual Studio akan mengantarkan pada file bersangkutan.

Bila project sukses di build maka dapat dilanjutkan proses selanjutnya yaitu proses eksekusi. Yang perlu diingat adalah tidak seluruh project dapat dieksekusi yang dilihat hasilnya. Sebagai contoh project dari template project Class Library tidak dapat dilihat hasilnya karena keluaran dari project ini hanya berupa library dalam bentuk file DLL dan tidak ada antarmuka sama sekali.

Contoh project yang dapat dieksekusi adalah project yang menggunakan project template Windows Form Application. Untuk mengeksekusi project ini langkah yang dilakukan adalah dengan melakukan klik kanan pada project tersebut dan pilih Debug > Start new instance.

Sedangkan untuk Web Site atau Web Application untuk mengeksekusi langkah yang dilakukan adalah dengan melakukan klik kanan pada website atau web application kemudian pilih View in browser pada menu. Maka Visual Studio akan menghidupkan web server internal dan membuat web browser untuk menampilkan web site atau web application.

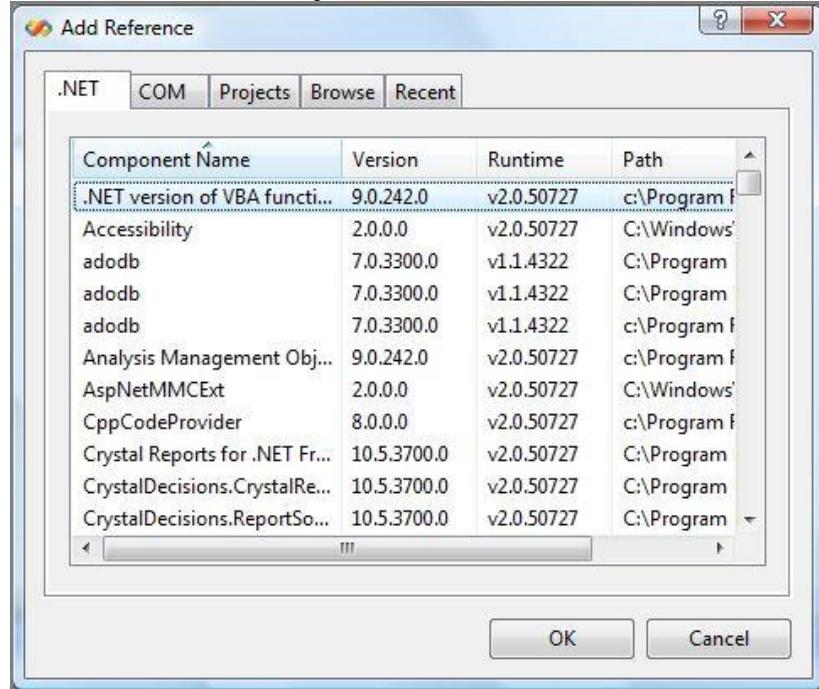
Menambah Reference pada Project

Ada kalanya suatu project memerlukan suatu library tertentu untuk digunakan di dalam aplikasi. Maka perlu ditambahkan reference pada project yang membutuhkan. Library tersebut dapat berupa :

1. Library yang telah tersedia pada .NET Framework.
2. Library dari suatu file dengan extension DLL.
3. Library hasil output suatu project (file dengan extension DLL) yang berada pada project yang berada pada solution yang sama.

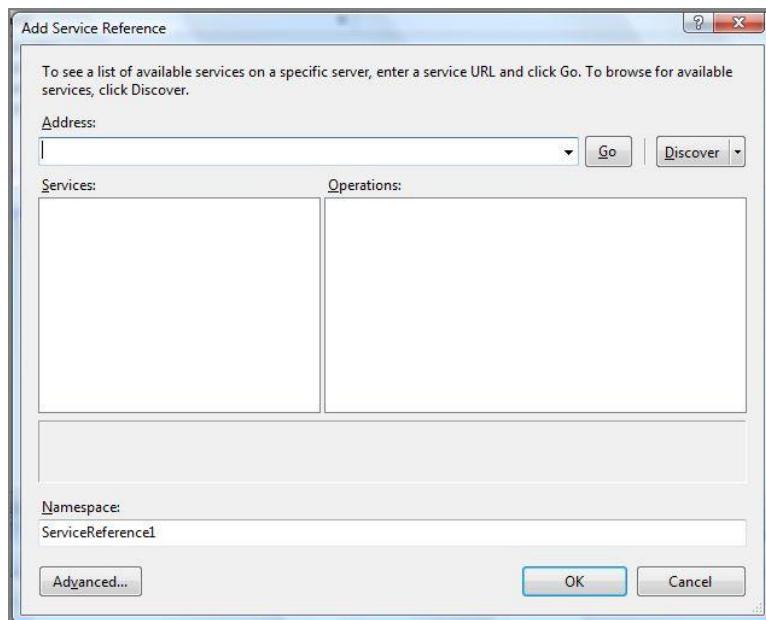
Untuk menambahkan library tersebut langkah yang mesti dilakukan adalah :

1. Klik kanan pada project kemudian pilih Add Reference. Pada akan ditampilkan windows Add Reference seperti berikut.



2. Bila ingin menambahkan library yang dimiliki .NET Framework maka pilih tab .NET, untuk menambahkan library dari file dengan extension DLL maka pilih tab Browser. Bila ingin menambahkan output suatu project pada solution yang sama sebagai reference maka klik tab Projects.
3. Pilih item yang diinginkan kemudian klik OK.

Selain reference di atas dapat juga menambahkan reference dalam bentuk web service. Caranya dengan melakukan klik kanan pada project yang ada pada Solution Explorer kemudian pilih Add Service Reference. Maka akan ditampilkan window Add Service Reference seperti berikut:



Tulis alamat service pada kolom Address.

3

Pengenalan C#

Pada bagian ini akan dipaparkan dasar-dasar pemrograman dengan C#. pada bagian ini tidak akan dibahas secara lengkap tentang pemrograman dengan C# tetapi hal-hal yang dianggap perlu saja sebagai dasar untuk memulai membangun aplikasi web.

Hello World with C#

Program kecil untuk menampilkan kalimat "Hello World" biasanya dibuat saat belajar bahasa pemrograman tertentu. Pada bagian ini akan dibuat program kecil untuk menampilkan kalimat tersebut dengan bahasa C#.

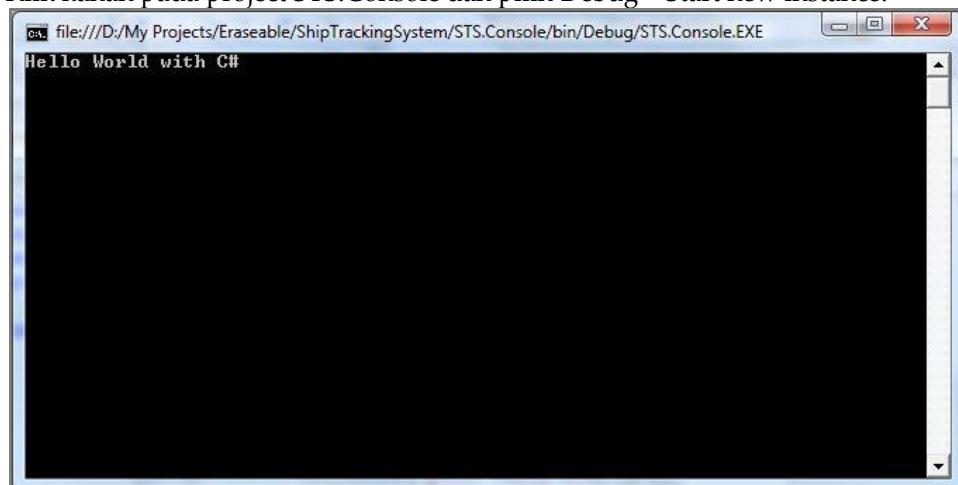
Untuk itu perlu diikuti langkah-langkah berikut :

1. Membuat project dengan menggunakan tipe project Visual C# > Windows, dan template project yang digunakan adalah Console Application. Tambahkan template project ini pada Solution dan beri nama STS.Console.
2. Pada file Program.cs tambahkan kode berikut :

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            System.Console.WriteLine("Hello World with C#");
            System.Console.ReadKey();
        }
    }
}
```

3. Klik kanan pada project STS.Console dan pilih Debug > Start new instance.



Pada kode di atas ada kata kunci yang mungkin baru ditemui oleh programmer yang baru memulai menggunakan C#, yaitu :

1. **namespace** adalah kata kunci yang digunakan untuk mendefinisikan suatu ruang lingkup atau batasan dalam kode program. Namespace berguna untuk mengorganisasikan kode program dan dapat digunakan untuk membuat type data yang global sekaligus unique.

Contoh :

```
namespace STS.Console  
{  
}
```

Namespace merupakan konsep yang dipinjam dari C++ yang dapat digunakan untuk menjamin bahwa semua penamaan yang anda gunakan dalam program bersifat unique. Dengan menggunakan namespace, anda bisa mengelompokkan class, interface, struct, enum dan delegate dengan namespace yang berbeda agar tidak terjadi konflik penamaan dan organisasi dari kode anda menjadi lebih terstruktur. Untuk developer yang bekerja membuat class library dalam proyek software berskala besar, namespace penting untuk mengorganisasikan class-class ke dalam satu struktur hirarki.

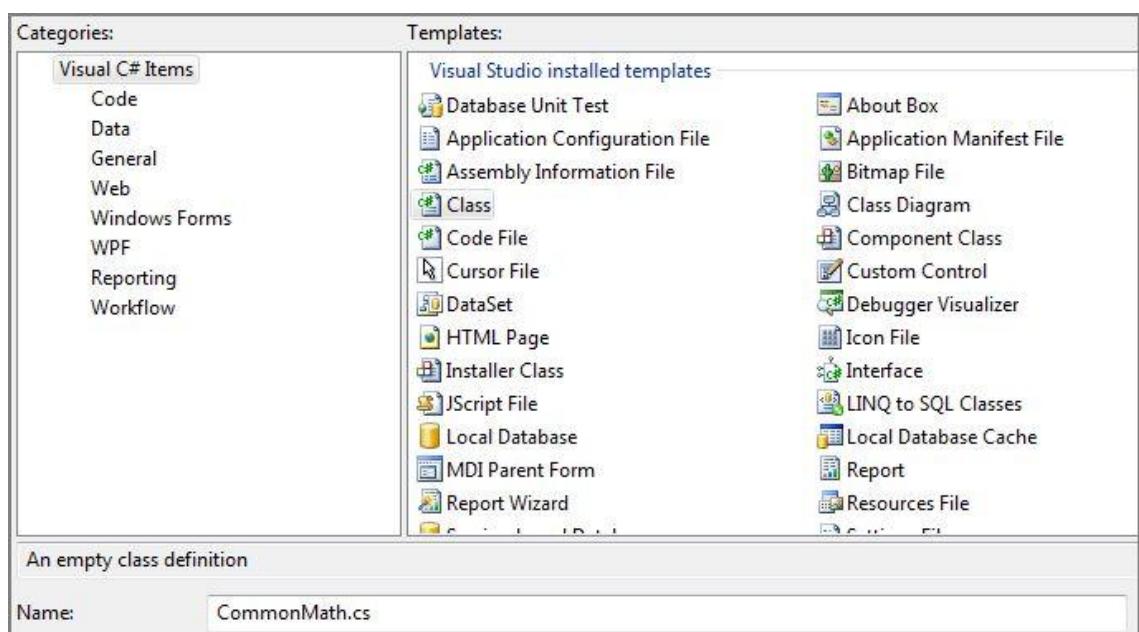
2. Program C# diorganisasikan oleh namespace. Namespace dapat digunakan sebagai sistem organisasi "internal" dalam program anda dan juga secara "eksternal" karena dapat diakses secara public menggunakan kata kunci **using**.

Contoh :

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;
```

Untuk mengerti lebih jauh penggunaan kata kunci namespace dan using maka dapat mengikuti contoh berikut ini.

Pada project STS.Console buat class dengan cara klik kanan pada project kemudian pilih Add > Class. Pada window Add New Item berikan nama CommonMath untuk class tersebut.



Dan berikut adalah kode pada class CommonMath.

```

CommonMath.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Common.Math
{
    public class CommonMath
    {
        public void Tambah(int a, int b)
        {
            int hasil = a + b;
            System.Console.WriteLine(hasil);
        }
    }
}

```

Pada class di atas dapat dilihat class CommonMath dikelompokkan dalam namespace STS.Common.Math.

Bila ingin menggunakan class ini pada aplikasi console ini, maka file Program.cs perlu dimodifikasi seperti berikut :

```

Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            CommonMath commonMath = new CommonMath();
            commonMath.Tambah(5, 6);

            System.Console.ReadKey();
        }
    }
}

```

Karena class CommonMath berada di dalam namespace STS.Common.Math maka pada file Program.cs perlu ditambahkan referensi ke namespace tersebut dengan menggunakan baris berikut :

```

using STS.Common.Math;

```

Tanpa menambahkan baris tersebut maka baris berikut ini tidak akan dikenali dan akan mengeluarkan pesan error saat proses debug.

```

CommonMath commonMath = new CommonMath();

```

Variabel

Variabel adalah tempat untuk menyimpan data dengan tipe data tertentu. Variabel akan mewakili suatu lokasi di memori komputer. Dengan menggunakan nama variabel ini maka data yang berada pada memori dapat diakses.

C# termasuk ke dalam bahasa pemrograman yang strong-typed artinya variabel harus mempunyai tipe data. Dan variabel harus diberikan nilai berdasarkan tipe data yang telah diberikan.

Aturan Penamaan Variabel

Dalam penamaan variabel ada hal-hal yang tidak boleh dilakukan, diantaranya adalah :

1. Nama variabel tidak boleh diawali dengan angka.
2. Nama variabel tidak boleh mengandung karakter #.
3. Nama variabel tidak boleh memakai kata kunci yang sudah digunakan pada C# seperti int, public, while dan lain-lain.

Untuk penamaan variabel dapat mengikuti konvensi yang umum digunakan, yaitu :

1. Notasi Pascal, setiap kata yang digunakan sebagai nama variabel diawali dengan huruf besar. Contoh :
 - a. Alamat.
 - b. NamaSiswa.
 - c. GroupId.
2. Notasi Camel, huruf awal dari variabel adalah huruf kecil sedangkan bila terdiri atas dua atau lebih kata maka huruf pertama dari kata kedua dan seterusnya diawali dengan huruf besar. Contoh :
 - a. alamat.
 - b. namaSiswa.
 - c. groupId.
3. Notasi Hungarian, sama dengan aturan penamaan dengan menggunakan notasi Camel tetapi kata awal adalah menyatakan tipe data dari variabel tersebut. Contoh :
 - a. strAlamat.
 - b. strNamaSiswa.
 - c. intGroupId.

Kategori Variabel

Variabel pada C# dapat dibedakan menjadi :

1. Variabel instance adalah variabel yang merupakan bagian dari instance.
2. Variabel static adalah variabel yang bukan merupakan bagian dari object.
3. Variabel lokal adalah variabel yang dideklarasikan di dalam suatu blok, statement for, switch atau using. Sebelum nilai dari variabel lokal dapat diakses maka variabel ini perlu diberikan suatu nilai.
4. Parameter adalah variabel yang ditemui di dalam suatu method. Parameter dibagi atas 3 jenis yaitu :
 - a. Value parameter.
 - b. Output parameter.
 - c. Reference parameter.

Value parameter dan reference parameter harus diberikan nilai awal, berbeda dengan output parameter yang boleh tidak diberikan nilai awal.

Contoh :

```
CommonMath.cs
using System;
using System.Collections.Generic;
using System.Linq;
```

```

using System.Text;

namespace STS.Common.Math
{
    public class CommonMath
    {
        public void Perhitungan(int x, out int y, ref int z)
        {
            x = x * 10;
            y = x + 10;
            z = x * 100;
        }
    }
}

```

Program.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            CommonMath commonMath = new CommonMath();
            int a = 10;
            int b;
            int c = 9;
            commonMath.Perhitungan(a, out b, ref c);
            System.Console.WriteLine("Nilai a = " + a);
            System.Console.WriteLine("Nilai b = " + b);
            System.Console.WriteLine("Nilai c = " + c);

            System.Console.ReadKey();
        }
    }
}

```

Hasilnya adalah sebagai berikut :

```

file:///D:/My Projects/Eraseable/ShipTrackingSystem/STS.C
Nilai a = 10
Nilai b = 110
Nilai c = 10000

```

5. Elemen array.

Berikut adalah contoh dari point 1,2,3 dan 5.

VariableExample.cs

```

class VariableExample
{
    public static int x;
    public int y;

    public void OlahKalimat(string kalimat)
    {
        string[] kata = kalimat.Split(',');
    }
}

```

```
}
```

Ekspresi

Ekspresi terbentuk dari rangkaian operator dan operand. Operator yang terdapat dalam suatu ekspresi menyatakan proses apa yang akan dilakukan pada suatu operand. Contoh dari operator adalah +, -, *, / dan new. Sedangkan contoh dari operand adalah literal, fields, variabel lokal dan ekspresi, suatu ekspresi dapat digunakan untuk membentuk ekspresi yang lain yang lebih besar.

C# memiliki tiga macam operator, yaitu :

1. Operator unary, memiliki satu buah operand dan menggunakan notasi prefix atau postfix. Contoh postfix:

```
for (int i = 0; i < 5; i++)  
{  
}  
}
```

Contoh prefix :

```
int i = 5;  
while (i > 0)  
{  
    System.Console.WriteLine(i);  
    --i;  
}
```

2. Operator binary, menggunakan dua buah operand dan menggunakan notifikasi infix.

Contoh :

```
public void Perhitungan(int x, out int y, ref int z)  
{  
    x = x * 10;  
    y = x + 10;  
    z = x * 100;  
}
```

3. Operator ternary, memiliki dua buah operator. Pada C# yang termasuk ke dalam operator ternary adalah ?:.

Operator ini bisa dikatakan sebagai cara untuk mempersingkat penulisan percabangan dengan menggunakan if-else.

Sebagai contoh biasanya untuk percabangan ditulis dengan kode berikut :

```
string status = String.Empty;  
  
if (a > c)  
{  
    status = "nilai a lebih besar dari c";  
}  
else  
{  
    status = "nilai a lebih kecil dari atau sama dengan c";  
}  
  
System.Console.WriteLine(status);
```

Dengan menggunakan operator ?: kode di atas dapat diganti menjadi berikut :

```
string status = String.Empty;  
  
status = a > c ? " a lebih besar" : "a lebih kecil";
```

```
System.Console.WriteLine(status);
```

Percabangan

Dalam percabangan terdapat dua hal yang mesti diketahui yaitu :

1. Kondisi adalah syarat yang harus dipenuhi untuk menjalankan suatu aksi.
2. Aksi adalah sekumpulan perintah yang akan dijalankan apabila kondisi terpenuhi.

Untuk proses percabangan ini digunakan selection statement. Pada C# selection statement yang dapat digunakan adalah :

1. if dan if-else.
2. switch.

if dan if-else

if statement digunakan untuk mengeksekusi kode program jika kondisi tertentu terpenuhi. Berikut adalah contoh penggunaan statement ini.

```
int a = 10;
int c = 9;

string status = String.Empty;

if (a > c)
{
    status = "nilai a lebih besar dari c";
}
else if (a < c)
{
    status = "nilai a lebih kecil dari c";
}
else
{
    status = "nilai a sama dengan c";
}
```

switch

Berikut contoh penggunaan statement switch.

Contoh :

```
string nama = "reza";
switch (nama)
{
    case "reza":
    {
        System.Console.WriteLine("nama saya adalah reza.");
        break;
    }
    default:
    {
        System.Console.WriteLine("nama saya bukan reza.");
        break;
    }
}
```

Pengulangan

Statement jenis ini digunakan untuk menentukan bagian mana dari program yang akan dieksekusi berulang-ulang dan apa kondisi yang menentukan perulangan tersebut.

Pada C# ada empat statement yang digunakan sebagai pengulangan, yaitu :

1. while.
2. for.
3. foreach.
4. do-while.

while

While statement berguna untuk melakukan perulangan selama kondisi bernilai true.

Contoh :

```
int j = 0;
while (j < 5)
{
    System.Console.WriteLine(j);
    j++;
}
```

for

For digunakan untuk melakukan perulangan yang didasarkan atas nilai diskrit, misalnya integer.

Contoh :

```
for (int j = 0; j < 5; j++)
{
    System.Console.WriteLine(j);
}
```

foreach

Statement foreach digunakan untuk penelusuri suatu collection.

Contoh :

```
string[] days = { "minggu", "senin", "selasa", "rabu", "kamis", "jumat", "sabtu" };

foreach (string day in days)
{
    System.Console.WriteLine(day);
}
```

do-while

berbeda dengan while, pada statement do-while pengecekan kondisi dilakukan diakhir.

Berikut contoh penggunaan statement do-while.

```
int j = 6;
do
{
    System.Console.WriteLine(j);
}

while (j < 5);
```

Jump Statement

Jump statement digunakan untuk mentransfer kontrol eksekusi dari suatu bagian ke bagian lain dalam program.

Yang termasuk dalam statement ini adalah :

1. continue, berfungsi untuk melanjutkan eksekusi program ke iterasi berikutnya dengan “melompati” statement-statement berikutnya dalam badan loop.

Contoh :

```
for (int j = 0; j < 10; j++)
{
    if (j == 3) continue;
    System.Console.WriteLine(j);
}
```

2. break, statement ini digunakan untuk “melompat” keluar dari while, for, dan switch statement yang sudah dibahas sebelumnya.

Contoh :

```
for (int j = 0; j < 10; j++)
{
    System.Console.WriteLine(j);
    if (j == 3)
    {
        break;
    }
}
```

3. return, digunakan untuk mengembalikan kontrol eksekusi ke pemanggil.

4. throw, berkaitan dengan penanganan error (try statement) dalam C#. Throw digunakan untuk membangkitkan exception dalam program.

Contoh :

```
try
{
    System.Console.WriteLine("Masukkan angka pembagi :");
    int angka = Convert.ToInt32(System.Console.ReadLine());

    if (angka == 0)
    {
        throw new Exception("Angka tidak boleh 0");
    }
    System.Console.WriteLine("Angka yang dimasukkan:" + angka);
}
catch (Exception ex)
{
    System.Console.WriteLine(ex.Message);
}

System.Console.ReadKey();
```

5. goto, digunakan untuk melanjutkan eksekusi program di label yang sudah didefinisikan sebelumnya.

Contoh :

```
for (int j = 0; j < 10; j++)
{
    if (j == 3) goto selesai;
    System.Console.WriteLine(j);
}
```

```
selesai:  
System.Console.WriteLine("Selesai");
```

Penanganan Error

Untunglah dalam bahasa-bahasa pemrograman modern, seperti C++, Java, dan C#, telah disediakan cara baru untuk menangani error. Cara ini, yang dikenal sebagai exception handling, berkaitan erat dengan paradigma pemrograman berbasis object (OPP).

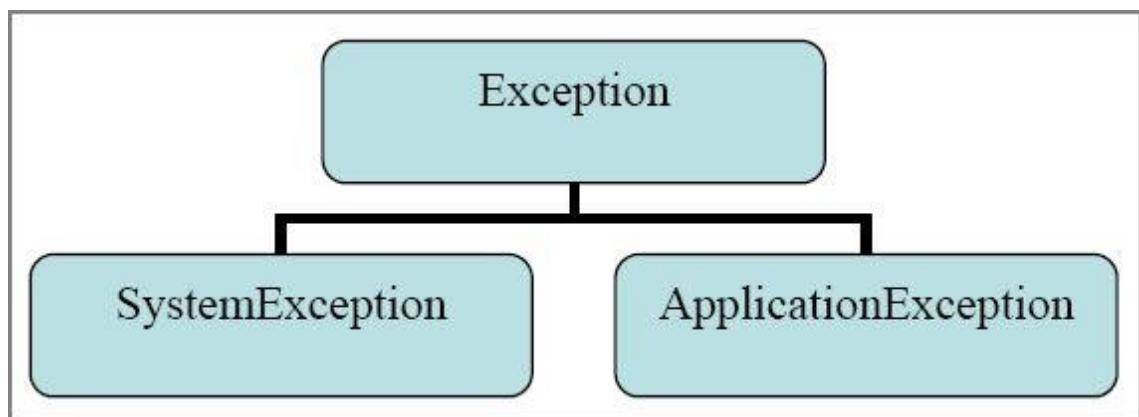
Ide utama dari exception handling ini adalah memisahkan antara kode-kode yang berisikan inti proses dan kode-kode yang digunakan untuk menangani error.

Secara teknis exception adalah objek yang merepresentasikan error yang muncul pada saat aplikasi dijalankan. Artinya mesti dilakukan proses pembuatan objek Exception baru, menginisiasi informasi-informasi didalamnya, dan menginformasikan pada sistem bahwa ada error yang muncul.

Exception menggunakan class yang berbeda untuk tiap kesalahan. Pada level yang paling atas adalah Exception dan mempunyai dua turunan yaitu :

1. SystemException, dihasilkan oleh CLR dan .NET Framework.
2. ApplicationException, digunakan untuk exception yang didefinisikan oleh aplikasi.

Berikut adalah gambar hirarki class Exception.



Dari atas ke bawah, class-class tersebut merepresentasikan exception yang makin spesifik.

Statement try

Statement yang digunakan untuk menangani exception adalah try-catch-finally dengan sintaks seperti berikut :

```
try  
{  
    // operasi yang mungkin menghasilkan exception  
}  
catch  
{  
    // penanganan exception  
}  
finally  
{  
    // langkah-langkah yang harus selalu dilakukan,  
    // ada atau tidak ada exception  
}
```

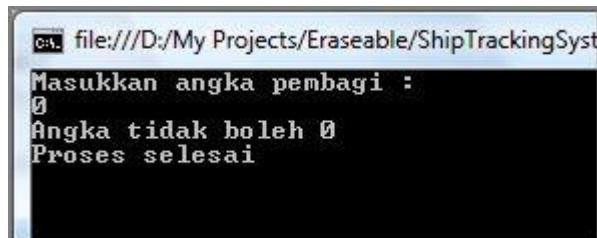
Contoh :

```
try
{
    System.Console.WriteLine("Masukkan angka pembagi :");
    int angka = Convert.ToInt32(System.Console.ReadLine());

    if (angka == 0)
    {
        throw new Exception("Angka tidak boleh 0");
    }
    System.Console.WriteLine("Angka yang dimasukkan adalah " + angka);
}
catch (Exception ex)
{
    System.Console.WriteLine(ex.Message);
}
finally
{
    System.Console.WriteLine("Proses selesai");
}

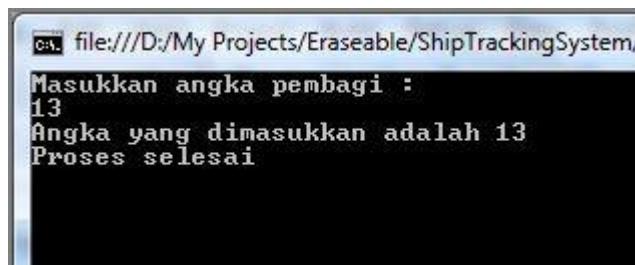
System.Console.ReadKey();
```

Hasil dari kode di atas bila dimasukkan angka 0 adalah sebagai berikut :



```
file:///D:/My Projects/Eraseable/ShipTrackingSyst
Masukkan angka pembagi :
0
Angka tidak boleh 0
Proses selesai
```

Dan berikut adalah hasil ketika angka yang dimasukkan adalah bilangan selain 0 :



```
file:///D:/My Projects/Eraseable/ShipTrackingSystem/
Masukkan angka pembagi :
13
Angka yang dimasukkan adalah 13
Proses selesai
```

Dari hasil di atas maka dapat diketahui perintah yang berada pada blok finally akan selalu dieksekusi baik ada atau tidak ada exception.

Pemrograman Object-Oriented

Class dan Objek

Untuk membuat Class dengan C# maka berikut adalah sintaks yang mesti diketahui.

```
[attribute]
[access-modifier] class nama_class [: [base-class [, interface(s)]]]
{
    class-body
}
```

Keterangan :

1. Attribute, adalah penanda untuk menerangkan class dan untuk kebutuhan tertentu.
2. Access-modifier dapat berupa public, private, protected, internal dan protected internal.
 - a. public, tidak ada batasan, class dapat digunakan oleh semua class.
 - b. private, terbatas tidak dapat digunakan pada class lain.
 - c. protected, terbatas digunakan oleh class turunannya.
 - d. internal, dapat digunakan secara internal di dalam class.
3. Tanda ":" untuk pengganti kata kunci extends bagi yang sudah mengenal OOP pada Java atau PHP.

Contoh :

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using STS.Common.Math;

namespace STS.Console
{
    [Obsolete("Class ini sudah tidak digunakan lagi", true)]
    public class VariableExample : CommonMath
    {
    }
}
```

Pada contoh di atas dapat dilihat class dengan nama VariableExample dengan access-modifier public. Class VariableExample merupakan turunan dari class CommonMath. Class VariableExample mempunyai attribute seperti yang telah ditulis di atas, fungsi attribute di atas sebagai penanda bahwa class ini sudah tidak bisa digunakan lagi dan akan menghasilkan error bila digunakan.

Setelah class dibuat maka pada kondisi tertentu class tersebut dapat diinstansiasi menjadi sebuah objek.

Berikut adalah contoh class dan cara instansiasi class tersebut menjadi objek pada C# :

```
ClassExample.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Console
{
    public class ClassExample
    {
    }
}
```

Berikut adalah contoh proses instansiasi class ClassExample :

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
    }}
```

```

        ClassExample classExample = new ClassExample();
    }
}
}

```

Method dan Field

Ada dua jenis method, yaitu :

1. Method yang tidak melemparkan nilai biasanya disebut procedure.

Sintaks :

```
[access-modifier] void nama_method(tipe_data parameter-1, ... tipe_data parameter-N)
{
}
```

Contoh :

```
ClassExamle.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Console
{
    public class ClassExample
    {
        public void Tambah(int bilanganA, int bilanganB)
        {
            int hasil = bilanganA + bilanganB;
            System.Console.WriteLine(hasil);
        }
    }
}
```

Contoh penggunaan :

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            ClassExample classExample = new ClassExample();
            classExample.Tambah(3, 10);

            System.Console.ReadKey();
        }
    }
}
```

2. Method yang melemparkan nilai biasanya disebut function.

Sintaks :

```
[access-modifier] tipe_data nama_method(tipe_data parameter-1, ... tipe_data parameter-N)
{
}
```

Contoh :

```

ClassExample.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Console
{
    public class ClassExample
    {
        public int Kurang(int bilanganA, int bilanganB)
        {
            return bilanganA - bilanganB;
        }
    }
}

```

Contoh penggunaan :

```

Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            ClassExample classExample = new ClassExample();
            System.Console.WriteLine(classExample.Kurang(17, 4));

            System.Console.ReadKey();
        }
    }
}

```

Field adalah property pada class seperti pada contoh kode berikut :

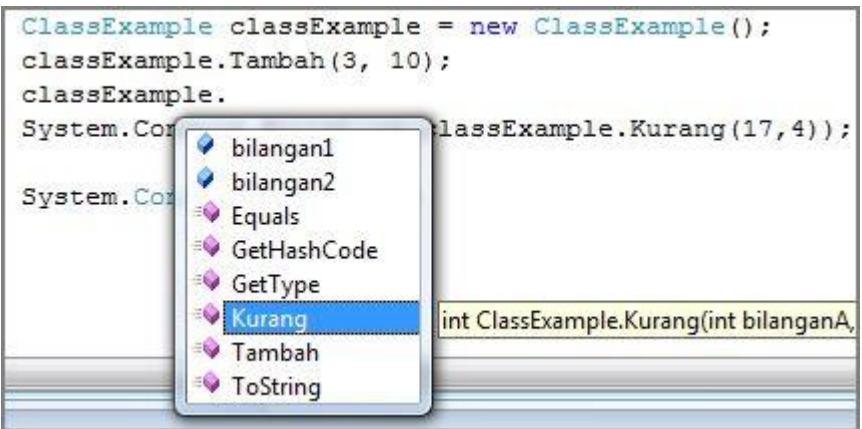
```

ClassExample.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Console
{
    public class ClassExample
    {
        public int bilangan1;
        public int bilangan2;
        private string hasil;
        protected string nama;
    }
}

```

Dari contoh di atas maka field bilangan1 dan bilangan2 dapat diakses secara langsung sebagai method dari objek di dalam class manapun karena menggunakan public sebagai access-modifier.



Sedangkan field nama, karena menggunakan access-modifier protected maka hanya dapat digunakan oleh class yang merupakan class turunan dari ClassExample.

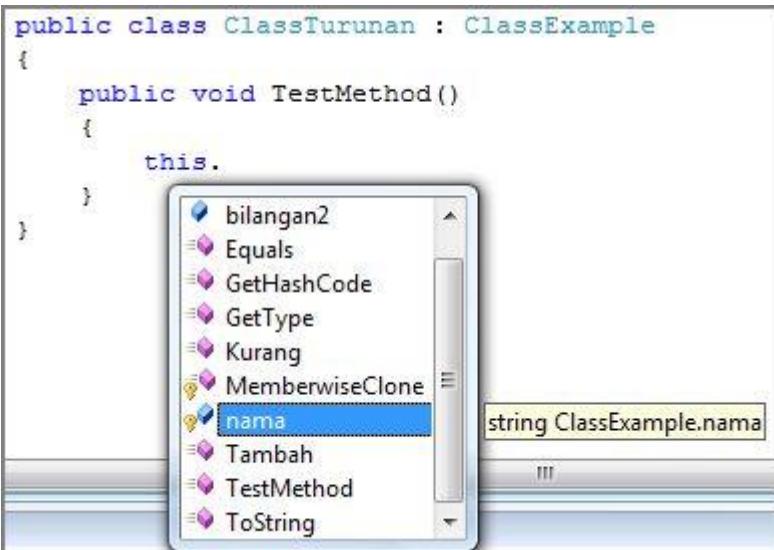
Contoh :

```
ClassTurunan.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using STS.Console;

namespace STS.Console
{
    public class ClassTurunan : ClassExample
    {
        public void TestMethod()
        {
            this.nama = "reza";
        }
    }
}
```

Atau dapat dilihat pada gambar berikut :



Pada class ClassExample terdapat field yang menggunakan access-modifier private. Artinya field ini hanya bisa digunakan di dalam class tersebut.

Field pada suatu class secara default mempunyai access-modifier private dan memang seharusnya tetap seperti itu. Agar class lain dapat melakukan pengambilan (get) atau

pengisian (set) nilai field tersebut maka perlu dibuat public method yang berfungsi untuk itu.

Public method ini dideklarasikan bukan seperti method yang telah dibuat sebelumnya, istilah yang tepat sebagai pengganti nama 'method' untuk fungsi seperti di atas adalah Property.

Cara mendeklarasikan Property adalah sebagai berikut :

```
ClassExample.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Console
{
    public class ClassExample
    {
        public int bilangan1;
        public int bilangan2;
        private string hasil;
        protected string nama;

        public string Hasil
        {
            get { return hasil; }
            set { hasil = value; }
        }
    }
}
```

Dengan cara di atas maka field hasil dapat diberikan nilai (set) dan diambil nilainya (get) oleh semua class.

```
public string Hasil
{
    get { return hasil; }
    set { hasil = value; }
}
```

Kode di atas dapat dipersingkat lagi penulisannya dengan cara berikut :

```
public string Hasil
{
    get;
    set;
}
```

Static Member

Pada bagian sebelumnya telah dijelaskan cara membuat class, method dan field. Pada contoh di atas terlebih dahulu harus dibuat instance dari class agar dapat mengakses method atau field.

Pada bagian ini akan dijelaskan cara membuat method agar dapat digunakan tanpa harus melakukan instansiasi class terlebih dahulu.

Caranya sangat mudah yaitu dengan menambahkan kata kunci static pada method yang diinginkan seperti pada contoh berikut :

```
ClassExample.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Console
```

```
{
    public class ClassExample
    {

        public static void Kali(int bilanganA, int bilanganB)
        {
            int hasil = bilanganA * bilanganB;
            System.Console.WriteLine(hasil);
        }
    }
}
```

Untuk menggunakan method ini cukup dengan menulis kode seperti berikut ini :

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        ClassExample.Kali(3, 10);

        System.Console.ReadKey();

    }
}
```

Inheritance dan Polymorphism

Inheritance atau penurunan adalah kemampuan membuat suatu class anak dari suatu class induk, dimana seluruh field dan method diturunkan kepada class anak, kecuali method atau field dengan access-modifier private.

Contoh kasus inheritance ini adalah seperti kode berikut :

```
ClassTurunan.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using STS.Console;

namespace STS.Console
{
    public class ClassTurunan : ClassExample
    {
        public void TestMethod()
        {
            this.nama = "reza";
        }
    }
}
```

Pada kode di atas, yang menjadi class induk adalah ClassExample dan ClassTurunan merupakan class anaknya. Baris berikut menyatakan cara membuat class ClassTurunan sebagai turunan dari class ClassExample.

```
public class ClassTurunan : ClassExample
```

Polymorphism bila diartikan dengan melihat arti dari kata penyusunnya yaitu poly yang berarti “banyak dan morph yang berarti “bentuk” maka secara keseluruhan artinya adalah “banyak bentuk”.

Masih berhubungan dengan inheritance atau penurunan, misalkan dibuat class yang akan menjadi class induk seperti berikut :

```
CurveMath.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Common.Math
{
    public class CurveMath
    {
        private int panjang;
        private int lebar;
        private int jejari;

        public int Panjang
        {
            set {panjang = value; }
            get { return panjang; }
        }

        public int Lebar
        {
            set { lebar = value; }
            get { return lebar; }
        }

        public int Jejari
        {
            set { jejari = value; }
            get { return jejari; }
        }

        public virtual void HitungLuas()
        {
            System.Console.WriteLine(Panjang * Lebar);
        }
    }
}
```

Pada kode di atas dapat dilihat terdapat kata kunci virtual pada method berikut :

```
public virtual void HitungLuas()
{
    System.Console.WriteLine(Panjang * Lebar);
}
```

Artinya method ini dapat diubah menjadi bentuk lagi dengan cara di-override. Seperti yang dilakukan oleh class Lingkaran.

```
Lingkaran.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using STS.Common.Math;

namespace STS.Console
{
    public class Lingkaran : CurveMath
    {
        public override void HitungLuas()
        {
            System.Console.WriteLine(3.14 * Jejari * Jejari);
        }
    }
}
```

```
}
```

Pada kode di atas dapat dilihat digunakan kata kunci override, kata kunci ini hanya dapat digunakan kepada method yang menggunakan kata kunci virtual.

Method Overloading

Method overloading, untuk mengetahui arti dari istilah ini maka bisa melihat terlebih dahulu baris kode di bawah ini :

```
Lingkaran.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

using STS.Common.Math;

namespace STS.Console
{
    public class Lingkaran : CurveMath
    {
        public override void HitungLuas()
        {
            System.Console.WriteLine(3.14 * Jejari * Jejari);
        }

        /// <summary>
        /// Tidak ada parameter
        /// </summary>
        public void Tambah()
        {
            System.Console.WriteLine(3 + 10);
        }

        /// <summary>
        /// Terdapat satu parameter
        /// </summary>
        public void Tambah(int bilangan)
        {
            System.Console.WriteLine(3 + bilangan);
        }

        /// <summary>
        /// Terdapat dua parameter dengan tipe data int
        /// </summary>
        public void Tambah(int bilangan1, int bilangan2)
        {
            System.Console.WriteLine(bilangan1 + bilangan2);
        }

        /// <summary>
        /// Terdapat dua parameter dengan tipe data double
        /// </summary>
        public void Tambah(double bilangan1, double bilangan2)
        {
            System.Console.WriteLine(bilangan1 + bilangan2);
        }
    }
}
```

Pada kode di atas dapat dilihat terdapat tiga method dengan nama yang sama yaitu Tambah perbedaan tiap method adalah dilihat dari jumlah parameter dan tipe data yang digunakan.

Contoh kode di atas adalah praktik dari Method Overloading. Pada Visual Studio saat ingin menambahkan method Tambah maka dapat dilihat tampilan seperti pada gambar berikut :

```
Lingkaran lingkaran = new Lingkaran();
lingkaran.Jejari = 200;
lingkaran.HitungLuas();
lingkaran.Tambah()
▲ 1 of 4 ▾ void Lingkaran.Tambah()
Tidak ada parameter
```

Collection

Array

Array adalah suatu struktur data yang dapat menyimpan data dengan tipe yang sama dan diakses dengan menggunakan suatu indeks yang menunjukkan suatu elemen didalam array tersebut.

Variabel yang disimpan di dalam array ini disebut juga dengan elemen array dan tipe datanya disebut dengan tipe elemen dari array.

Mendeklarasikan Array

Adapun sintaks yang digunakan untuk mendeklarasikan array adalah :

```
type_data[] namaVariable;
```

Contoh :

```
string[] namaBulan;
```

Inisialisasi Array

Untuk memberikan nilai ke dalam suatu variabel array dapat dilakukan dengan beberapa cara. Pertama dengan menggunakan kata kunci new untuk menentukan jumlah elemen yang dapat disimpan oleh array dilanjutkan dengan memberi nilai masing-masing elemen.

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
```

```

    static void Main(string[] args)
    {
        string[] namaBulan = new string[12];
        namaBulan[0] = "Januari";
        namaBulan[1] = "Februari";
    }
}

```

Cara kedua dengan cara langsung memberi nilai pada saat deklarasi seperti contoh berikut :

```

Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] namaBulan = { "Januari", "Februari" };
        }
    }
}

```

Berikut ini adalah contoh program untuk mengakses array.

```

Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] namaBulan = { "Januari", "Februari", "Maret" };

            for (int i = 0; i < namaBulan.Length; i++)
            {
                System.Console.WriteLine(namaBulan[i]);
            }

            System.Console.ReadKey();
        }
    }
}

```

Collection

Collection berfungsi sebagai kontainer untuk menyimpan sekumpulan objek. Collection mirip dengan konsep array. Untuk mengakses data dalam collection biasanya dilakukan dengan melakukan iterasi terhadap data yang ada dalam collection atau dapat juga diakses dengan menggunakan indeks atau indexers.

ArrayList

Berikut adalah contoh penggunaan ArrayList.

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;
using System.Collections;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            ArrayList al = new ArrayList();

            Pegawai p1 = new Pegawai();
            p1.Nama = "Reza";
            p1.Umur = 30;
            al.Add(p1);

            Pegawai p2 = new Pegawai();
            p2.Nama = "Faisal";
            p2.Umur = 25;
            al.Add(p2);

            System.Console.WriteLine("Jumlah objek : " + al.Count);

            for (int i = 0; i < al.Count; i++)
            {
                int j = i + 1;
                Pegawai pegawai = (Pegawai)al[i];
                System.Console.WriteLine(j + ". " + pegawai>Nama
                +" ("+pegawai.Umur+")");
            }

            System.Console.ReadKey();
        }
    }
}
```

Hashtable

Berbeda dengan ArrayList, Hashtable mempunyai key. Berikut contoh pemanfaatan Hashtable.

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;
using System.Collections;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            Hashtable ht = new Hashtable();

            Pegawai p1 = new Pegawai();
            p1.Nama = "Reza";
            p1.Umur = 30;
            ht.Add("P1", p1);

            Pegawai p2 = new Pegawai();
```

```
p2.Nama = "Faisal";
p2.Umur = 25;
ht.Add("P2", p2);

System.Console.WriteLine("Jumlah objek : " + ht.Count);

IDictionaryEnumerator enumerator = ht.Getenumerator();
while (enumerator.MoveNext())
{
    Pegawai pegawai = (Pegawai)enumerator.Value;

    System.Console.WriteLine(enumerator.Key.ToString() + " " +
pegawai.Nama
+ "(" + pegawai.Umur + ")");
}

System.Console.ReadKey();

}
}
```

Referensi

- C# in a Nutshell, O'Reilly.
- Pengenalan Bahasa C#, Project Otak.
- Programming C# 3.0, O'Reilly.

4

ASP.NET

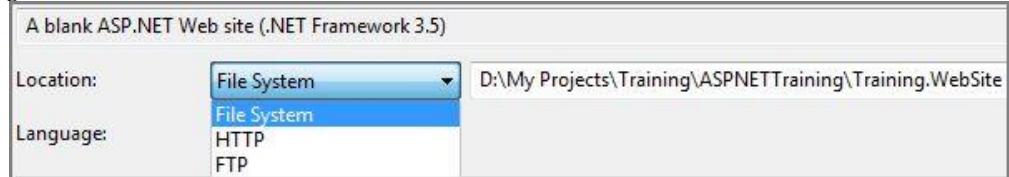
Web Site dan Web Application

Untuk membangun aplikasi web dengan ASP.NET, Visual Studio memberikan pilihan template yang dapat digunakan yaitu :

1. Web Site.
2. Web Application.

Berikut adalah hal-hal yang membedakan antara Web Site dan Web Application :

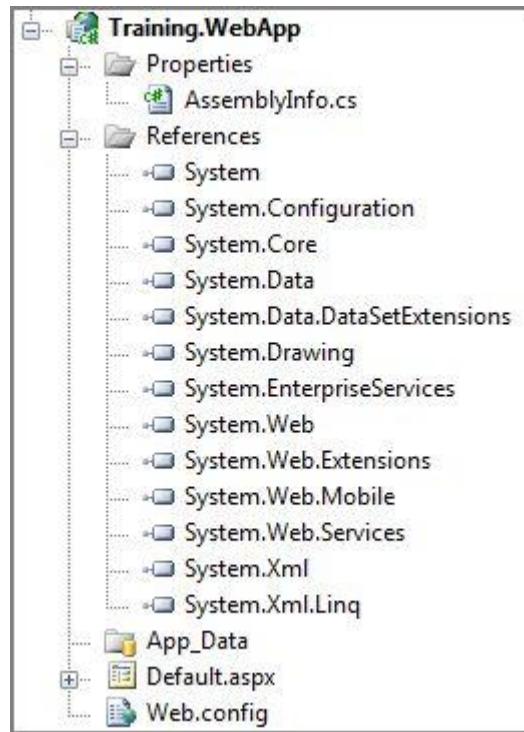
1. Lokasi penyimpanan, Web Application hanya mempunyai 1 tempat penyimpanan file-filenya yaitu pada File System. Sedangkan Web Site terdapat 3 cara untuk mengakses file-filenya yaitu File System, HTTP dan FTP, seperti dilihat pada gambar berikut.



2. Struktur Template, berikut ini adalah struktur default dari template Web Site.

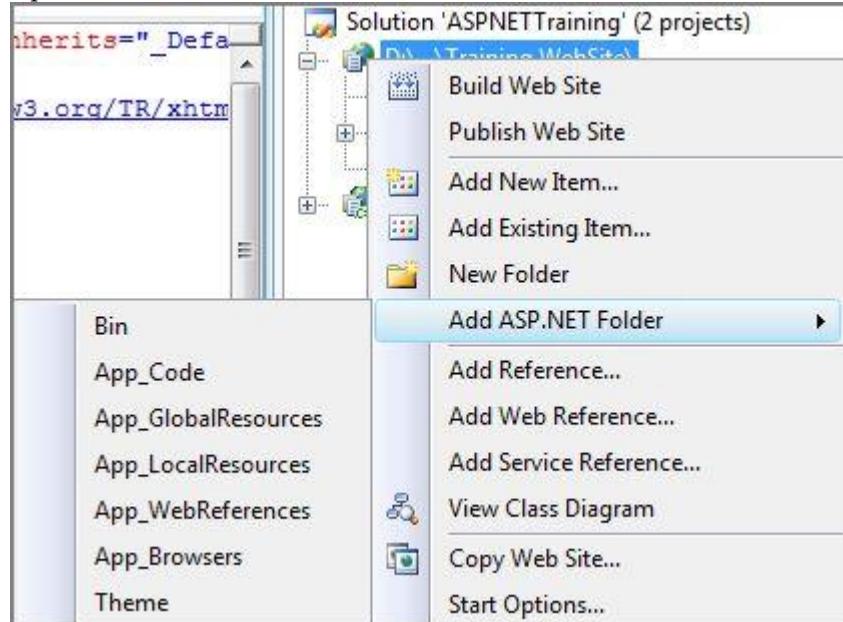


Sedangkan berikut adalah gambar struktur default dari template Web Application.

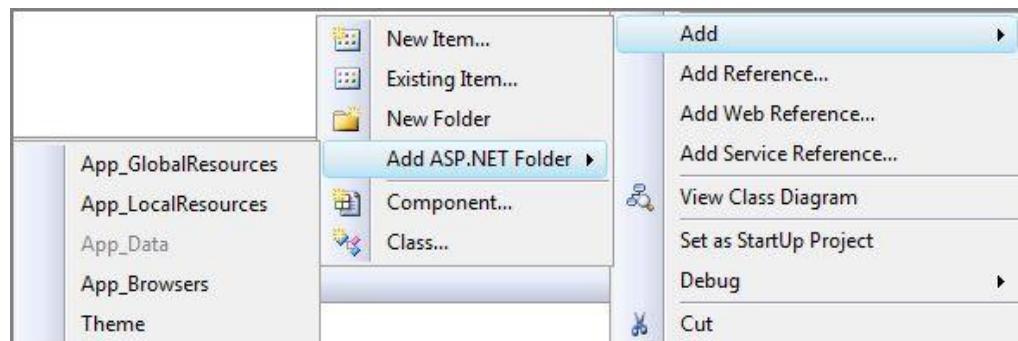


Dari gambar di atas perbedaan yang mencolok dilihat pada template Web Application terdapat Properties dan Reference.

3. ASP.NET Folder, folder ASP.NET yang dapat ditambahkan pada Web Site adalah seperti berikut.



Sedangkan folder ASP.NET yang dapat ditambahkan pada Web Application adalah sebagai berikut :



Dapat dilihat pada Web Application tidak ada fasilitas untuk menambahkan direktori Bin dengan menggunakan Visual Studio. Tetapi bila diperhatikan pada direktori dimana Web Application disimpan akan dilihat folder dan file-file seperti berikut.

App_Data	7/1/2008 2:05 PM	File Folder	
bin	7/1/2008 2:05 PM	File Folder	
obj	7/1/2008 2:05 PM	File Folder	
Properties	7/1/2008 2:05 PM	File Folder	
Default.aspx	7/1/2008 2:05 PM	ASP.NET Server Pa...	1 KB
Default.aspx.cs	7/1/2008 2:05 PM	Visual C# Source f...	1 KB
Default.aspx.designer.cs	7/1/2008 2:05 PM	Visual C# Source f...	1 KB
Training.WebApp.csproj	7/1/2008 3:46 AM	Visual C# Project f...	5 KB
Training.WebApp.cspr...	7/1/2008 3:46 AM	Visual Studio Proj...	2 KB
Web.config	7/1/2008 3:46 AM	XML Configuratio...	8 KB

Fungsi dari folder ASP.NET ini akan dijelaskan lebih lanjut dibagian selanjutnya.

- Eksekusi Project, pada Web Site untuk mengeksekusi dan untuk melihat hasilnya pada browser hanya terdapat 1 cara yaitu dengan klik kanan pada Web Site dan pilih View in browser. Sedangkan pada Web Application selain dengan cara tersebut juga dapat dilakukan dengan klik kanan pada project, pilih Debug > Start new instance.

Bahasan selanjutnya pada buku digital ini akan menggunakan Web Site sebagai template yang digunakan untuk membuat aplikasi web.

Struktur Web Site

Pada bagian ini akan disebutkan file-file dan folder-folder penting pada Web Site beserta fungsi dan kegunaanya.

File Konfigurasi

Berikut adalah file setting dan konfigurasi untuk aplikasi web.

Web.config

File ini merupakan file yang sangat penting pada suatu aplikasi web yang dibangun dengan ASP.NET. File ini berfungsi sebagai file utama setting dan konfigurasi. File ini menggunakan format XML. Setting atau konfigurasi yang disimpan oleh file ini diantaranya adalah :

- Connection String, konfigurasi untuk melakukan koneksi ke database server.
- Application Setting, adalah konstanta yang dapat digunakan pada aplikasi web.

3. Security Configuration.
4. Session State Configuration.
5. Control Module Loading.
6. Compilation Setting.
7. Application Language.
8. Dan lain-lain.

Berikut adalah sebagian contoh isi dari file Web.config:

Web.config

```
<?xml version="1.0"?>
<configuration>
    ...
    <appSettings>
        <add key="DefaultPageSize" value="13"/>
    </appSettings>
    <connectionStrings>
        <add name="SQLConnString"
            connectionString="Persist Security Info=False;
                            Integrated Security=false;
                            User ID=sa;Password=rahasia;
                            Initial Catalog=NamaDatabase;
                            server=localhost;
                            Connect Timeout=30"
            providerName="System.Data.SqlClient"/>
    </connectionStrings>
    ...
</configuration>
```

Pada contoh di atas bisa dilihat file Web.config menyimpan konfigurasi untuk koneksi ke database server.

Tipe File ASP.NET

Berikut ini adalah tipe file yang umum digunakan pada ASP.NET, yaitu :

1. *.master adalah tipe file yang berperan sebagai MasterPage. File ini berfungsi sebagai template halaman aplikasi web agar mempunyai antarmuka yang seragam.
2. *.aspx adalah file web form.
3. *.ascx adalah file web user control.
4. *.asmx adalah file untuk keperluan web service.
5. *.resx adalah file resource yang dapat digunakan untuk mendukung globalization.
6. Dan lain-lain.

Fungsi detail dan contoh masing-masing tipe file tersebut akan diberikan pada bagian selanjutnya.

Folder ASP.NET

Berikut ini adalah folder-folder yang mempunyai fungsi khusus bagi ASP.NET, yaitu :

1. Bin, direktori ini menyimpan DLL yang digunakan pada aplikasi web. Direktori ini juga berfungsi menyimpan file DLL hasil aplikasi web yang di-publish.
2. App_Data, menyimpan data seperti file MDF milik SQL Express, XML dan lain-lain.
3. App_Code, untuk menyimpan file class, dataset dan lain-lain. File-file yang disimpan pada folder ini akan secara otomatis di kompilasi atau di transform ke bentuk lain sesuai dengan tipe file tersebut.
4. App_Theme, folder untuk menyimpan file ber-extension *.skin dan *.css.
5. App_GlobalResources, file untuk menyimpan file *.resx, data pada file *.resx pada direktori ini dapat digunakan pada seluruh halaman Web Site.
6. App_Browsers, file ini berfungsi untuk menyimpan file dengan extension *.browser. file *.browser mempunyai format XML yang berfungsi untuk mengidentifikasi browser yang mengakses halaman pada Web Site.
7. App_LocalResources, file untuk menyimpan file *.resx, data pada file *.resx pada direktori ini hanya dapat digunakan pada 1 halaman Web Site yang spesifik.
8. App_WebReferences, direktori ini menyimpan file *.wsdl yang menjadi referensi web service yang akan dikonsumsi oleh aplikasi web.

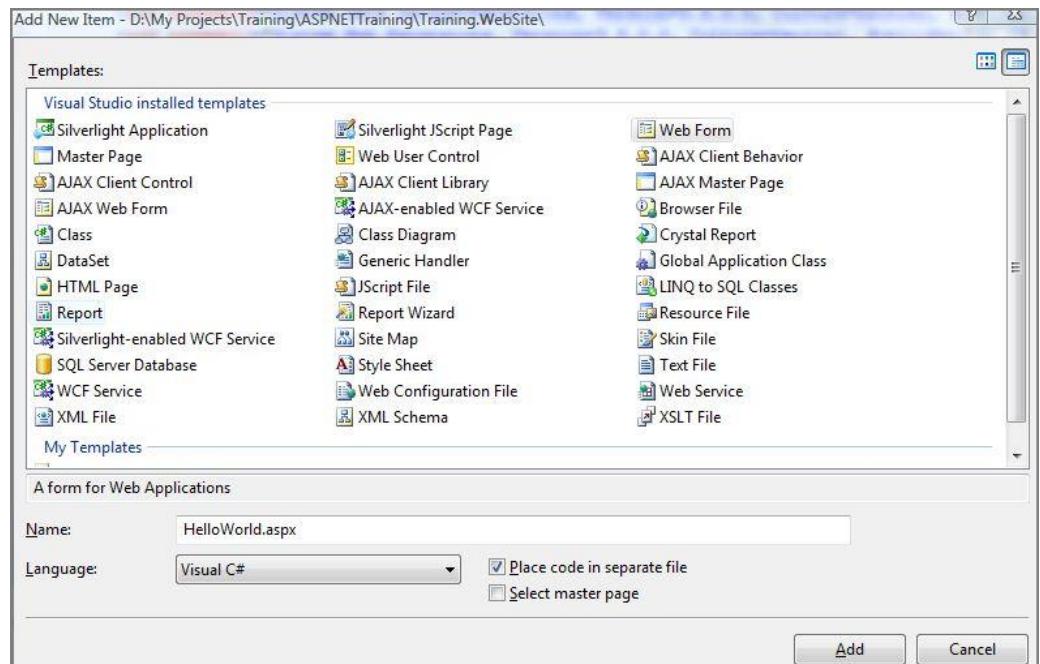
Fungsi detail dan contoh penggunaan direktori ini akan dijelaskan pada bagian selanjutnya.

Hello World with ASP.NET

Sebelum mengetahui lebih lanjut tentang ASP.NET ada baiknya terlebih dahulu berkenalan dengan ASP.NET dengan cara membuat halaman sederhana.

Berikut adalah langkah-langkah yang mesti diikuti :

1. Buat Solution dengan nama ASPNETTraining.
2. Tambahkan Web Site pada Solution.
3. Klik kanan pada Web Site dan pilih Add New Item.
4. Pada window Add New Item pilih Web Form pada kolom Templates. Pada kolom Name beri nama file dengan HelloWorld.aspx.
Biarkan checkbox Place code in separate file dicek, maka selain HelloWorld.aspx nanti akan dapat dilihat ada file lain dengan nama HelloWorld.aspx.cs, bila language yang dipilih adalah Visual Basic maka nama file tersebut adalah HelloWorld.aspx.vb.



Pada Web Site dapat dilihat tambahan file HelloWorld.aspx dan HelloWorld.aspx.cs seperti berikut.



Dan berikut adalah isi masing-masing file tersebut.

```
HelloWorld.aspx
<%@ Page Language="C#"
    AutoEventWireup="true"
    CodeFile="HelloWorld.aspx.cs"
    Inherits="HelloWorld" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

        </div>
    </form>
</body>
</html>
```

Dan berikut adalah isi dari file HelloWorld.aspx.cs

```
HelloWorld.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
```

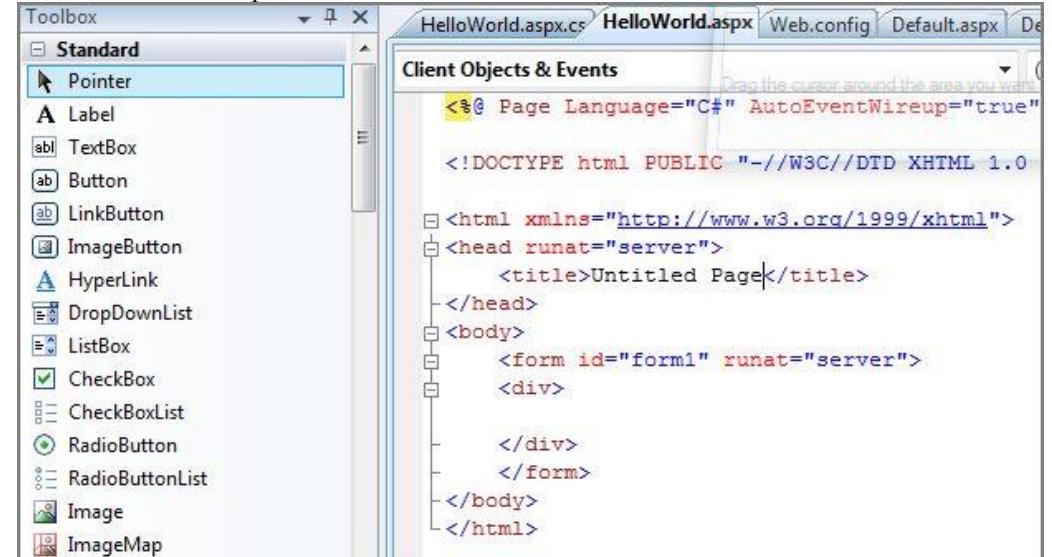
```

using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class HelloWorld : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
}

```

5. Pada HelloWorld.aspx tambahkan control Label, TextBox dan Button.



Sehingga kode HelloWorld.aspx menjadi seperti berikut :

```

HelloWorld.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="HelloWorld.aspx.cs"
   Inherits="HelloWorld" %>

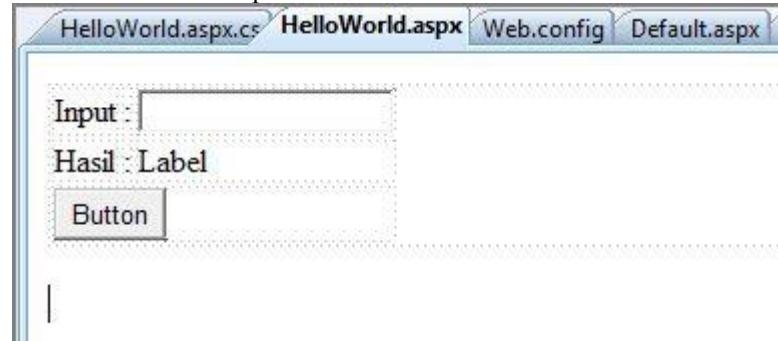
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table>
                <tr>
                    <td>Input</td>
                    <td>:</td>
                    <td><asp:TextBox ID="TextBox1" runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td>Hasil</td>
                    <td>:</td>
                    <td><asp:Label ID="Label1" runat="server" Text="Label"></asp:Label></td>
                </tr>
                <tr>
                    <td colspan="3">
                        <asp:Button ID="Button1" runat="server" Text="Button" />
                    </td>
                </tr>
            </table>
        </div>
    </form>
</body>

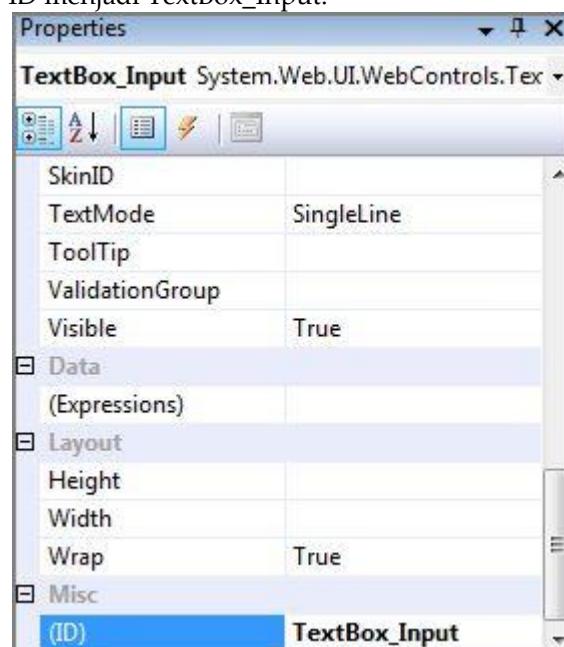
```

```
</html>
```

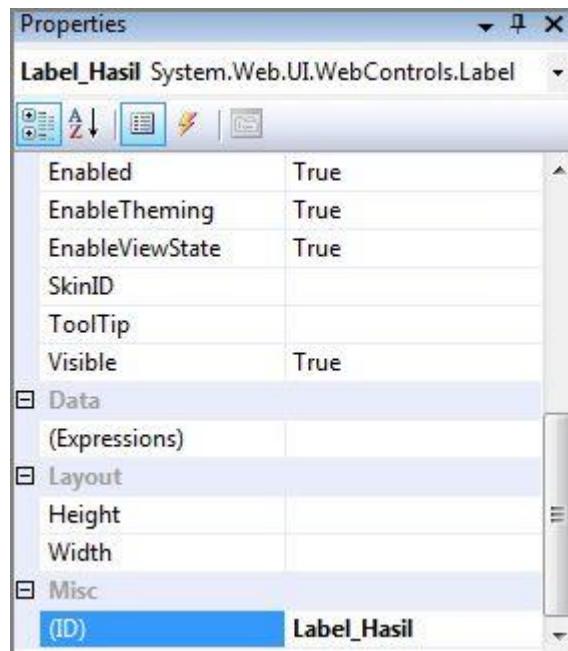
Hasilnya dapat dilihat dengan mengaktifkan mode Design pada editor, dan akan dilihat antarmuka seperti berikut.



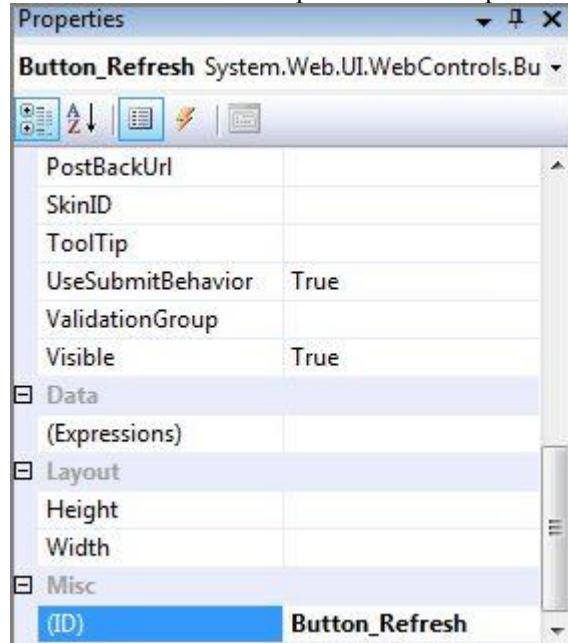
6. Pada mode Design ini, klik pada control TextBox dan pada kolom Properties ubah ID menjadi TextBox_Input.



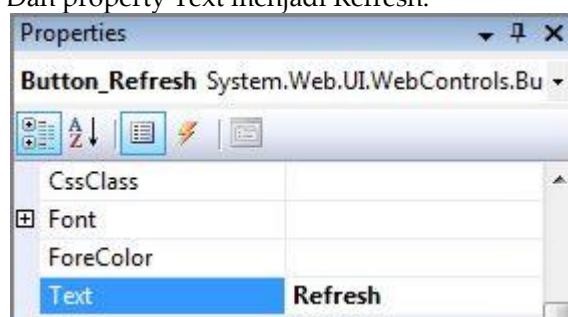
Klik control Label dan pada kolom Properties ubah ID menjadi Label_Hasil.



Pilih control Button dan pada kolom Properties ubah ID menjadi Button_Refresh.



Dan property Text menjadi Refresh.



Maka hasilnya pada mode Design adalah seperti berikut :



Dan pada kode file HelloWorld.aspx akan mengalami perubahan seperti berikut :

HelloWorld.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="HelloWorld.aspx.cs"
Inherits="HelloWorld" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<table>
    <tr>
        <td>Input</td>
        <td>:</td>
        <td><asp:TextBox ID="TextBox_Input" runat="server"></asp:TextBox></td>
    </tr>
    <tr>
        <td>Hasil</td>
        <td>:</td>
        <td>
            <asp:Label ID="Label_Hasil" runat="server"
Text="Label"></asp:Label>
        </td>
    </tr>
    <tr>
        <td colspan="3" style="text-align: center;">
            <asp:Button ID="Button_Refresh" runat="server" Text="Refresh" />
        </td>
    </tr>
</table>
</div>
</form>
</body>
</html>
```

- Untuk melihat hasilnya pada web browser, pada Solution Explorer klik kanan pada file HelloWorld.aspx dan pilih View in browser. Ini bukan langkah terakhir tetapi hanya untuk melihat antarmuka yang telah dibuat.
- Langkah selanjutnya buka file HelloWorld.aspx.cs dan pada method Page_Load tambahkan baris sehingga isi file HelloWorld.aspx.cs menjadi seperti berikut :

HelloWorld.aspx.cs

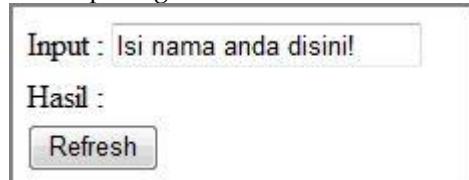
```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
```

```

public partial class HelloWorld : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            TextBox_Input.Text = "Isi nama anda disini!";
            Label_Hasil.Text = String.Empty;
        }
    }
}

```

Eksekusi HelloWorld.aspx untuk di tampilkan pada web browser. Hasilnya dapat dilihat pada gambar berikut.



Dapat dilihat pengaruh baris kode yang berada pada method Page_Load pada halaman ASP.NET di atas. Property dari control TextBox dan Label diubah secara pemrograman dengan cara di atas.

- Kembali ke mode Design, klik control Button dan pada kolom Properties klik tombol Events dan klik double pada Action > Click.

Hasilnya pada kolom Properties akan menjadi seperti berikut.



Dan pada file HelloWorld.aspx.cs akan ditambahkan baris berikut :

```

HelloWorld.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class HelloWorld : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            TextBox_Input.Text = "Isi nama anda disini!";
            Label_Hasil.Text = String.Empty;
        }
    }

    protected void Button_Refresh_Click(object sender, EventArgs e)
    {
        Label_Hasil.Text = "Hello World with ASP.NET from " +
TextBox_Input.Text;
    }
}

```

Dan pada halaman HelloWorld.aspx akan dilihat terdapat penambahan property onclick pada control Button seperti berikut :

```
<asp:Button ID="Button_Refresh" runat="server" Text="Refresh"  
onclick="Button Refresh Click" />
```

10. Tambahkan baris kode pada body method Button_Refresh_Click agar menjadi seperti berikut.

```
protected void Button_Refresh_Click(object sender, EventArgs e)  
{  
    Label_Hasil.Text = "Hello World with ASP.NET from " + TextBox_Input.Text;  
}
```

11. Eksekusi halaman HelloWorld.aspx, pada halaman HelloWorld.aspx yang ditampilkan pada web browser isikan nama pada TextBox dan klik tombol hasilnya dapat dilihat pada gambar berikut.

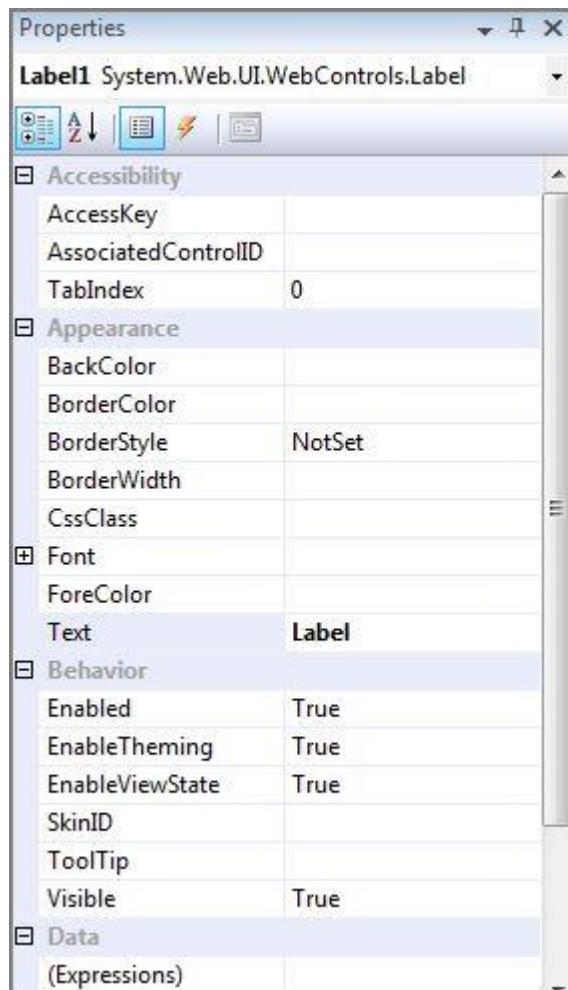
The screenshot shows a simple ASP.NET page. At the top, there is a text input field containing the text "Reza". Below it is a label displaying the text "Hasil : Hello World with ASP.NET from Reza". At the bottom, there is a blue "Refresh" button.

Standard Server Control

Pada bagian ini akan dipraktekan penggunaan server control standar sering digunakan.

Label

Control ini mempunyai Property tetapi tidak mempunyai event. Berikut adalah sebagian Property yang dimiliki oleh control Label.



Fungsi control ini adalah untuk menampilkan text yang ingin ditampilkan pada halaman web. Dengan menggunakan control Label, text yang ditampilkan dapat ditulis secara programmatically dari CodeFile/CodeBehind. Bukan hanya text yang akan ditampilkan tetapi property juga dapat diubah secara programmatically.

Contoh :

```
LabelExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="LabelExample.aspx.cs"
Inherits="StandarServerControl_LabelExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label_TodayDate" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Dan berikut isi file LabelExample.aspx.cs

```
LabelExample.aspx.cs
using System;
using System.Collections;
```

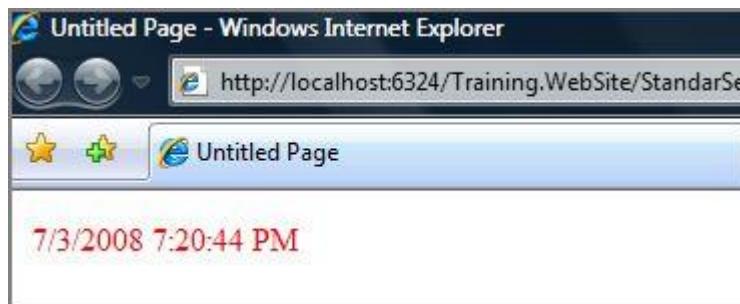
```

using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_LabelExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label_TodayDate.Text = DateTime.Now.ToString();
        Label_TodayDate.ToolTip = "Tanggal hari ini";
        Label_TodayDate.ForeColor = System.Drawing.Color.Red;
    }
}

```

Hasilnya dapat dilihat pada gambar berikut.



TextBox

TextBox mempunyai property yang dapat dilihat pada kolom Properties. TextBox mempunyai satu event yang dapat digunakan yaitu TextChanged, event ini akan dipanggil ketika nilai pada TextBox berubah.

Contoh :

```

TextBoxExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="TextBoxExample.aspx.cs"
Inherits="StandarServerControl_TextBoxExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server" Text="Tulis nama Anda"></asp:Label> :
            <asp:TextBox ID="TextBox1" runat="server" AutoPostBack="true"
                ontextchanged="TextBox1_TextChanged">
            </asp:TextBox>
            <br /><br />
            <asp:Label ID="Label_Hasil" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>

```

Code File :

```

TextBoxExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_TextBoxExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Label_Hasil.Text = String.Empty;
        }
    }

    protected void TextBox1_TextChanged(object sender, EventArgs e)
    {
        Label_Hasil.Text = "Hallo " + TextBox1.Text + ", apa kabar?";
    }
}

```

Button

Ada tiga jenis tombol yang dapat digunakan, yaitu :

1. Button.
2. LinkButton.
3. ImageButton.

Event yang dimiliki oleh control ini adalah :

1. Click.
2. Command.

Contoh :

```

ButtonExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ButtonExample.aspx.cs"
Inherits="StandarServerControl_ButtonExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
            <br />
            <asp:Button ID="Button1" runat="server" Text="Button 1"
                oncommand="Button1_Command" />
            <asp:Button ID="Button2" runat="server" Text="Button 2"
                onclick="Button2_Click" />
        </div>
    </form>
</body>
</html>

```

Code file :

```
ButtonExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_ButtonExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Command(object sender, CommandEventArgs e)
    {
        Label1.Text = DateTime.Now.ToString() + " from Command event";
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString() + " from Click event ";
    }
}
```

Contoh dengan menggunakan LinkButton.

```
LinkButtonExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="LinkButton.aspx.cs"
Inherits="StandarServerControl_LinkButton" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="TextBox_Bilangan1" runat="server"></asp:TextBox>
            +
            <asp:TextBox ID="TextBox_Bilangan2" runat="server"></asp:TextBox>
            =
            <asp:TextBox ID="TextBox_Hasil" runat="server"></asp:TextBox>
            <br />
            <asp:Label ID="Label_Status" runat="server" Text="Label"></asp:Label>
            <br />
            <asp:LinkButton ID="LinkButton_Jumlah" runat="server"
                onclick="LinkButton_Jumlah_Click">Klik Saya
            </asp:LinkButton>
        </div>
    </form>
</body>
</html>
```

Code file :

```
LinkButtonExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
```

```

using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_LinkButton : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label_Status.Text = String.Empty;
    }

    protected void LinkButton_Jumlah_Click(object sender, EventArgs e)
    {
        if (String.IsNullOrEmpty(TextBox_Bilangan1.Text) ||
            String.IsNullOrEmpty(TextBox_Bilangan1.Text))
        {
            Label_Status.Text = "Kolom bilangan tidak boleh kosong";
        }
        else
        {
            int bilangan1 = Convert.ToInt32(TextBox_Bilangan1.Text);
            int bilangan2 = Convert.ToInt32(TextBox_Bilangan2.Text);
            int hasil = bilangan1 + bilangan2;
            TextBox_Hasil.Text = hasil.ToString();
        }
    }
}

```

Contoh penggunaan ImageButton.

```

ImageButtonExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ImageButtonExample.aspx.cs"
Inherits="StandarServerControl_ImageButtonExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label_Today" runat="server" Text="Label"></asp:Label>
            <br />
            <asp:ImageButton ID="ImageButton_ShowDate"
                ImageUrl("~/Images/icon_button.gif"
                runat="server"
                onclick="ImageButton_ShowDate_Click" />
        </div>
    </form>
</body>
</html>

```

Code file :

```

ImageButtonExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_ImageButtonExample : System.Web.UI.Page

```

```
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void ImageButton_ShowDate_Click(object sender, ImageClickEventArgs e)
    {
        Label_Today.Text = DateTime.Now.ToString();
    }
}
```

HyperLink

Control ini berfungsi seperti layaknya hyperlink yang dengan menggunakan tag <a> pada HTML. Control ini hanya memungkinkan data yang dipilih hanya satu.

Contoh :

```
HyperLinkExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="HyperLinkExample.aspx.cs"
Inherits="StandarServerControl_HyperLinkExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:HyperLink ID="HyperLink_GoTo"
                NavigateUrl("~/StandarServerControl/ImageButtonExample.aspx"
                runat="server">ImageButtonExample.aspx
            </asp:HyperLink>
        </div>
    </form>
</body>
</html>
```

DropDownList

Akan dibuat antarmuka seperti berikut :

Contoh :

```
DropDownListExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DropDownListExample.aspx.cs"
Inherits="StandarServerControl_DropDownListExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table>
                <tr>
                    <td>Provinsi</td>
                    <td>
                        <asp:DropDownList ID="DropDownList_Provinsi" AutoPostBack="true"
                            runat="server">
                        </asp:DropDownList>
                    </td>
                </tr>
            </table>
        </div>
    </form>
</body>
</html>
```

```

        runat="server"

onselectedindexchanged="DropDownList_Provinsi_SelectedIndexChanged">
    <asp:ListItem Value="">Pilih Provinsi</asp:ListItem>
    <asp:ListItem Value="1">Jawa Barat</asp:ListItem>
    <asp:ListItem Value="2">Kalimantan Selatan</asp:ListItem>

```

Code file :

```

DropDownListExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_DropDownListExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void DropDownList_Provinsi_SelectedIndexChanged(object sender, EventArgs e)
    {
        ListItemCollection items = new ListItemCollection();

        ListItem item = new ListItem();
        item.Text = "Pilih Kabupaten/Kota";
        item.Value = "";
        items.Add(item);

        if (!String.IsNullOrEmpty(DropDownList_Provinsi.SelectedValue))
        {
            if (DropDownList_Provinsi.SelectedValue.Equals("1"))
            {
                item = new ListItem();
                item.Text = "Kota Bandung";
                item.Value = "1_1";
                items.Add(item);

                item = new ListItem();
                item.Text = "Kabupaten Bandung";
                item.Value = "1_2";
                items.Add(item);
            }
        }
    }
}

```

```

        item = new ListItem();
        item.Text = "Cimahi";
        item.Value = "1_3";
        items.Add(item);
    }
    else if (DropDownList_Provinsi.SelectedValue.Equals("2"))
    {
        item = new ListItem();
        item.Text = "Banjarmasin";
        item.Value = "2_1";
        items.Add(item);

        item = new ListItem();
        item.Text = "Banjarbaru";
        item.Value = "2_2";
        items.Add(item);

        item = new ListItem();
        item.Text = "Kandangan";
        item.Value = "2_3";
        items.Add(item);
    }
}

DropDownList_KabKota.DataSource = items;
DropDownList_KabKota.DataTextField = "Text";
DropDownList_KabKota.DataValueField = "Value";
DropDownList_KabKota.DataBind();
}
}

```

ListBox

Berbeda dengan DropDownList yang hanya bisa memilih satu nilai saja, ListBox memungkinkan nilai yang dipilih lebih dari satu.

Contoh :

```

ListBoxExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ListBoxExample.aspx.cs"
Inherits="StandarServerControl_ListBoxExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Language Programming :
            <asp:TextBox ID="TextBox_Language" runat="server"></asp:TextBox>
            <asp:Button ID="Button_Add" runat="server" Text="Add"
                onclick="Button_Add_Click" />
            <br /><br />
            <asp:ListBox ID="ListBox_Skill" Width="200px" Height="200px"
                SelectionMode="Multiple" runat="server">
                <asp:ListItem>C#</asp:ListItem>
                <asp:ListItem>VB.NET</asp:ListItem>
                <asp:ListItem>PHP</asp:ListItem>
                <asp:ListItem>Java</asp:ListItem>
            </asp:ListBox>
            <br /><br />
            <asp:Button ID="Button_Show" runat="server" Text="Show"
                onclick="Button_Show_Click" />
            <asp:Button ID="Button_Refresh" runat="server" Text="Refresh"
                onclick="Button_Refresh_Click" />
            <br />
            <asp:Label ID="Label-LanguageList" runat="server" Text="Label"></asp:Label>
        </div>
    </form>

```

```

</form>
</body>
</html>

```

Code file :

```

ListBoxExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_ListBoxExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label_LanguageList.Text = String.Empty;
    }

    protected void Button_Add_Click(object sender, EventArgs e)
    {
        if (!String.IsNullOrEmpty(TextBox_Language.Text))
        {
            ListBox_Skill.Items.Add(TextBox_Language.Text);
            TextBox_Language.Text = String.Empty;
        }
    }

    protected void Button_Show_Click(object sender, EventArgs e)
    {
        Label_LanguageList.Text = "Bahasa yang dipilih : <br/>";
        foreach (ListItem li in ListBox_Skill.Items)
        {
            if (li.Selected == true)
            {
                Label_LanguageList.Text += li.Text + "<br/>";
            }
        }
    }

    protected void Button_Refresh_Click(object sender, EventArgs e)
    {
        Response.Redirect(Request.Path);
    }
}

```

CheckBox

Berikut adalah contoh penggunaan control CheckBox.

```

CheckBoxExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CheckBoxExample.aspx.cs"
Inherits="StandarServerControl_CheckBoxExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

```

```

<asp:CheckBox ID="CheckBox_Status"
    Text="Centak cek box bila anda mengusai Algoritma"
    AutoPostBack="true"
    TextAlign="Left"
    runat="server"
    oncheckedchanged="CheckBox_Status_CheckedChanged" />
<br />
<br />
<asp:CheckBox ID="CheckBox_IsProgrammer"
    Text="Programmer" runat="server" />
<br />
<asp:CheckBox ID="CheckBox_NotProgrammer"
    Text="Bukan Programmer" runat="server" />
</div>
</form>
</body>
</html>

```

Code file :

```

CheckBoxExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_CheckBoxExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void CheckBox_Status_CheckedChanged(object sender, EventArgs e)
    {
        if (CheckBox_Status.Checked)
        {
            CheckBox_IsProgrammer.Checked = true;
            CheckBox_NotProgrammer.Checked = false;
        }
        else
        {
            CheckBox_IsProgrammer.Checked = false;
            CheckBox_NotProgrammer.Checked = true;
        }
    }
}

```

CheckBoxList

Berikut contoh penggunaan CheckBoxList.

```

CheckBoxListExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CheckBoxListExample.aspx.cs"
Inherits="StandarServerControl_CheckBoxListExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>

```

```

<form id="form1" runat="server">
<div>
    Sarat peserta yang dapat ikut :
    <asp:CheckBoxList ID="CheckBoxList_Kelas" runat="server">
        <asp:ListItem Value="1">Kelas 1</asp:ListItem>
        <asp:ListItem Value="2">Kelas 2</asp:ListItem>
        <asp:ListItem Value="3">Kelas 3</asp:ListItem>
    </asp:CheckBoxList>
    <asp:Button ID="Button_Submit" runat="server" Text="Submit"
        onclick="Button_Submit_Click" />
    <br />
    <asp:Label ID="Label_Result" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>

```

Code file :

```

CheckBoxList.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_CheckBoxListExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Label_Result.Visible = false;
        }
    }

    protected void Button_Submit_Click(object sender, EventArgs e)
    {
        string hasil = String.Empty;

        foreach (ListItem item in CheckBoxList_Kelas.Items)
        {
            if (item.Selected)
            {
                hasil += item.Text + "<br/>";
            }
        }

        if (!String.IsNullOrEmpty(hasil))
        {
            Label_Result.Text = hasil;
            Label_Result.Visible = true;
        }
        else
        {
            Label_Result.Visible = false;
        }
    }
}

```

RadioButton

Contoh penggunaan control RadioButton.

RadioButtonExample.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="RadioButtonExample.aspx.cs"
Inherits="StandarServerControl_RadioButtonExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Apakah Anda programmer ?
            <asp:RadioButton ID="RadioButton1"
                Text="Ya"
                GroupName="Programmer"
                AutoPostBack="true"
                oncheckedchanged="RadioButton_CheckedChanged"
                runat="server" />
            <asp:RadioButton ID="RadioButton2"
                Text="Tidak"
                GroupName="Programmer"
                AutoPostBack="true"
                oncheckedchanged="RadioButton_CheckedChanged"
                runat="server" />

            <br />
            <asp:Label ID="Label_Hasil" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>

```

Code file :

<pre> RadioButtonExample.aspx.cs using System; using System.Collections; using System.Configuration; using System.Data; using System.Linq; using System.Web; using System.Web.Security; using System.Web.UI; using System.Web.UI.HtmlControls; using System.Web.UI.WebControls; using System.Web.UI.WebControls.WebParts; using System.Xml.Linq; public partial class StandarServerControl_RadioButtonExample : System.Web.UI.Page { protected void Page_Load(object sender, EventArgs e) { Label_Hasil.Visible = false; } protected void RadioButton_CheckedChanged(object sender, EventArgs e) { Label_Hasil.Visible = true; if (RadioButton1.Checked) { Label_Hasil.Text = "Saya adalah programmer"; } else { Label_Hasil.Text = "Saya bukan programmer"; } } } </pre>

RadioButtonList

Contoh penggunaan RadioButtonList.

```
RadioButtonListExample.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="RadioButtonListExample.aspx.cs"
Inherits="StandarServerControl_RadioButtonListExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:RadioButtonList ID="RadioButtonList_Language"
                RepeatDirection="Horizontal"
                RepeatLayout="Flow"
                runat="server">
                <asp:ListItem>Indonesia</asp:ListItem>
                <asp:ListItem>Inggris</asp:ListItem>
                <asp:ListItem>Jepang</asp:ListItem>
            </asp:RadioButtonList>
            <br />
            <asp:LinkButton ID="LinkButton_Submit" runat="server"
                onclick="LinkButton_Submit_Click">
                Submit
            </asp:LinkButton>
            <br />
            <asp:Label ID="Label_Hasil" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Code file :

```
RadioButtonListExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_RadioButtonListExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Label_Hasil.Text = String.Empty;
        }
    }

    protected void LinkButton_Submit_Click(object sender, EventArgs e)
    {
        Label_Hasil.Text = "Bahasa yang dipilih : ";
        Label_Hasil.Text += RadioButtonList_Language.SelectedItem.ToString();
    }
}
```

Image

Contoh penggunaan control Image.

ImageExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ImageExample.aspx.cs"
Inherits="StandarServerControl_ImageExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:DropDownList ID="DropDownList_Image"
                AutoPostBack="true" runat="server"
                onselectedIndexChanged="DropDownList_Image_SelectedIndexChanged">
                <asp:ListItem Value="">Pilih gambar</asp:ListItem>
                <asp:ListItem Value="batman.jpg">Batman</asp:ListItem>
                <asp:ListItem Value="iamlegend.jpg">I am Legend</asp:ListItem>
                <asp:ListItem Value="hancock.jpg">Hancock</asp:ListItem>
            </asp:DropDownList>
            <br />
            <asp:Image ID="Image_View" Width="400" runat="server" />
        </div>
    </form>
</body>
</html>
```

Code file :

ImageExample.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_ImageExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Image_View.Visible = false;
        }
    }

    protected void DropDownList_Image_SelectedIndexChanged(object sender, EventArgs e)
    {
        if (!String.IsNullOrEmpty(DropDownList_Image.SelectedValue))
        {
            Image_View.Visible = true;
            Image_View.ImageUrl = "~/Images/" + DropDownList_Image.SelectedValue;
        }
    }
}
```

Calendar

Control Calendar dapat digunakan sebagai control untuk menampilkan atau memasukkan nilai berupa tanggal. Pada calendar tanggal yang dipilih dapat satu tanggal atau beberapa tanggal yang biasanya berupa tanggal dalam satu minggu.

Berikut adalah contoh penggunaan control calendar yang untuk operasi pemilihan satu tanggal.

CalendarExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CalendarExample.aspx.cs"
Inherits="StandarServerControl_CalendarExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Calendar ID="Calendar_DateBirth" runat="server"
                onselectionchanged="Calendar_DateBirth_SelectionChanged"></asp:Calendar>
            <br />
            <asp:Label ID="Label_DateBirth" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Code file :

CalendarExample.aspx.cs

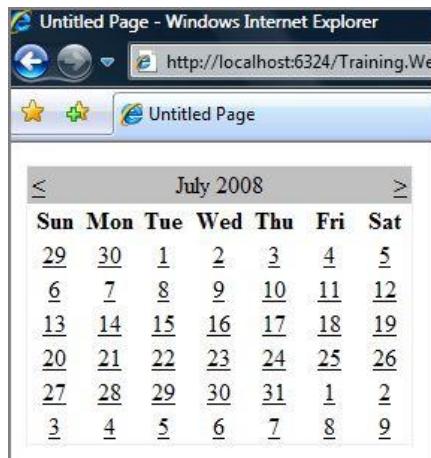
```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_CalendarExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Calendar_DateBirth_SelectionChanged(object sender, EventArgs e)
    {
        Label_DateBirth.Text = Calendar_DateBirth.SelectedDate.ToString();
    }
}
```

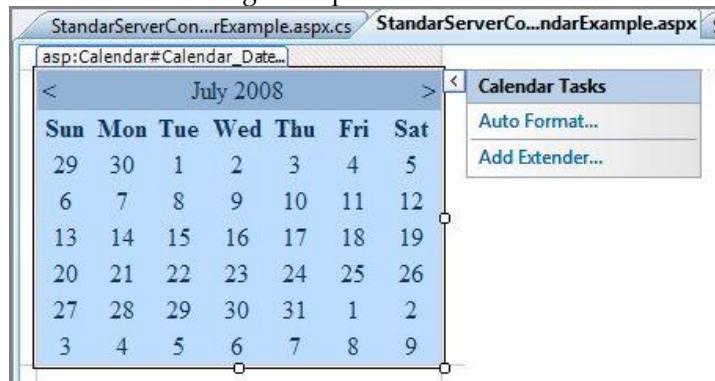
Pada contoh di atas dapat dilihat control Calendar yang ditampilkan seperti berikut :



July 2008						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
<u>29</u>	<u>30</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>
<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>	<u>19</u>
<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>	<u>25</u>	<u>26</u>
<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>	<u>31</u>	<u>1</u>	<u>2</u>
<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>	<u>7</u>	<u>8</u>	<u>9</u>

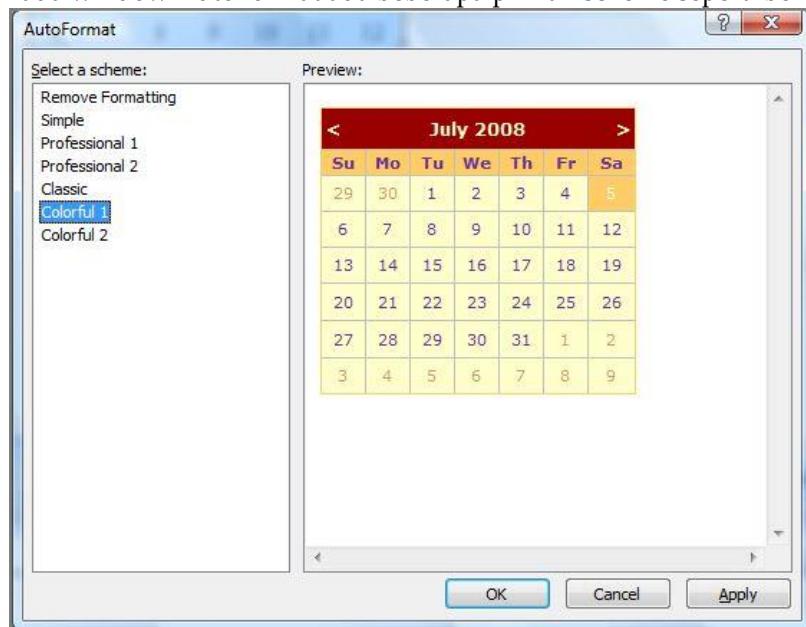
Style dari antarmuka control Calendar dapat diubah menjadi berbagai macam scheme dengan cara seperti berikut :

1. Aktifkan mode Design dan pilih control Calendar.



Pilih Auto Format.

2. Pada window AutoFormat ada beberapa pilihan scheme seperti berikut.



3. Pilih scheme dan klik tombol OK.

Perubahan scheme di atas akan berdampak perubahan kode pada halaman ASPX seperti berikut :

CalendarExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CalendarExample.aspx.cs"
Inherits="StandarServerControl_CalendarExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Calendar ID="Calendar_DateBirth" runat="server" BackColor="#FFFFCC"
                BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest"
                Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399" Height="200px"
                onselectionchanged="Calendar_DateBirth_SelectionChanged"
                ShowGridLines="True"
                Width="220px">
                <SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
                <SelectorStyle BackColor="#FFCC66" />
                <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
                <OtherMonthDayStyle ForeColor="#CC9966" />
                <NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />
                <DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
                <TitleStyle BackColor="#990000" Font-Bold="True" Font-Size="9pt"
                    ForeColor="#FFFFCC" />
            </asp:Calendar>
            <br />
            <asp:Label ID="Label_DateBirth" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Berikut ini contoh memilih tanggal pada Calendar secara programmatically.

CalendarChoose.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CalendarChoose.aspx.cs"
Inherits="StandarServerControl_CalendarChoose" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:DropDownList ID="DropDownList_Date" AutoPostBack="true" runat="server"
                onselectedindexchanged="DropDownList_Date_SelectedIndexChanged">
                <asp:ListItem Value="-2">-2</asp:ListItem>
                <asp:ListItem Value="-1">-1</asp:ListItem>
                <asp:ListItem Value="0">Hari ini</asp:ListItem>
                <asp:ListItem Value="+1">+1</asp:ListItem>
                <asp:ListItem Value="+2">+2</asp:ListItem>
            </asp:DropDownList>
            <br />
            <asp:Calendar ID="Calendar1" runat="server" BackColor="#FFFFCC"
                BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest"
                Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399" Height="200px"
                ShowGridLines="True" Width="220px">
                <SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
                <SelectorStyle BackColor="#FFCC66" />
                <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
                <OtherMonthDayStyle ForeColor="#CC9966" />
            </asp:Calendar>
        </div>
    </form>
</body>
</html>
```

```

<NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />
<DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
<TitleStyle BackColor="#990000" Font-Bold="True" Font-Size="9pt"
    ForeColor="#FFFFCC" />
</asp:Calendar>
</div>
</form>
</body>
</html>

```

Code file :

CalendaChoose.aspx.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_CalendarChoose : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void DropDownList_Date_SelectedIndexChanged(object sender, EventArgs e)
    {
        string date = DropDownList_Date.SelectedValue;

        switch (date)
        {
            case "-2":
                {
                    Calendar1.SelectedDate = DateTime.Now.Date.AddDays(-2);
                    break;
                }
            case "-1":
                {
                    Calendar1.SelectedDate = DateTime.Now.Date.AddDays(-1);
                    break;
                }
            case "+1":
                {
                    Calendar1.SelectedDate = DateTime.Now.Date.AddDays(1);
                    break;
                }
            case "+2":
                {
                    Calendar1.SelectedDate = DateTime.Now.Date.AddDays(2);
                    break;
                }
            default:
                {
                    Calendar1.SelectedDate = DateTime.Now;
                    break;
                }
        }
    }
}

```

Berikut ini contoh penggunaan pemilihan tanggal dalam satu minggu atau dalam satu bulan.

CalendarWeekDay.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CalendarWeekDay.aspx.cs"
Inherits="StandarServerControl_CalendarWeekDay" %>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Calendar ID="Calendar1" SelectionMode="DayWeekMonth" runat="server"
                BackColor="#FFFFCC"
                BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest"
                Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399" Height="200px"
                ShowGridLines="True" Width="220px"
                onselectionchanged="Calendar1_SelectionChanged">
                <SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
                <SelectorStyle BackColor="#FFCC66" />
                <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
                <OtherMonthDayStyle ForeColor="#CC9966" />
                <NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />
                <DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
                <TitleStyle BackColor="#990000" Font-Bold="True" Font-Size="9pt"
                    ForeColor="#FFFFCC" />
            </asp:Calendar>
            <br />
            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>

```

Code file :

CalendarWeekDay.aspx.cs
<pre> using System; using System.Collections; using System.Configuration; using System.Data; using System.Linq; using System.Web; using System.Web.Security; using System.Web.UI; using System.Web.UI.HtmlControls; using System.Web.UI.WebControls; using System.Web.UI.WebControls.WebParts; using System.Xml.Linq; public partial class StandarServerControl_CalendarWeekDay : System.Web.UI.Page { protected void Page_Load(object sender, EventArgs e) { } protected void Calendar1_SelectionChanged(object sender, EventArgs e) { Label1.Text = "Tanggal yang dipilih :
"; for (int i = 0; i < Calendar1.SelectedDates.Count; i++) { Label1.Text += Calendar1.SelectedDates[i].ToShortDateString() + "
"; } } } </pre>

Pada control Calendar dapat ditambahkan text seperti contoh berikut.



Berikut adalah contoh halaman untuk membuat contoh seperti di atas.

CalendarAddItem.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CalendarAddItem.aspx.cs"
Inherits="StandarServerControl_CalendarAddItem" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Calendar ID="Calendar1" runat="server" BackColor="#FFFFCC"
                BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest"
                Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399" Height="200px"
                ondayrender="Calendar1_DayRender" ShowGridLines="True" Width="220px">
                <SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
                <SelectorStyle BackColor="#FFCC66" />
                <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
                <OtherMonthDayStyle ForeColor="#CC9966" />
                <NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />
                <DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
                <TitleStyle BackColor="#990000" Font-Bold="True" Font-Size="9pt"
                    ForeColor="#FFFFCC" />
            </asp:Calendar>
        </div>
    </form>
</body>
</html>
```

Code file :

CalendarAddItem.aspx.cs

```
using System;
using System.Collections;
```

```

using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_CalendarAddItem : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Calendar1_DayRender(object sender, DayRenderEventArgs e)
    {
        e.Cell.VerticalAlign = VerticalAlign.Top;

        if (e.Day.DayNumberText.Equals("17"))
        {
            e.Cell.Controls.Add(new LiteralControl("<p>Upacara bendera</p>"));
            e.Cell.BackColor = System.Drawing.Color.Black;
            e.Cell.BorderWidth = 1;
            e.CellBorderStyle = BorderStyle.Solid;
            e.Cell.BackColor = System.Drawing.Color.LightGray;
        }
    }
}

```

Panel

Berikut contoh penggunaan Panel. Pada contoh ini dibuat dua buah panel, yaitu :

1. Panel untuk menampilkan form inputan.
2. Panel untuk menampilkan nilai-nilai yang dimasukkan pada form inputan.

```

PanelExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="PanelExample.aspx.cs"
Inherits="StandarServerControl_PanelExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Panel ID="Panel_View" runat="server">
                <asp:Label ID="Label_Message" runat="server"
                    Text="Klik tombol Add untuk mengisi field-field ini">
                </asp:Label>
                <br />
                <table>
                    <tr>
                        <td>
                            <asp:Label ID="Label_View_Nama" runat="server" Text="Nama">
                            </asp:Label>
                        </td>
                        <td>
                            <asp:Label ID="Label_View_NamaValue" runat="server"
                                Text="Label">
                            </asp:Label>
                        </td>
                    </tr>
                </table>
            </asp:Panel>
        </div>
    </form>
</body>
</html>

```

```

        <tr>
            <td valign="top">
                <asp:Label ID="Label_View_Alamat" runat="server"
Text="Alamat">
                    </asp:Label>
            </td>
            <td valign="top">:</td>
            <td>
                <asp:Label ID="Label_View_AlamatValue" runat="server"
Text="Label">
                    </asp:Label>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label_View_TanggalLahir" runat="server"
Text="Tanggal Lahir">
                    </asp:Label>
            </td>
            <td>:</td>
            <td>
                <asp:Label ID="Label_View_TanggalLahirValue" runat="server"
Text="Label">
                    </asp:Label>
            </td>
        </tr>
        <tr>
            <td colspan="3">
                <asp:Button ID="Button_Add" runat="server"
Text="Add"
                onclick="Button_Add_Click" />
            </td>
        </tr>
    </table>
</asp:Panel>

<asp:Panel ID="Panel_Form" runat="server">
    <table>
        <tr>
            <td>
                <asp:Label ID="Label_Form_Nama" runat="server" Text="Nama">
                    </asp:Label>
            </td>
            <td>:</td>
            <td>
                <asp:TextBox ID="TextBox_Form_Nama" runat="server">
                    </asp:TextBox>
            </td>
        </tr>
        <tr>
            <td valign="top">
                <asp:Label ID="Label_Form_Alamat" runat="server"
Text="Alamat">
                    </asp:Label>
            </td>
            <td valign="top">:</td>
            <td>
                <asp:TextBox ID="TextBox_Form_Alamat" TextMode="MultiLine"
runat="server">
                    </asp:TextBox>
            </td>
        </tr>
        <tr>
            <td valign="top">
                <asp:Label ID="Label_Form_TanggalLahir" runat="server"
Text="Tanggal Lahir">
                    </asp:Label>
            </td>
            <td valign="top">:</td>
            <td>
                <asp:Calendar ID="Calendar_TanggalLahir" runat="server"
BackColor="#FFFCC"
BorderColor="#FFCC66" BorderWidth="1px"
DayNameFormat="Shortest"
Font-Names="Verdana" Font-Size="8pt" ForeColor="#663399" />
            </td>
        </tr>
    </table>

```

```

        Height="200px"
        ShowGridLines="True" Width="220px">
        <SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
        <SelectorStyle BackColor="#FFCC66" />
        <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
        <OtherMonthDayStyle ForeColor="#CC9966" />
        <NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />
        <DayHeaderStyle BackColor="#FFCC66" Font-Bold="True"
                         Height="1px" />
        <TitleStyle BackColor="#990000" Font-Bold="True" Font-
                         Size="9pt"
                         ForeColor="#FFFFCC" />
    </asp:Calendar>
</td>
</tr>
<tr>
    <td colspan="3">
        <asp:Button ID="Button_Save" runat="server"
                    Text="Save"
                    onclick="Button_Save_Click" />
        <asp:Button ID="Button_Cancel" runat="server"
                    Text="Cancel"
                    onclick="Button_Cancel_Click" />
    </td>
</tr>
</table>
</asp:Panel>
</div>
</form>
</body>
</html>
```

Code file :

```

PanelExample.aspx
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_PanelExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Panel_View.Visible = true;
            Panel_Form.Visible = false;

            Label_View_NamaValue.Text = "-";
            Label_View_AlamatValue.Text = "-";
            Label_View_TanggalLahirValue.Text = "-";
        }
    }

    protected void Button_Add_Click(object sender, EventArgs e)
    {
        Panel_View.Visible = false;
        Panel_Form.Visible = true;

        TextBox_Form_Nama.Text = String.Empty;
        TextBox_Form_Alamat.Text = String.Empty;
        Calendar_TanggalLahir.SelectedDate = DateTime.Now;
    }

    protected void Button_Save_Click(object sender, EventArgs e)
```

```

    {
        Panel_View.Visible = true;
        Panel_Form.Visible = false;

        Label_View_NamaValue.Text = TextBox_Form_Nama.Text;
        Label_View_AlamatValue.Text = TextBox_Form_Alamat.Text.Replace("\n",
"\  
");
        Label_View_TanggalLahirValue.Text = String.Format("{0:dddd, dd MMMM yyyy}",
Calendar_TanggalLahir.SelectedDate);
    }

    protected void Button_Cancel_Click(object sender, EventArgs e)
    {
        Panel_View.Visible = true;
        Panel_Form.Visible = false;

        Label_View_NamaValue.Text = "-";
        Label_View_AlamatValue.Text = "-";
        Label_View_TanggalLahirValue.Text = "-";
    }
}

```

Multiview dan View

Multiview dan View adalah satu kesatuan control yang harus digunakan. Walau pada ToolBox kedua control ini terpisah. Multiview fungsinya seperti control Panel dengan penggunaan yang lebih mudah.

Pada contoh control Panel di atas, bila terdapat dua buat control Panel, maka untuk menampilkan hanya salah satu saja maka control Panel yang lain harus disembunyikan terlebih dahulu. Bayangkan bila jumlah control Panel yang ada pada halaman adalah 10 buah maka ada banyak kode yang ditulis hanya untuk menampilkan dan menyembunyikan control Panel.

Bila menggunakan control Multiview dan View, cukup ditunjuk saja View yang akan ditampilkan maka otomatis control View yang lain akan disembunyikan.

```

MultiviewExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="MultiviewExample.aspx.cs"
Inherits="StandarServerControl_MultiviewExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:MultiView ID="MultiView_Utama" runat="server">
                <asp:View ID="View_View" runat="server">
                    <asp:Label ID="Label_Message" runat="server"
                        Text="Klik tombol Add untuk mengisi field-field ini">
                    </asp:Label>
                    <br />
                    <table>
                        <tr>
                            <td>
                                <asp:Label ID="Label_View_Nama" runat="server"
Text="Nama">
                                    </asp:Label>
                                </td>
                            <td></td>
                            <td>
                                <asp:Label ID="Label_View_NamaValue" runat="server"
Text="Label">
                                    </asp:Label>
                            </td>
                        </tr>
                    </table>
                </asp:View>
            </asp:MultiView>
        </div>
    </form>
</body>
</html>

```

```

        </td>
    </tr>
    <tr>
        <td valign="top">
            <asp:Label ID="Label_View_Alamat" runat="server"
                Text="Alamat">
            </asp:Label>
        </td>
        <td valign="top"></td>
        <td>
            <asp:Label ID="Label_View_AlamatValue" runat="server"
                Text="Label">
            </asp:Label>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="Label_View_TanggalLahir" runat="server"
                Text="Tanggal Lahir">
            </asp:Label>
        </td>
        <td></td>
        <td>
            <asp:Label ID="Label_View_TanggalLahirValue"
runat="server"
                Text="Label">
            </asp:Label>
        </td>
    </tr>
    <tr>
        <td colspan="3">
            <asp:Button ID="Button_Add" runat="server"
                Text="Add"
                onclick="Button_Add_Click" />
        </td>
    </tr>
</table>
</asp:View>
<asp:View ID="View_Form" runat="server">
    <table>
        <tr>
            <td>
                <asp:Label ID="Label_Form_Nama" runat="server"
Text="Nama">
                    </asp:Label>
                </td>
                <td></td>
                <td>
                    <asp:TextBox ID="TextBox_Form_Nama" runat="server">
                    </asp:TextBox>
                </td>
            </tr>
            <tr>
                <td valign="top">
                    <asp:Label ID="Label_Form_Alamat" runat="server"
                        Text="Alamat">
                    </asp:Label>
                </td>
                <td valign="top"></td>
                <td>
                    <asp:TextBox ID="TextBox_Form_Alamat"
TextMode="MultiLine"
                        runat="server">
                    </asp:TextBox>
                </td>
            </tr>
            <tr>
                <td valign="top">
                    <asp:Label ID="Label_Form_TanggalLahir" runat="server"
                        Text="Tanggal Lahir">
                    </asp:Label>
                </td>
                <td valign="top"></td>
                <td>
                    <asp:Calendar ID="Calendar_TanggalLahir" runat="server">

```

```

        BackColor="#FFFFCC"
        BorderColor="#FFCC66" BorderWidth="1px"
        DayNameFormat="Shortest"
        Font-Names="Verdana" Font-Size="8pt"
        ForeColor="#663399"

    />
    <SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
    <SelectorStyle BackColor="#FFCC66" />
    <TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
    <OtherMonthDayStyle ForeColor="#CC9966" />
    <NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />
    <DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
    <TitleStyle BackColor="#990000" Font-Bold="True" />
    Font-
        Size="9pt"
        ForeColor="#FFFFCC" />
    </asp:Calendar>
</td>
</tr>
<tr>
    <td colspan="3">
        <asp:Button ID="Button_Save" runat="server" Text="Save" onclick="Button_Save_Click" />
        <asp:Button ID="Button_Cancel" runat="server" Text="Cancel" onclick="Button_Cancel_Click" />
    </td>
</tr>
</table>
</asp:View>
</asp:MultiView>
</div>
</form>
</body>
</html>

```

Code file :

```

MultiviewExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_MultiviewExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            MultiView_Utama.SetActiveView(View_View);

            Label_View_NamaValue.Text = "-";
            Label_View_AlamatValue.Text = "-";
            Label_View_TanggalLahirValue.Text = "-";
        }
    }

    protected void Button_Add_Click(object sender, EventArgs e)
    {
        MultiView_Utama.SetActiveView(View_Form);
    }
}

```

```

        TextBox_Form_Nama.Text = String.Empty;
        TextBox_Form_Alamat.Text = String.Empty;
        Calendar_TanggalLahir.SelectedDate = DateTime.Now;
    }

    protected void Button_Save_Click(object sender, EventArgs e)
    {
        MultiView_Utama.SetActiveView(View_View);

        Label_View_NamaValue.Text = TextBox_Form_Nama.Text;
        Label_View_AlamatValue.Text = TextBox_Form_Alamat.Text.Replace("\n",
"\\n");
        Label_View_TanggalLahirValue.Text = String.Format("{0:dddd, dd MMMM yyyy}",
Calendar_TanggalLahir.SelectedDate);
    }

    protected void Button_Cancel_Click(object sender, EventArgs e)
    {
        MultiView_Utama.SetActiveView(View_View);

        Label_View_NamaValue.Text = "-";
        Label_View_AlamatValue.Text = "-";
        Label_View_TanggalLahirValue.Text = "-";
    }
}

```

Wizard

Berikut contoh penggunaan control Wizard.

WizardExample.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="WizardExample.aspx.cs"
Inherits="StandarServerControl_WizardExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
    <asp:Wizard ID="Wizard_DataPenduduk" runat="server" ActiveStepIndex="0"
        BackColor="#EFF3FB" BorderColor="#B5C7DE" BorderWidth="1px"
        Font-Names="Verdana" Font-Size="0.8em"
        onfinishbuttonclick="Wizard_DataPenduduk_FinishButtonClick">
        <StepStyle Font-Size="0.8em" ForeColor="#333333" />
        <WizardSteps>
            <asp:WizardStep ID="WizardStep1" runat="server"
                Title="Langkah 1 : Data Pribadi">
                <table>
                    <tr>
                        <td>Nama</td>
                        <td>:</td>
                        <td>
                            <asp:TextBox ID="TextBox_Nama"
runat="server"></asp:TextBox>
                        </td>
                    </tr>
                    <tr>
                        <td valign="top">Alamat</td>
                        <td>:</td>
                        <td valign="top">
                            <asp:TextBox ID="TextBox_Alamat" TextMode="MultiLine"
runat="server"></asp:TextBox>
                        </td>
                    </tr>
                </table>
            </asp:WizardStep>
        </WizardSteps>
    </asp:Wizard>
</div>
</form>

```

```

        </table>
    </asp:WizardStep>
    <asp:WizardStep ID="WizardStep2" runat="server"
                    Title="Langkah 2 : Data Pendidikan">
        <table>
            <tr>
                <td>Data Pendidikan</td>
            </tr>
            <tr>
                <td>
                    <asp:TextBox ID="TextBox_Pendidikan" TextMode="MultiLine"
                                Rows="5" runat="server">
                    </asp:TextBox>
                </td>
            </tr>
        </table>
    </asp:WizardStep>
    <asp:WizardStep ID="WizardStep3" runat="server"
                    Title="Langkah 3 : Data Pekerjaan">
        <table cellpadding="3">
            <tr>
                <td>Pengalaman Kerja</td>
            </tr>
            <tr>
                <td>
                    <asp:TextBox ID="TextBox_Pekerjaan" TextMode="MultiLine"
                                Rows="5" runat="server">
                    </asp:TextBox>
                </td>
            </tr>
        </table>
    </asp:WizardStep>
</WizardSteps>
<SideBarButtonStyle BackColor="#507CD1" Font-Names="Verdana"
                    ForeColor="White" />
<NavigationBarStyle BackColor="White" BorderColor="#507CD1"
                    BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-
Size="0.8em"
                    ForeColor="#284E98" />
<SideBarStyle BackColor="#507CD1" Font-Size="0.9em" VerticalAlign="Top" />
<HeaderStyle BackColor="#284E98" BorderColor="#EFF3FB" BorderStyle="Solid"
                BorderWidth="2px" Font-Bold="True" Font-Size="0.9em" ForeColor="White"
                HorizontalAlign="Center" />
</asp:Wizard>
<asp:Panel ID="Panel_Hasil" Font-Names="Verdana" Font-Size="11px" runat="server">
    <table>
        <tr>
            <td>
                <asp:Label ID="Label_Nama" runat="server"
Text="Nama"></asp:Label>
                </td>
                <td>:</td>
                <td>
                    <asp:Label ID="Label_NamaValue" runat="server" Text="Label">
                    </asp:Label>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Label ID="Label_Alamat" runat="server" Text="Alamat">
                    </asp:Label>
                </td>
                <td>:</td>
                <td valign="top">
                    <asp:Label ID="Label_AlamatValue" runat="server" Text="Label">
                    </asp:Label>
                </td>
            </tr>
            <tr>
                <td>
                    <asp:Label ID="Label_Pendidikan" runat="server"
Text="Pendidikan">
                    </asp:Label>
                </td>
                <td>:</td>
            </tr>
        </table>
    </Panel>

```

```

        <td valign="top">
            <asp:Label ID="Label_PendidikanValue" runat="server"
Text="Label">
                </asp:Label>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label_Pekerjaan" runat="server" Text="Pekerjaan">
                    </asp:Label>
            </td>
            <td></td>
            <td valign="top">
                <asp:Label ID="Label_PekerjaanValue" runat="server" Text="Label">
                    </asp:Label>
            </td>
        </tr>
    </table>
</asp:Panel>
</div>
</form>
</body>
</html>c

```

Code file :

```

WizardExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_WizardExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Panel_Hasil.Visible = false;
        }
    }

    protected void Wizard_DataPenduduk_FinishButtonClick(object sender,
    WizardNavigationEventArgs e)
    {
        Label_NamaValue.Text = TextBox_Nama.Text;
        Label_AlamatValue.Text = TextBox_Alamat.Text.Replace("\n", "<br/>");
        Label_PendidikanValue.Text = TextBox_Pendidikan.Text.Replace("\n", "<br/>");
        Label_PekerjaanValue.Text = TextBox_Pekerjaan.Text.Replace("\n", "<br/>");
        Panel_Hasil.Visible = true;
    }
}

```

FileUpload

Berikut adalah contoh proses upload file dengan menggunakan control FileUpload.

```

FileUploadExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="FileUpload.aspx.cs"
Inherits="StandarServerControl_FileUpload" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:FileUpload ID="FileUpload_Document" runat="server" />
            <asp:Button ID="Button_Upload" 
                Text="Upload"
                onclick="Button_Upload_Click"
                runat="server"/>
            <asp:Label ID="Label_Status" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>

```

Code file :

```

FileUploadExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_FileUpload : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Label_Status.Visible = false;
        }
    }

    protected void Button_Upload_Click(object sender, EventArgs e)
    {
        if (FileUpload_Document.HasFile)
        {
            string uploadDir =
ConfigurationManager.AppSettings["UploadFolder"].ToString();
            string fileName = FileUpload_Document.FileName;
            string savePath = uploadDir + fileName;

            if (!System.IO.Directory.Exists(uploadDir))
            {
                System.IO.Directory.CreateDirectory(uploadDir);
            }

            FileUpload_Document.SaveAs(savePath);
            Label_Status.Visible = true;
            Label_Status.Text = "File " + fileName + " telah disimpan";
        }
    }
}

```

PlaceHolder

Control ini dapat digunakan untuk menambahkan control-control lain secara pemrograman. Berikut adalah contoh penggunaan control ini.

PlaceHolderExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="PlaceHolderExample.aspx.cs" %>
```

```

Inherits="StandarServerControl_PlaceHolderExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:PlaceHolder ID="PlaceHolder1" runat="server"></asp:PlaceHolder>
        </div>
    </form>
</body>
</html>

```

Code file:

```

PlaceHolderExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_PlaceHolderExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Create_UI();
    }

    protected void Create_UI()
    {
        Label lb = new Label();
        lb.Text = "Nama : ";
        PlaceHolder1.Controls.Add(lb);

        TextBox tb = new TextBox();
        tb.ID = "TextBox_Nama";
        tb.Width = Unit.Pixel(200);
        tb.ToolTip = "Isi nama di sini";
        tb.Text = "Reza";
        PlaceHolder1.Controls.Add(tb);
    }
}

```

Contoh lain penggunaan control PlaceHolder adalah sebagai berikut.

```

PlaceHolderForm.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="PlaceHolderForm.aspx.cs"
Inherits="StandarServerControl_PlaceHolderForm" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="TextBox_JumlahControl" Width="30"
runat="server"></asp:TextBox>
            <asp:Button ID="Button_CreateControl" runat="server" Text="Create Control"

```

```

        onclick="Button_CreateControl_Click" />
<br />
<asp:Label ID="Label_Status" ForeColor="Red" runat="server"
    Text="Ada kesalahan, cek nilai yang anda masukkan.">
</asp:Label>
<br />
<asp:PlaceHolder ID="PlaceHolder_Form" runat="server"></asp:PlaceHolder>
<asp:PlaceHolder ID="PlaceHolder_View" runat="server"></asp:PlaceHolder>
<asp:HiddenField ID="HiddenField_Status" runat="server" />
<br />
<asp:Button ID="Button_Submit" runat="server" Text="Submit"
    onclick="Button_Submit_Click" />
</div>
</form>
</body>
</html>

```

Code file :

```

PlaceHolderForm.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class StandarServerControl_PlaceHolderForm : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Button_Submit.Visible = false;
            Label_Status.Visible = false;
            HiddenField_Status.Value = "form";
        }

        if (!String.IsNullOrEmpty(TextBox_JumlahControl.Text) &&
            HiddenField_Status.Value.Equals("form"))
        {
            PlaceHolder_Form.Dispose();
            CreateForm();
        }
    }

    protected void Button_CreateControl_Click(object sender, EventArgs e)
    {
        HiddenField_Status.Value = "form";
    }

    protected void Button_Submit_Click(object sender, EventArgs e)
    {
        GetValueFromControl();
    }

    private void CreateForm()
    {
        try
        {
            int controlCount = Convert.ToInt32(TextBox_JumlahControl.Text);

            PlaceHolder_Form.Controls.Add(new LiteralControl("<table>"));
            for (int i = 1; i <= controlCount; i++)
            {
                TextBox tb = new TextBox();
                tb.ID = "TextBox_" + i.ToString();
            }
        }
    }
}

```

```

PlaceHolder_Form.Controls.Add(new LiteralControl("<tr>"));
PlaceHolder_Form.Controls.Add(new LiteralControl("<td>TextBox " +
    i.ToString() + "</td>"));
PlaceHolder_Form.Controls.Add(new LiteralControl("<td>:</td>"));
PlaceHolder_Form.Controls.Add(new LiteralControl("<td>"));
PlaceHolder_Form.Controls.Add(tb);
PlaceHolder_Form.Controls.Add(new LiteralControl("</td>"));
PlaceHolder_Form.Controls.Add(new LiteralControl("</tr>"));
}
PlaceHolder_Form.Controls.Add(new LiteralControl("</table>"));

Button_Submit.Visible = true;
}
catch (Exception ex)
{
    Label_Status.Visible = true;
}

PlaceHolder_View.Visible = false;
PlaceHolder_Form.Visible = true;
}

private void GetValueFromControl()
{
    int controlCount = Convert.ToInt32(TextBox_JumlahControl.Text);

    PlaceHolder_View.Controls.Add(new LiteralControl("<table>"));
    for (int i = 1; i <= controlCount; i++)
    {
        TextBox tb = (TextBox)PlaceHolder_Form.FindControl("TextBox_" +
            i.ToString());

        Label lb = new Label();
        lb.ID = "Label_" + i.ToString();
        lb.Text = tb.Text;

        PlaceHolder_View.Controls.Add(new LiteralControl("<tr>"));
        PlaceHolder_View.Controls.Add(new LiteralControl("<td>Label " +
            i.ToString()
            + "</td>"));
        PlaceHolder_View.Controls.Add(new LiteralControl("<td>:</td>"));
        PlaceHolder_View.Controls.Add(new LiteralControl("<td>"));
        PlaceHolder_View.Controls.Add(lb);
        PlaceHolder_View.Controls.Add(new LiteralControl("</td>"));
        PlaceHolder_View.Controls.Add(new LiteralControl("</tr>"));
    }
    PlaceHolder_Form.Controls.Add(new LiteralControl("</table>"));

    Button_Submit.Visible = false;
    PlaceHolder_View.Visible = true;
    PlaceHolder_Form.Visible = false;
}
}

```

5

ASP.NET dan Database

ADO.NET

ADO.NET adalah teknologi data access dari Microsoft. ADO.NET merupakan koleksi class-class yang merupakan bagian dari .NET Framework. ADO.NET dapat digunakan untuk mengakses banyak data source seperti :

1. Microsoft SQL Server.
2. Oracle.
3. Microsoft Access.
4. XML.
5. OLE DB.
6. ODBC.
7. Dan lain-lain.

Koneksi

Koneksi ke data source adalah langkah yang pertama mesti dilakukan sebelum dapat melakukan operasi database.

Berikut ini beberapa contoh koneksi database SQL Server dengan menggunakan beberapa cara.

ConnectionExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ConnectionExample.aspx.cs"
Inherits="DataAccess_ConnectionExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label_Status" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Code file :

ConnectionExample.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
```

```

using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using System.Data.SqlClient;
using System.Data.OleDb;
using System.Data.Odbc;

public partial class DataAccess_ConnectionExample : System.Web.UI.Page
{
    string status = String.Empty;

    protected void Page_Load(object sender, EventArgs e)
    {
        IntegratedSecurityConnection();
        AuthConnection();
        OleDbConnection();
        ODBCConnection();
    }

    protected void IntegratedSecurityConnection()
    {
        status += "Connect using .NET data provider for SQL Server and integrated
                  security :<br/>";
        string sqlConnectionString = "Data Source=(local);"
                                    + "Integrated security=SSPI;Initial Catalog=AdventureWorks;";

        SqlConnection conn = new SqlConnection(sqlConnectionString);
        conn.Open();

        status += "State : " + conn.State + "<br/>";
        status += "Datasource : " + conn.DataSource + "<br/>";
        status += "Server version : " + conn.ServerVersion + "<br/>";

        Label_Status.Text += status;
    }

    protected void AuthConnection()
    {
        status = "<br/>Connect using .NET data provider for SQL Server and SQL Server
                  authentication :<br/>";
        string sqlConnectionString = "Data Source=(local);"
                                    + "User Id=sa;Password=MRezaFaisal;Initial Catalog=AdventureWorks;";

        SqlConnection conn = new SqlConnection(sqlConnectionString);
        conn.Open();

        status += "State : " + conn.State + "<br/>";
        status += "Datasource : " + conn.DataSource + "<br/>";
        status += "Server version : " + conn.ServerVersion + "<br/>";

        Label_Status.Text += status;
    }

    protected void OleDbConnection()
    {
        status = "<br/>Connect using .NET data provider for OLE DB :<br/>";
        string oleDbConnectionString = "Provider=SQLOLEDB;Data Source=(local);"
                                    + "Initial Catalog=AdventureWorks;Integrated Security=SSPI;";

        OleDbConnection conn = new OleDbConnection(oleDbConnectionString);
        conn.Open();

        status += "State : " + conn.State + "<br/>";
        status += "Datasource : " + conn.DataSource + "<br/>";
        status += "Server version : " + conn.ServerVersion + "<br/>";

        Label_Status.Text += status;
    }

    protected void ODBCConnection()
    {
        status = "<br/>Connect using .NET data provider for ODBC :<br/>";
        string odbcConnectionString = "Driver={SQL Native Client};"
                                    + "Server=(local);Database=AdventureWorks;uid=sa;pwd=MRezaFaisal;";
```

```

        OdbcConnection conn = new OdbcConnection(odbcConnectionString);
        conn.Open();

        status += "State : " + conn.State + "<br/>";
        status += "Datasource : " + conn.DataSource + "<br/>";
        status += "Server version : " + conn.ServerVersion + "<br/>";

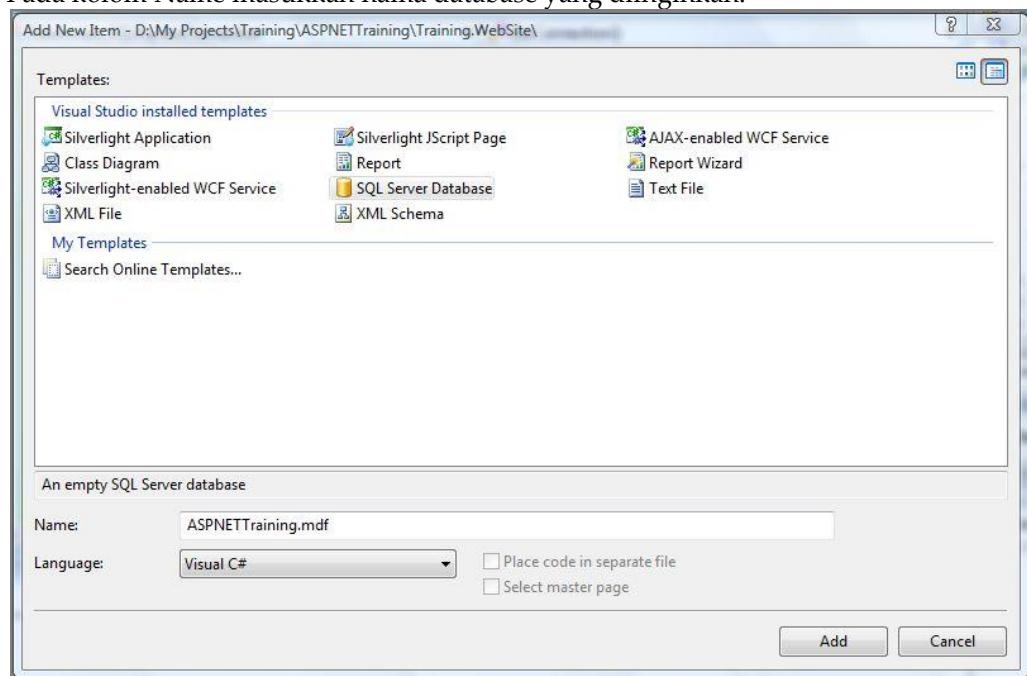
        Label_Status.Text += status;
    }
}

```

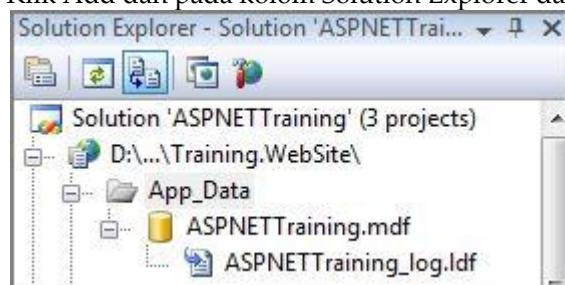
Pada Visual Studio memungkinkan untuk membuat file database SQL Server yang akan disimpan pada direktori App_Data. Untuk file ini database SQL Server diperlukan SQL Server Express.

Untuk menambahkan file database SQL Server ini dapat dilakukan dengan langkah-langkah berikut :

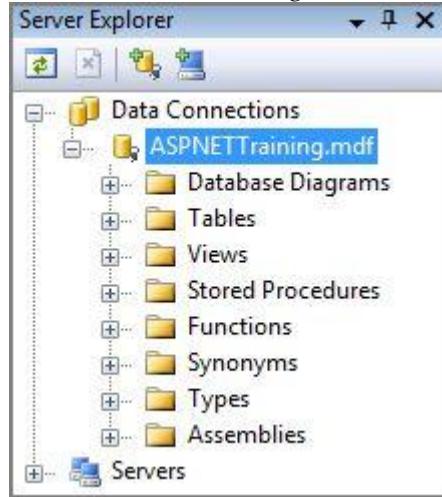
1. Bila pada Web Site belum terdapat folder App_Data maka klik kanan pada Web Site dan pilih Add ASP.NET Folder > App_Data.
2. Untuk menambahkan file database, klik kanan pada folder App_Data kemudian pilih Add New Item.
3. Pada window Add New Item pilih SQL Server Database pada kolom Templates.
4. Pada kolom Name masukkan nama database yang diinginkan.



5. Klik Add dan pada kolom Solution Explorer dapat dilihat hasil seperti berikut.



6. Pada bagian sebelah kanan Visual Studio dapat dilihat ditampilkan kolom Server Explorer. Pada kolom ini dapat dilihat table-table pada file database yang baru saja dibuat. Pada kolom Server Explorer dapat dilakukan aksi-aksi seperti membuat table, melihat data, mengisi data dan lain-lain.



Berikut ini adalah cara untuk melakukan koneksi pada file database.

SQLExpressExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="SQLExpressConnection.aspx.cs"
Inherits="DataAccess_SQLExpressConnection" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label_Status" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Code file :

SQLExpressExample.aspx

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using System.Data.SqlClient;

public partial class DataAccess_SQLExpressConnection : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
```

```

        Label_Status.Text = String.Empty;
    }
    GetConnection();
}

protected void GetConnection()
{
    string status = "Connect to SQL Express :<br/>";
    string sqlConnectionString = "Data Source=LEAF\\SQLEXPRESS,
AttachDbFilename=|DataDirectory|ASPNETTraining.mdf,
Integrated Security=True,
User Instance=True;";

    SqlConnection conn = new SqlConnection(sqlConnectionString);
    conn.Open();

    status += "State : " + conn.State + "<br/>";
    status += "Datasource : " + conn.DataSource + "<br/>";
    status += "Server version : " + conn.ServerVersion + "<br/>";

    Label_Status.Text += status;
}
}

```

Dari contoh-contoh di atas connection string ditulis langsung pada kode program. Pada contoh berikutnya connection string akan disimpan pada file web.config.

```

ConnectionStringExample.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ConnectionStringExample.aspx.cs"
Inherits="DataAccess_ConnectionStringExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label_Status" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>

```

Code file :

```

ConnectionStringExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using System.Data.SqlClient;

public partial class DataAccess_ConnectionStringExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
    }
}

```

```

        {
            Label_Status.Text = String.Empty;
        }
        GetConnection();
    }

    protected void GetConnection()
    {
        string status = "Connect to SQL Express :<br/>";
        string sqlConnectionString =
ConfigurationManager.ConnectionStrings["SQLExpressConnString"].ToString();

        SqlConnection conn = new SqlConnection(sqlConnectionString);
        conn.Open();

        status += "State : " + conn.State + "<br/>";
        status += "Datasource : " + conn.DataSource + "<br/>";
        status += "Server version : " + conn.ServerVersion + "<br/>";

        Label_Status.Text += status;
    }
}

```

Operasi Data

Ada beberapa tipe query yang dapat dijalankan, yaitu :

1. Query tanpa mengembalikan result set.
2. Query dengan pengembalian single value.
3. Query dengan pengembalian result set.

Query tanpa mengembalikan result set

Contoh query yang tidak mengembalikan nilai berupa single value atau result set diantaranya adalah :

1. INSERT.
2. UPDATE.
3. DELETE.

Berikut adalah contoh penggunaannya untuk proses insert data.

```

ADONETInsert.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ADONETInsert.aspx.cs"
Inherits="DataAccess_ADONETInsert" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table>
                <tr>
                    <td>
                        <asp:Label ID="Label_Nama" AssociatedControlID="TextBox_Nama"
                            runat="server" Text="Nama">
                        </asp:Label>
                    </td>
                    <td></td>
                    <td>
                        <asp:TextBox ID="TextBox_Nama" runat="server">
                        </asp:TextBox>
                    </td>
                </tr>
            </table>
        </div>
    </form>
</body>

```

```

        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="Label_Pekerjaan"
                AssociatedControlID="TextBox_Pekerjaan" runat="server"
                Text="Pekerjaan">
            </asp:Label>
        </td>
        <td></td>
        <td>
            <asp:TextBox ID="TextBox_Pekerjaan" runat="server">
            </asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Label ID="Label_Alamat" AssociatedControlID="TextBox_Alamat"
                runat="server" Text="Alamat">
            </asp:Label>
        </td>
        <td></td>
        <td>
            <asp:TextBox ID="TextBox_Alamat" runat="server">
            </asp:TextBox>
        </td>
    </tr>
    <tr>
        <td>
            <asp:Button ID="Button_Submit"
                Text="Submit"
                onclick="Button_Submit_Click"
                runat="server"/>
        </td>
    </tr>
</table>
<asp:Label ID="Label_Status" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>

```

Code file :

```

ADONETInsert.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataAccess_ADONETInsert : System.Web.UI.Page
{
    string connStr =
    ConfigurationManager.ConnectionStrings["SQLEXPRESSConnString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            Label_Status.Visible = false;
        }
    }

    protected void InsertData()
    {
        string strSQL = "INSERT INTO PesertaTraining (Nama, Pekerjaan, Alamat)

```

```

        Values (@Nama, @Pekerjaan, @Alamat");

    try
    {
        SqlConnection conn = new SqlConnection(connStr);
        SqlCommand cmd = new SqlCommand(strSQL, conn);

        conn.Open();
        cmd.Parameters.Add("@Nama", SqlDbType.NVarChar).Value =
        TextBox_Nama.Text;
        cmd.Parameters.Add("@Pekerjaan", SqlDbType.NVarChar).Value =
        TextBox_Pekerjaan.Text;
        cmd.Parameters.Add("@Alamat", SqlDbType.NVarChar).Value =
        TextBox_Alamat.Text;
        int rowsAffected = cmd.ExecuteNonQuery();

        Label_Status.Text = "Data saved. <br/>";
        Label_Status.Text += rowsAffected + " row(s) affected.";
        Label_Status.Visible = true;
        conn.Close();
    }
    catch (Exception ex)
    {
        Label_Status.Text = ex.Message;
    }
    finally
    {
        TextBox_Nama.Text = String.Empty;
        TextBox_Pekerjaan.Text = String.Empty;
        TextBox_Alamat.Text = String.Empty;
    }
}

protected void Button_Submit_Click(object sender, EventArgs e)
{
    InsertDate();
}
}

```

Contoh di atas dapat juga digunakan untuk proses update dan hapus data, dengan cara mengubah perintah SQL.

Query dengan pengembalian single value

Berikut ini contoh kode yang digunakan untuk menjalankan query dengan pengembalian single value.

```

ADONETSingleValue.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ADONETSingleValue.aspx.cs"
Inherits="DataAccess_ADONETSingleValue" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Label ID="Label_Value" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>

```

Code file :

```

ADONETSingleValue.aspx.cs
using System;

```

```

using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class DataAccess_ADONETSsingleValue : System.Web.UI.Page
{
    string connStr =
    ConfigurationManager.ConnectionStrings["SQLExpressConnString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        Label_Value.Text = "Jumlah data pada tabel PesertaTraining :
    "+GetSingleValue();
    }

    protected string GetSingleValue()
    {
        string strSQL = "SELECT COUNT(*) FROM PesertaTraining";

        try
        {
            SqlConnection conn = new SqlConnection(connStr);
            SqlCommand cmd = new SqlCommand(strSQL, conn);

            conn.Open();
            string count = Convert.ToString(cmd.ExecuteScalar());
            conn.Close();

            return count;
        }
        catch (Exception ex)
        {
            return ex.Message;
        }
    }
}

```

Query dengan pengembalian berupa result set

Berikut contoh penggunaan query untuk mengambil nilai yang berupa result set.

DataReaderExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DataReaderExample.aspx.cs" Inherits="DataAccess_DataReaderExample" %> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml"> <head runat="server"> <title>Untitled Page</title> </head> <body> <form id="form1" runat="server"> <div> <asp:PlaceHolder ID="PlaceHolder_Data" runat="server"></asp:PlaceHolder> </div> </form> </body> </html>

Code file :

DataReaderExample.aspx.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class DataAccess_DataReaderExample : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["SQLEXpressConnString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        GetDataOrganisasi();
    }

    protected void GetDataOrganisasi()
    {
        string strSQL = "SELECT * FROM Organisasi";

        SqlConnection conn = new SqlConnection(connStr);
        SqlCommand cmd = new SqlCommand(strSQL, conn);

        conn.Open();
        SqlDataReader dr = cmd.ExecuteReader();
        PlaceHolder_Data.Controls.Add(new LiteralControl("<table border=\"1\">"));
        while (dr.Read())
        {
            PlaceHolder_Data.Controls.Add(new LiteralControl("<tr>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("<td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl(
                dr["OrganisasiID"].ToString()));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("<td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl(
                dr["OrganisasiNama"].ToString()));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</tr>"));
        }
        PlaceHolder_Data.Controls.Add(new LiteralControl("</table>"));
        conn.Close();
    }
}

```

Berikut contoh lain dengan menggunakan DataReader untuk menampilkan data satu record yang telah dipilih pada suatu perintah SQL.

```

DataReaderGetOneData.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DataReaderGetOneData.aspx.cs"
Inherits="DataAccess_DataReaderGetOneData" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="TextBox_ID" Width="30" runat="server"></asp:TextBox>
            <asp:Button ID="Button_Search" runat="server" Text="Search"
                onclick="Button_Search_Click" />
        <br />
            <asp:Label ID="Label_Status" ForeColor="Red"
                Font-Italic="true" />
        </div>
    </form>
</body>

```

```

        runat="server"
        Text="Label">
    </asp:Label>
    <table>
        <tr>
            <td>
                <asp:Label ID="Label_ID" runat="server" Text="ID">
                    </asp:Label>
            </td>
            <td>:</td>
            <td>
                <asp:Label ID="Label_IDValue" runat="server" Text="Label">
                    </asp:Label>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label_Nama" runat="server" Text="Nama">
                    </asp:Label>
            </td>
            <td>:</td>
            <td>
                <asp:Label ID="Label_NamaValue" runat="server" Text="Label">
                    </asp:Label>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label_Pekerjaan" runat="server" Text="Pekerjaan">
                    </asp:Label>
            </td>
            <td>:</td>
            <td>
                <asp:Label ID="Label_PekerjaanValue" runat="server" Text="Label">
                    </asp:Label>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label_Alamat" runat="server" Text="Alamat">
                    </asp:Label>
            </td>
            <td>:</td>
            <td>
                <asp:Label ID="Label_AlamatValue" runat="server" Text="Label">
                    </asp:Label>
            </td>
        </tr>
    </table>
    </div>
    </form>
</body>
</html>

```

Code file :

```

DataReaderGetOneData.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class DataAccess_DataReaderGetOneData : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["SQLExpressConnString"].ToString();
}

```

```

protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        Label_Status.Text = String.Empty;
        Label_IDValue.Text = "-";
        Label_NamaValue.Text = "-";
        Label_PekerjaanValue.Text = "-";
        Label_AlamatValue.Text = "-";
    }
}

protected void Button_Search_Click(object sender, EventArgs e)
{
    if (!String.IsNullOrEmpty(textBox_ID.Text))
    {
        int id = Convert.ToInt32(textBox_ID.Text);
        GetPesertaTrainingById(id);
    }
}

protected void GetPesertaTrainingById(int id)
{
    string strSQL = "SELECT * FROM PesertaTraining WHERE ID = @ID";

    try
    {
        SqlConnection conn = new SqlConnection(connStr);
        SqlCommand cmd = new SqlCommand(strSQL, conn);
        cmd.Parameters.AddWithValue("@ID", SqlDbType.Int).Value = id;

        conn.Open();

        SqlDataReader dr = cmd.ExecuteReader();
        dr.Read();
        Label_IDValue.Text = dr["ID"].ToString();
        Label_NamaValue.Text = dr["Nama"].ToString();
        Label_PekerjaanValue.Text = dr["Pekerjaan"].ToString();
        Label_AlamatValue.Text = dr["Alamat"].ToString();
        conn.Close();

        Label_Status.Text = String.Empty;
    }
    catch (Exception ex)
    {
        Label_Status.Text = ex.Message;
        Label_IDValue.Text = "-";
        Label_NamaValue.Text = "-";
        Label_PekerjaanValue.Text = "-";
        Label_AlamatValue.Text = "-";
    }
}
}

```

Selain menggunakan DataReader untuk mengambil data pada query yang melemparkan hasil berupa result set, dapat juga digunakan class lain seperti :

1. DataTable.
2. DataSet.

Berikut contoh penggunaan DataTable.

DataTableExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DataTableExample.aspx.cs" Inherits="DataAccess_DataTableExample" %> <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml">

```

<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:PlaceHolder ID="PlaceHolder_Data" runat="server"></asp:PlaceHolder>
            <asp:Label ID="Label_Count" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>

```

Code file :

```

DataTableExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class DataAccess_DataTableExample : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["SQLExpressConnString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        GetData();
    }

    protected void GetData()
    {
        string strSQL = "SELECT * FROM Organisasi";

        SqlDataAdapter da = new SqlDataAdapter(strSQL, connStr);
        DataTable dt = new DataTable();
        da.Fill(dt);

        PlaceHolder_Data.Controls.Add(new LiteralControl("<table border=\"1\">"));
        foreach (DataRow row in dt.Rows)
        {
            PlaceHolder_Data.Controls.Add(new LiteralControl("<tr>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("<td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl(
                row["OrganisasiID"].ToString()));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("<td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl(
                row["OrganisasiNama"].ToString()));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</tr>"));
        }
        PlaceHolder_Data.Controls.Add(new LiteralControl("</table>"));

        Label_Count.Text = "Jumlah data : " + dt.Rows.Count.ToString();
    }
}

```

Dan berikut adalah contoh penggunaan DataSet untuk menampung data hasil perintah SQL.

```

DataSetExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DataSetExample.aspx.cs"
Inherits="DataAccess_DataSetExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"

```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:PlaceHolder ID="PlaceHolder_Data" runat="server"></asp:PlaceHolder>
        </div>
    </form>
</body>
</html>

```

Code file :

```

DataSetExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class DataAccess_DataSetExample : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["SQLEXPRESSConnString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        GetData();
    }

    protected void GetData()
    {
        string strSQL = "SELECT * FROM Organisasi";

        SqlDataAdapter da = new SqlDataAdapter(strSQL, connStr);
        DataSet ds = new DataSet();
        da.Fill(ds, "Organisasi");

        PlaceHolder_Data.Controls.Add(new LiteralControl("<table border=\"1\">"));
        foreach (DataRow row in ds.Tables["Organisasi"].Rows)
        {
            PlaceHolder_Data.Controls.Add(new LiteralControl("<tr>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("<td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl(
                row["OrganisasiID"].ToString()));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("<td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl(
                row["OrganisasiNama"].ToString()));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</td>"));
            PlaceHolder_Data.Controls.Add(new LiteralControl("</tr>"));
        }
        PlaceHolder_Data.Controls.Add(new LiteralControl("</table>"));
    }
}

```

LINQ

C# 3.0

Sebelum berkenalan dengan LINQ, ada beberapa feature baru yang terdapat pada C# yang mesti diketahui. Feature-feature ini adalah :

1. Kata kunci var, kata kunci var berfungsi untuk menggantikan penggunaan tipe data saat mendeklarasikan variabel. Pada C#, berikut ini adalah pendeklarasian variabel pada umumnya, yaitu :

```
int umur;  
string nama;
```

Dengan deklarasi di atas artinya variabel umur hanya dapat diisi dengan nilai yang bertipe integer dan variabel nama hanya dapat diisi dengan nilai yang bertipe integer. Bila nilai yang dimasukkan tidak sesuai dengan tipe data yang telah dideklarasikan maka akan diberikan pesan kesalahan saat proses build.

Penggunaan kata kunci var sebagai pengganti tipe data dapat dilihat pada contoh berikut :

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using STS.Common.Math;  
using System.Collections;  
  
namespace STS.Console  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            var umur = 0;  
            var nama = String.Empty;  
  
            umur = 13;  
  
            System.Console.ReadKey();  
        }  
    }  
}
```

Pada contoh di atas, dapat dilihat bagaimana kata kunci var digunakan sebagai pengganti tipe data. Penggunaan kata kunci var dalam mendeklarasikan suatu variabel harus diikuti dengan pemberian nilai inisial pada variabel tersebut. Pada contoh variabel nama diberikan nilai inisial berupa string, sehingga selanjutnya variabel nama hanya bisa diisi dengan nilai bertipe string.

Berikut adalah contoh lain dari penggunaan kata kunci var.

Program.cs

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using STS.Common.Math;  
using System.Collections;  
  
namespace STS.Console  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
        }
```

```

        var ht = new Hashtable();

        var p1 = new Pegawai();
        p1.Nama = "Reza";
        p1.Umur = 30;
        ht.Add("P1", p1);

        var p2 = new Pegawai();
        p2.Nama = "Faisal";
        p2.Umur = 25;
        ht.Add("P2", p2);

        System.Console.WriteLine("Jumlah objek : " + ht.Count);

        IDictionaryEnumerator enumerator = ht.Getenumerator();
        while (enumerator.MoveNext())
        {
            var pegawai = (Pegawai)enumerator.Value;

            System.Console.WriteLine(enumerator.Key.ToString() + " " +
                pegawai.Nama + "(" + pegawai.Umur + ")");
        }

        System.Console.ReadKey();
    }
}

```

Istilah untuk penggunaan kata kunci var ini adalah Implicit typed local variables.

2. Pada C# 3.0 terdapat feature yang memudahkan untuk pembuatan objek atau pendeklarasian collection. Berikut adalah contoh pembuatan objek yang sering dilakukan.

```

Pegawai p1 = new Pegawai();
p1.Nama = "Reza";
p1.Umur = 30;

Pegawai p2 = new Pegawai();
p2.Nama = "Faisal";
p2.Umur = 25;

```

Dengan menggunakan C# 3.0, maka kode di atas dapat ditulis kembali menjadi seperti ini.

```

var p1 = new Pegawai { Nama = "Reza", Umur = 31 };

var p2 = new Pegawai { Nama = "Faisal", Umur = 25 };

```

Contoh di atas adalah pembuatan objek dengan C# 3.0.

Berikut ini adalah contoh yang bisa digunakan untuk inisialisasi dan memberikan nilai pada collection.

```

Hashtable ht = new Hashtable();

Pegawai p1 = new Pegawai();
p1.Nama = "Reza";
p1.Umur = 30;

Pegawai p2 = new Pegawai();
p2.Nama = "Faisal";
p2.Umur = 25;

ht.Add("P1", p1);
ht.Add("P2", p2);

```

Dengan C# 3.0 maka contoh di atas dapat ditulis menjadi berikut ini :

```

Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;
using System.Collections;

```

```

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            var data = new List<Pegawai>
            {
                new Pegawai{Nama="Reza", Umur=31},
                new Pegawai{Nama="Faisal", Umur=25}
            };

            System.Console.WriteLine("Jumlah objek : " + data.Count);

            for (int i = 0; i < data.Count; i++)
            {
                var pegawai = (Pegawai)data[i];
                System.Console.WriteLine(pegawai.Nama+" ("+pegawai.Umur+" )");
            }

            System.Console.ReadKey();
        }
    }
}

```

Contoh yang lain. Sebelumnya bila pada class terdapat field dengan access-modifier private maka perlu dibuat public method seperti pada kode berikut :

```

CurveMath.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace STS.Common.Math
{
    public class CurveMath
    {
        private int panjang;
        private int lebar;
        private int jejari;

        public int Panjang
        {
            set {panjang = value;}
            get { return panjang; }
        }

        public int Lebar
        {
            set { lebar = value; }
            get { return lebar; }
        }

        public int Jejari
        {
            set { jejari = value; }
            get { return jejari; }
        }

        public virtual void HitungLuas()
        {
            System.Console.WriteLine(Panjang * Lebar);
        }
    }
}

```

Dengan menggunakan C# 3.0 maka kode di atas dapat ditulis ulang seperti kode berikut :

```

CurverMath.cs
using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;

namespace STS.Common.Math
{
    public class CurveMath
    {
        public int Panjang { set; get; }
        public int Lebar { set; get; }
        public int Jejari { set; get; }

        public virtual void HitungLuas()
        {
            System.Console.WriteLine(Panjang * Lebar);
        }
    }
}

```

3. Extension Method, memungkinkan menambahkan method untuk digunakan pada suatu objek atau variabel bahkan konstanta.

Berikut ini contoh class yang berisi extension method.

```

RezaExtension.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;

namespace STS.Console
{
    public static class RezaExtension
    {
        public static bool IsValidEmailAddress(this string s)
        {
            Regex regex = new Regex(@"^[\w-\.]+@[^\w-]+\.\w{2,4}$");
            return regex.IsMatch(s);
        }

        public static string AreYouReza(this string s)
        {
            if (!String.IsNullOrEmpty(s))
            {
                if (s.Equals("Reza"))
                {
                    return "You are Reza";
                }
                else
                {
                    return "You are not Reza";
                }
            }
            else
            {
                return "You are not Reza";
            }
        }
    }
}

```

Biasanya method-method pada class di atas dapat digunakan seperti berikut :

```

Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;
using System.Collections;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            CurveMath cm = new CurveMath();
            cm.HitungLuas();
        }
    }
}

```

```

    {
        string email = "reza@yahoo.com";

        if (RezaExtension.IsValidEmailAddress(email))
        {
            System.Console.WriteLine(email + " is valid");
        }
        else
        {
            System.Console.WriteLine(email + " is not valid");
        }

        System.Console.ReadKey();
    }
}

```

Dengan feature C# 3.0 ini maka cara penggunaanya cukup dengan cara berikut :

Program.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;
using System.Collections;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            string email = "reza@yahoo.com";

            if (email.IsValidEmailAddress())
            {
                System.Console.WriteLine(email + " is valid");
            }
            else
            {
                System.Console.WriteLine(email + " is not valid");
            }

            System.Console.ReadKey();
        }
    }
}

```

Contoh lain :

Program.cs

```

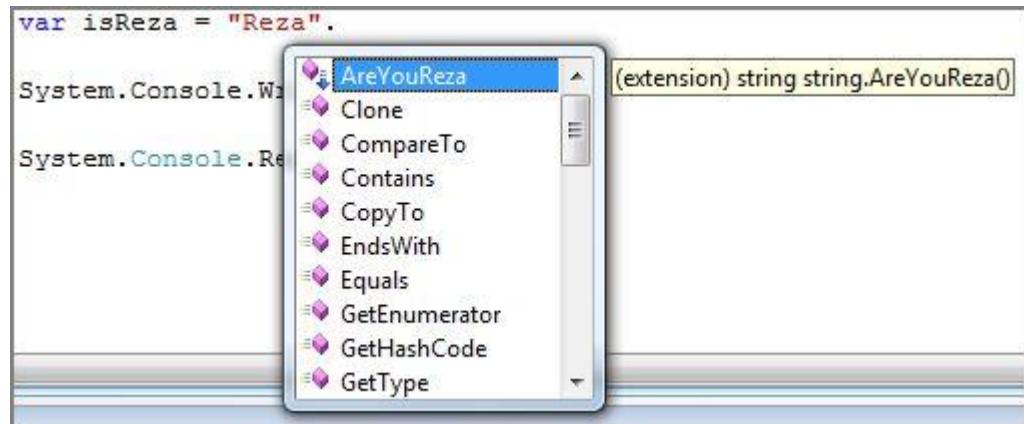
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;
using System.Collections;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            var isReza = "Reza".AreYouReza();

            System.Console.WriteLine(isReza);

            System.Console.ReadKey();
        }
    }
}

```



4. Berikut ini contoh kode yang menggunakan Lambda Expression.

```
p => p.Nama == "Reza"
```

Dengan sintaks :

```
[input_parameter] => [expression atau statement block]
```

Tanda “=>” adalah lambda operator.

Berikut adalah contoh lengkap penggunaan Lambda Expression.

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using STS.Common.Math;
using System.Collections;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            var data = new List<Pegawai>
            {
                new Pegawai{Nama="Reza", Umur=31},
                new Pegawai{Nama="Faisal", Umur=25},
                new Pegawai{Nama="Adi", Umur=31},
                new Pegawai{Nama="Negara", Umur=33}
            };

            IEnumerable<Pegawai> hasil = data.Where((Pegawai p) => p.Umur == 31);
            foreach (Pegawai pegawai in hasil)
            {
                System.Console.WriteLine(pegawai.Nama);
            }

            System.Console.ReadKey();
        }
    }
}
```

Atau dapat ditulis dengan lebih singkat seperti berikut :

```
Program.cs
using System;
using System.Collections.Generic;
using System.Linq;
```

```

using System.Text;
using STS.Common.Math;
using System.Collections;

namespace STS.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            var data = new List<Pegawai>
            {
                new Pegawai{Nama="Reza", Umur=31},
                new Pegawai{Nama="Faisal", Umur=25},
                new Pegawai{Nama="Adi", Umur=31},
                new Pegawai{Nama="Negara", Umur=33}
            };

            var hasil = data.Where(p => p.Umur == 31);
            foreach (Pegawai pegawai in hasil)
            {
                System.Console.WriteLine(pegawai.Nama);
            }

            System.Console.ReadKey();
        }
    }
}

```

LINQ

LINQ adalah singkatan dari Language INtegrated Query. Dengan feature ini dapat ditulis query langsung pada kode program. Dengan LINQ bisa dikatakan ‘apapun’ bisa di-query.

Berikut contoh-contoh penulisan query pada kode program.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace LINQ.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            string[] peserta = {
                "Faisal", "Frans", "Fandy",
                "Ganto", "Garfield",
                "Heni", "Hendra", "Herman",
                "Ivan", "Ida", "Indiana Jones",
                "Jenifer", "Jainal",
                "Adam", "Adi", "Antono",
                "Benjamin", "Bedy", "Bono",
                "Cintia", "Candy",
                "Dedy",
                "Evi", "Endang",
                "Karni", "Kesya", "Kennedy",
                "Lincon", "Lina", "Lena", "Lenatan",
                "Mia", "Mya", "M Reza Faisal",
                "Nia", "Nene",
                "Paula", "Paul", "Paulina",
                "Rerie", "Ria", "Rena",
                "Susi",
                "Tiara", "Tina Toon", "Thomas",
                "Ujang",
                "Van Buren", "Via", "Vanda",
                "Wiwiien",
                "Zainal"
            };

            var query = from n in peserta select n;
            foreach (string nama in query)

```

```
        {
            System.Console.WriteLine(nama);
        }

        System.Console.ReadKey();
    }
}
```

Contoh kode di atas akan menampilkan nama seluruh data yang ada pada array peserta.

Sorting

Berikut ini contoh kode untuk mengurutkan data pada array peserta.

```
var query = from n in peserta orderby n ascending select n;
```



A screenshot of a Windows Command Prompt window titled 'file:///D:/My Projects/Eraseable/ShipTra'. The window displays a list of names, each on a new line, sorted in ascending alphabetical order. The names listed are: Adam, Adi, Antono, Bedy, Benjamin, Bono, Candy, Cintia, Dedy, Endang, Evi, Faisal, Fandy, Frans, Ganto, Garfield, Hendra, Heni, Herman, Ida, Indiana Jones, Ivan, Jaina1, Jenifer, Karni.

Atau :

```
var query = from n in peserta orderby n descending select n;
```



The screenshot shows a Windows command prompt window titled 'file:///D:/My Projects/Eraseab'. The window contains a list of names, each on a new line, starting with 'Zainal' and ending with 'Lena'. The names are: Zainal, Wiwien, Uia, Uanda, Van Buren, Ujang, Tina Toon, Tiara, Thomas, Susi, Rierie, Ria, Rena, Paulina, Paula, Paul, Nia, Nene, Mya, Mia, M Reza Faisal, Lincon, Lina, Lenatan, Lena.

```
file:///D:/My Projects/Eraseab
Zainal
Wiwien
Uia
Uanda
Van Buren
Ujang
Tina Toon
Tiara
Thomas
Susi
Rierie
Ria
Rena
Paulina
Paula
Paul
Nia
Nene
Mya
Mia
M Reza Faisal
Lincon
Lina
Lenatan
Lena
```

Contoh lain :

```
var query = from n in peserta select new { namaPeserta = n, jumlahHuruf = n.Length };

foreach (var nama in query)
{
    System.Console.WriteLine(nama.namaPeserta+" ("+nama.jumlahHuruf+")");
}
```

Filter

Berikut contoh kode untuk filter data.

```
var query = from n in peserta where n == "Susi" select n;
```

Pada kode di atas diketahui array peserta bertipe data string, sehingga property dan method pada string dapat pula digunakan sebagai ekspresi seperti contoh berikut.

```
var query = from n in peserta where n.Length < 4 select n;
```

Contoh lain :

```
var query = from n in peserta where n.ToUpper().Contains("S") select n;
```

Standar Query Operator

Selain melakukan query dengan 'gaya' SQL dapat juga proses query dilakukan dengan menggunakan method-method yang terkumpul dalam suatu Standar Query Operator.

Select

Berikut contoh penggunaan operasi select.

```
var query = from n in peserta select n;
```

Dapat dituliskan menjadi berikut :

```
var query = peserta.Select(n => n);
```

Contoh lain :

```
var query = from n in peserta select new { namaPeserta = n, jumlahHuruf = n.Length };

foreach (var nama in query)
{
    System.Console.WriteLine(nama.namaPeserta + " (" + nama.jumlahHuruf + ")");
}
```

Dapat ditulis menjadi seperti berikut :

```
var query = peserta.Select(p => new { namaPeserta = p, jumlahHuruf = p.Length });

foreach (var nama in query)
{
    System.Console.WriteLine(nama.namaPeserta + " (" + nama.jumlahHuruf + ")");
}
```

Order

Berikut contoh penggunaan operasi order.

```
var query = from n in peserta orderby n ascending select n;
```

Dapat ditulis menjadi seperti berikut :

```
var query = peserta.OrderBy(p => p);
```

Contoh lain.

```
var query = from n in peserta orderby n descending select n;
```

Dapat ditulis menjadi seperti berikut :

```
var query = peserta.OrderByDescending(p => p);
```

Filter

Berikut contoh penggunaan operasi filter.

```
var query = from n in peserta where n == "Susi" select n;
```

Dapat ditulis menjadi seperti berikut :

```
var query = peserta.Where(p => p.Equals("Susi"));
```

Contoh lain :

```
var query = from n in peserta where n.Length < 4 select n;
```

Dapat ditulis menjadi seperti ini :

```
var query = peserta.Where(p => p.Length < 4);
```

Masih banyak lagi query expression atau standard query operation (SQO) yang dapat dituliskan yang nanti akan dipaparkan pada bagian selanjutnya.

Pada LINQ terdapat beberapa feature yang mempermudah para programmer yaitu :

1. LINQ to Objects.
2. LINQ to XML.
3. LINQ to DataSet.
4. LINQ to SQL.

pada bagian selanjutnya hanya akan dibahas tentang LINQ to SQL.

LINQ to SQL

LINQ to SQL dapat disebut sebagai implementasi Object Relational Mapping. Yang menjadi layer data access untuk komunikasi aplikasi dengan database.

Apa saja yang akan ditemui saat menggunakan LINQ to SQL ? berikut beberapa hal yang akan ditemui pada LINQ to SQL, yaitu :

1. Representasi database, tabel dan field pada sebuah class. Class yang seakan-akan menggantikan fungsi tabel pada suatu database. Class yang menjadi mapping ke tabel-tabel pada database. Berikut contoh class tersebut.

```
[Table(Name="dbo.PesertaTraining")]
public partial class PesertaTraining : INotifyPropertyChanging,
INotifyPropertyChanged
{
    ...

    private int _ID;
    private string _Nama;
    private string _Pekerjaan;
    private string _Alamat;
    ...

    [Column(Storage="_ID", AutoSync=AutoSync.OnInsert, DbType="Int NOT
NULL
IDENTITY", IsPrimaryKey=true, IsDbGenerated=true)]
    public int ID
    {
        get
        {
            return this._ID;
        }
        set
        {
            if ((this._ID != value))
            {
                this.OnIDChanging(value);
                this.SendPropertyChanged();
                this._ID = value;
                this.SendPropertyChanged("ID");
                this.OnIDChanged();
            }
        }
    }

    [Column(Storage="_Nama", DbType="NVarChar(100)")]
    public string Nama
    {
        get
        {
            return this._Nama;
        }
        set
        {
            if ((this._Nama != value))
            {
                this.OnNamaChanging(value);
                this.SendPropertyChanged();
                this._Nama = value;
                this.SendPropertyChanged("Nama");
                this.OnNamaChanged();
            }
        }
    }
}
```

```

        }
    }

[Column(Storage="_Pekerjaan", DbType="NVarChar(50)")]
public string Pekerjaan
{
    get
    {
        return this._Pekerjaan;
    }
    set
    {
        if ((this._Pekerjaan != value))
        {
            this.OnPekerjaanChanging(value);
            this.SendPropertyChanging();
            this._Pekerjaan = value;
            this.SendPropertyChanged("Pekerjaan");
            this.OnPekerjaanChanged();
        }
    }
}

[Column(Storage="_Alamat", DbType="NVarChar(MAX)")]
public string Alamat
{
    get
    {
        return this._Alamat;
    }
    set
    {
        if ((this._Alamat != value))
        {
            this.OnAlamatChanging(value);
            this.SendPropertyChanging();
            this._Alamat = value;
            this.SendPropertyChanged("Alamat");
            this.OnAlamatChanged();
        }
    }
}

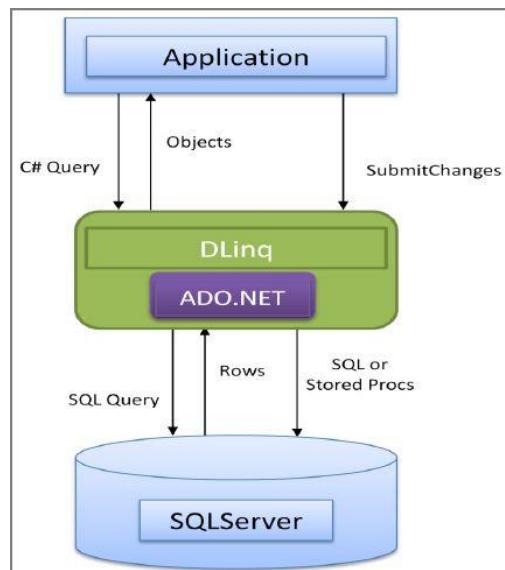
...
}

```

Class di atas merupakan representasi tabel berikut.



2. Method-method dan hal-hal lain yang memudahkan untuk operasi database seperti select, insert, update dan delete. Pada gambar berikut dapat dilihat bagaimana LINQ to SQL bekerja.

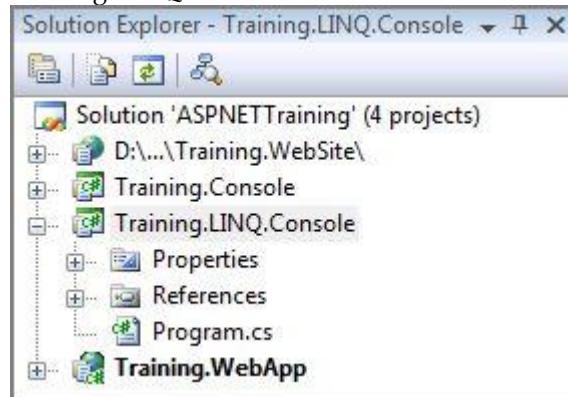


Dalam gambar ini terlihat bahwa query dari bahasa C# akan diproses oleh lapisan Dlinq (LINQ to SQL) untuk diterjemahkan menjadi Query SQL untuk kemudian dikirimkan ke basisdata. Ketika basisdata mengembalikan sekumpulan data yang berbentuk row data tersebut oleh Dlinq akan diterjemahkan menjadi objek untuk kemudian diproses pada bagian aplikasi. Ketika objek tersebut telah selesai diproses maka objek tersebut akan kembali diterjemahkan menjadi sebuah statement SQL atau stored procedure untuk disimpan di basis data.

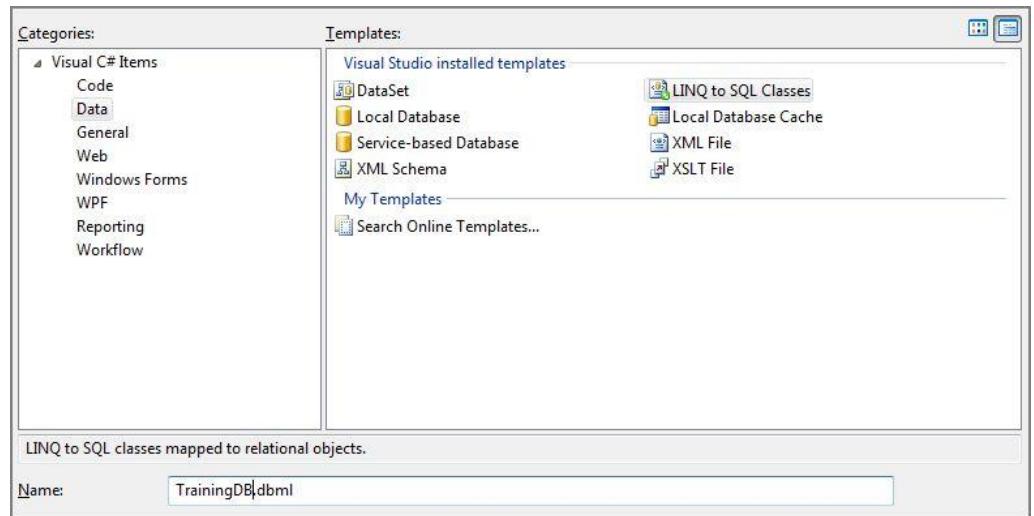
A Glance with LINQ to SQL

Berikut ini membangun aplikasi yang mengimplementasikan LINQ to SQL dengan menggunakan Visual Studio 2008. Langkah-langkah yang mesti diikuti adalah sebagai berikut.

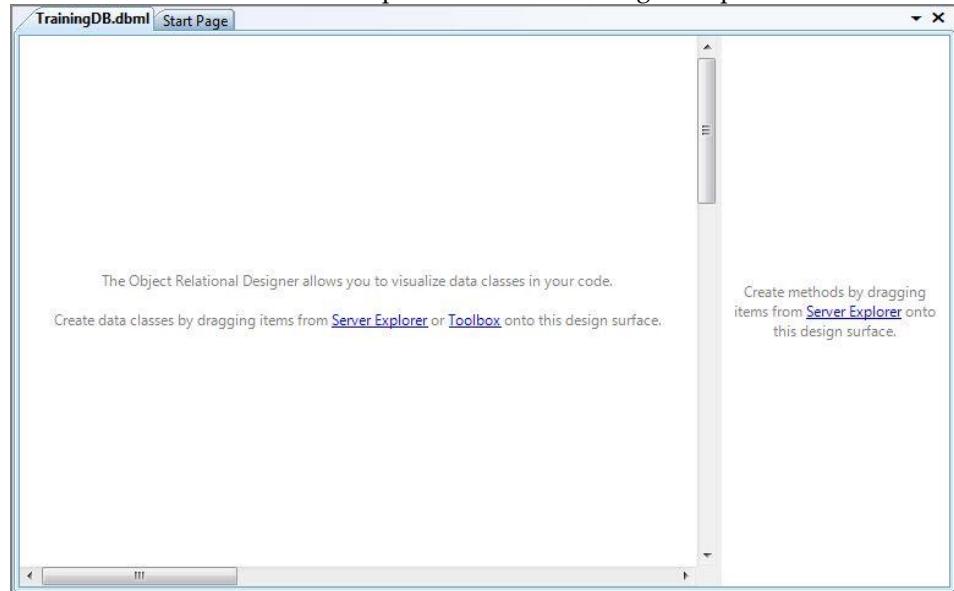
1. Tambahkan project Windows > Console Application dengan nama Training.LINQ.Console.



2. Klik kanan pada Project Training.LINQ.Console dan pilih Add > New item. Pada window Add New Item, pada kolom Categories pilih Data. Pada kolom Templates pilih LINQ to SQL Classes.

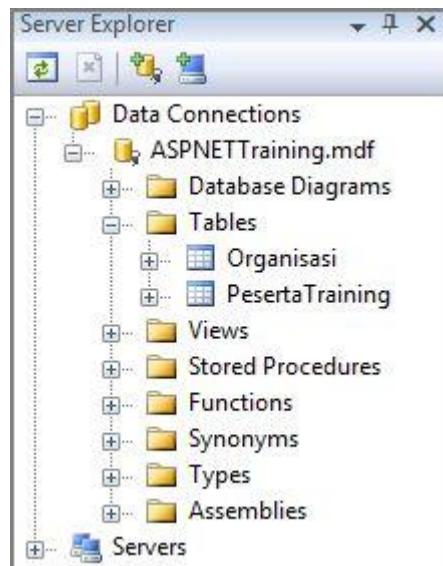


Setelah class ini dibuat maka dapat dilihat editor designer seperti berikut.

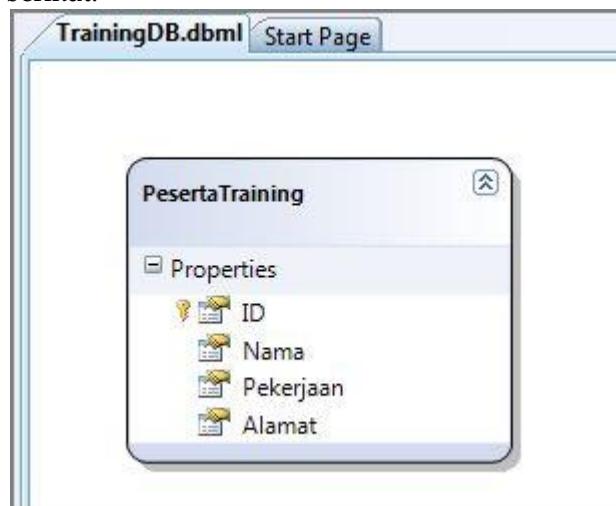


Pada antamuka designer di atas dapat di drag-n-drop tabel-tabel atau store procedure yang ada pada database dari kolom Server Explorer.

3. Pilih salah satu tabel yang ada pada database yang ditampilkan pada kolom Server Explorer. Misalnya, drag-n-drop tabel PesertaTraining pada designer TrainingDB.dbml.



Hasilnya, pada designer TrainingDB.dbml akan dapat dilihat tampilan seperti berikut.

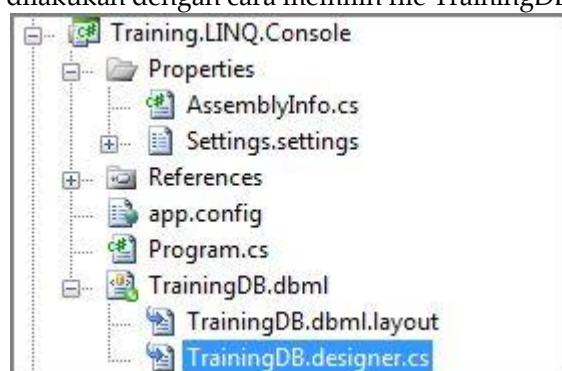


Setelah memilih tabel dari Server Explorer maka perhatikan terdapat dua file konfigurasi yaitu :

- app.config.
- Settings.settings.

Kedua file ini berfungsi untuk menyimpan connection string untuk koneksi ke database.

- Untuk melihat kode yang telah dibuat oleh designer pada TraiingDB.dbml dapat dilakukan dengan cara memilih file TrainingDB.designer.cs pada Solution Explorer.



Setelah langkah-langkah di atas diikuti maka berikut ini contoh-contoh penggunaan LINQ to SQL untuk operasi database.

Menampilkan Data

Berikut ini kode proses menampilkan data dari tabel PesertaTraining.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Training.LINQ.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            ShowData();
        }

        private static void ShowData()
        {
            TrainingDBDataContext db = new TrainingDBDataContext();

            var query = from p in db.PesertaTrainings select p;
            foreach (PesertaTraining peserta in query)
            {
                System.Console.WriteLine("ID : " + peserta.ID);
                System.Console.WriteLine("Nama : " + peserta.Nama);
                System.Console.WriteLine("Pekerjaan : " + peserta.Pekerjaan);
                System.Console.WriteLine("Alamat : " + peserta.Alamat);
                System.Console.WriteLine("-----");
            }

            System.Console.ReadKey();
        }
    }
}
```

Atau dengan kode berikut ini.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Training.LINQ.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            ShowData();
        }

        private static void ShowData()
        {
            TrainingDBDataContext db = new TrainingDBDataContext();

            var query = db.PesertaTrainings.Select(p => p);
            foreach (PesertaTraining peserta in query)
            {
                System.Console.WriteLine("ID : " + peserta.ID);
                System.Console.WriteLine("Nama : " + peserta.Nama);
                System.Console.WriteLine("Pekerjaan : " + peserta.Pekerjaan);
                System.Console.WriteLine("Alamat : " + peserta.Alamat);
                System.Console.WriteLine("-----");
            }

            System.Console.ReadKey();
        }
    }
}
```

```

    private static void InsertData()
    {
        TrainingDBDataContext db = new TrainingDBDataContext();

        PesertaTraining peserta = new PesertaTraining();
        peserta.Nama = "Aiz";
        peserta.Pekerjaan = "Dentist";
        peserta.Alamat = "Jakarta";

        db.PesertaTrainings.InsertOnSubmit(peserta);
        db.SubmitChanges();
    }
}

```

Menambah Data

Berikut contoh kode untuk menambah data.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Training.LINQ.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            InsertData();
            ShowData();
        }

        private static void ShowData()
        {
            TrainingDBDataContext db = new TrainingDBDataContext();

            var query = db.PesertaTrainings.Select(p => p);

            foreach (PesertaTraining peserta in query)
            {
                System.Console.WriteLine("ID : " + peserta.ID);
                System.Console.WriteLine("Nama : " + peserta>Nama);
                System.Console.WriteLine("Pekerjaan : " + peserta.Pekerjaan);
                System.Console.WriteLine("Alamat : " + peserta.Alamat);
                System.Console.WriteLine("-----");
            }

            System.Console.ReadKey();
        }

        private static void InsertData()
        {
            try
            {
                TrainingDBDataContext db = new TrainingDBDataContext();

                PesertaTraining peserta = new PesertaTraining();
                peserta.Nama = "Aiz";
                peserta.Pekerjaan = "Dokter";
                peserta.Alamat = "Jakarta";

                db.PesertaTrainings.InsertOnSubmit(peserta);
                db.SubmitChanges();

                System.Console.WriteLine("1 data saved.");
                System.Console.WriteLine("");
                System.Console.ReadKey();
            }
            catch (Exception ex)
            {
                System.Console.WriteLine(ex.Message);
            }
        }
    }
}

```

```
        }
    }
}
```

Mengupdate Data

Berikut ini contoh penggunaan LINQ to SQL untuk proses update data.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Training.LINQ.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            UpdateData();
            ShowData();
        }

        private static void ShowData()
        {
            TrainingDBDataContext db = new TrainingDBDataContext();

            var query = db.PesertaTrainings.Select(p => p);

            foreach (PesertaTraining peserta in query)
            {
                System.Console.WriteLine("ID : " + peserta.ID);
                System.Console.WriteLine("Nama : " + peserta>Nama);
                System.Console.WriteLine("Pekerjaan : " + peserta.Pekerjaan);
                System.Console.WriteLine("Alamat : " + peserta.Alamat);
                System.Console.WriteLine("-----");
            }

            System.Console.ReadKey();
        }

        private static void UpdateData()
        {
            try
            {
                TrainingDBDataContext db = new TrainingDBDataContext();

                PesertaTraining peserta = db.PesertaTrainings.First(p => p.ID == 6);
                peserta>Nama = "Siti Aisyah";
                peserta.Pekerjaan = "Dokter Gigi";

                db.SubmitChanges();

                System.Console.WriteLine("1 data update.");
                System.Console.WriteLine("");
                System.Console.ReadKey();
            }
            catch (Exception ex)
            {
                System.Console.WriteLine(ex.Message);
                System.Console.ReadKey();
            }
        }
    }
}
```

Menghapus data

Berikut adalah contoh penggunaan LINQ to SQL untuk menghapus data.

```
using System;
```

```

using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace Training.LINQ.Console
{
    class Program
    {
        static void Main(string[] args)
        {
            DeleteData();
        }

        private static void DeleteData()
        {
            try
            {
                TrainingDBDataContext db = new TrainingDBDataContext();

                PesertaTraining peserta = db.PesertaTrainings.First(p => p.ID == 7);
                db.PesertaTrainings.DeleteOnSubmit(peserta);
                db.SubmitChanges();

                System.Console.WriteLine("1 data deleted.");
                System.Console.WriteLine("");
                System.Console.ReadKey();
            }
            catch (Exception ex)
            {
                System.Console.WriteLine(ex.Message);
            }
        }
    }
}

```

Data Binding pada ASP.NET

Pada bagian akan diberikan contoh-contoh untuk memasukkan data pada control-control yang ada pada ASP.NET.

Contoh-contoh yang akan diberikan akan menggunakan ADO.NET dan LINQ to SQL.

Standard Control

Berikut contoh-contoh memasukkan data pada Standard Control.

DropDownList

Berikut contoh bind data dengan menggunakan ADO.NET

```

DropDownListExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DropDownListExample.aspx.cs"
Inherits="DataBinding_DropDownListExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            With ADO.NET :
            <asp:DropDownList ID="DropDownList ADONET" runat="server">

```

```

</asp:DropDownList>
</div>
</form>
</body>
</html>

```

Code file :

```

DropDownListExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using System.Data.SqlClient;
using System.Collections.Generic;

public partial class DataBinding_DropDownListExample : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["SQLExpressConnString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        DropDownList_ADONET.DataBind();
    }

    protected void DropDownList_ADONET_DataBind()
    {
        string strSQL = "SELECT * FROM Organisasi";

        SqlDataAdapter da = new SqlDataAdapter(strSQL, connStr);
        DataTable dt = new DataTable();
        da.Fill(dt);

        DropDownList_ADONET.DataSource = dt;
        DropDownList_ADONET.DataTextField = "OrganisasiNama";
        DropDownList_ADONET.DataValueField = "OrganisasiID";
        DropDownList_ADONET.DataBind();
    }
}

```

Berikut contoh bind data dengan menggunakan LINQ to SQL.

```

DropDownListExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DropDownListExample.aspx.cs"
Inherits="DataBinding_DropDownListExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            With LINQ to SQL :
            <asp:DropDownList ID="DropDownList_LINQ" runat="server">
            </asp:DropDownList>
        </div>
    </form>
</body>
</html>

```

Code file :

```

DropDownListExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using System.Data.SqlClient;
using System.Collections.Generic;

public partial class DataBinding_DropDownListExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        DropDownList_LINQ_DataBind();
    }

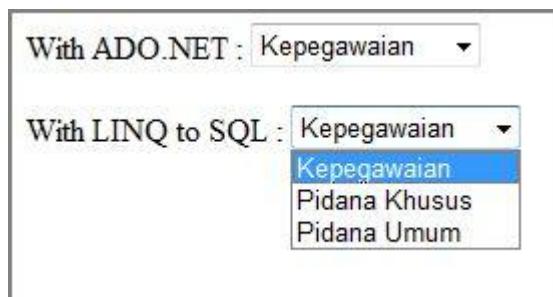
    protected void DropDownList_LINQ_DataBind()
    {
        TrainingDBDataContext db = new TrainingDBDataContext();

        var query = from o in db.Organisasis select o;
        List<Organisasi> data = query.ToList();

        DropDownList_LINQ.DataSource = data;
        DropDownList_LINQ.DataTextField = "OrganisasiNama";
        DropDownList_LINQ.DataValueField = "OrganisasiID";
        DropDownList_LINQ.DataBind();
    }
}

```

Berikut hasil dari kode di atas :



RadioButtonList

Berikut contoh binding data pada control RadioButtonList.

```

RadioButtonListExample.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="RadioButtonListExample.aspx.cs"
Inherits="DataBinding_RadioButtonListExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">

```

```

<div>
    With ADO.NET :
    <asp:RadioButtonList ID="RadioButtonList_ADO.NET"
        RepeatDirection="Horizontal" runat="server">
    </asp:RadioButtonList>
    <br /><br />
    With LINQ to SQL :
    <asp:RadioButtonList ID="RadioButtonList_LINQ" runat="server">
    </asp:RadioButtonList>
</div>
</form>
</body>
</html>

```

Code file :

```

RadioButtonListExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using System.Data.SqlClient;
using System.Collections.Generic;

public partial class DataBinding_RadioButtonListExample : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["SQLEXPRESSConnString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        RadioButtonList_ADO.NET_DataBind();
        RadioButtonList_LINQ_DataBind();
    }

    protected void RadioButtonList_ADO.NET_DataBind()
    {
        string strSQL = "SELECT * FROM Organisasi";

        SqlDataAdapter da = new SqlDataAdapter(strSQL, connStr);
        DataTable dt = new DataTable();
        da.Fill(dt);

        RadioButtonList_ADO.NET.DataSource = dt;
        RadioButtonList_ADO.NET.DataTextField = "OrganisasiNama";
        RadioButtonList_ADO.NET.DataValueField = "OrganisasiID";
        RadioButtonList_ADO.NET.DataBind();
    }

    protected void RadioButtonList_LINQ_DataBind()
    {
        TrainingDBDataContext db = new TrainingDBDataContext();

        var query = from o in db.Organisasis select o;
        List<Organisasi> data = query.ToList();

        RadioButtonList_LINQ.DataSource = data;
        RadioButtonList_LINQ.DataTextField = "OrganisasiNama";
        RadioButtonList_LINQ.DataValueField = "OrganisasiID";
        RadioButtonList_LINQ.DataBind();
    }
}

```

Berikut adalah hasil dari kode di atas.

With ADO.NET :

Kepegawaian Pidana Khusus Pidana Umum

With LINQ to SQL :

Kepegawaian
 Pidana Khusus
 Pidana Umum

CheckBoxList

Berikut adalah contoh bind data pada CheckBoxList.

CheckBoxListExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CheckBoxListExample.aspx.cs"
Inherits="DataBinding_CheckBoxListExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            with ADONET :
            <asp:CheckBoxList ID="CheckBoxList_ADONET" RepeatDirection="Horizontal"
                runat="server">
            </asp:CheckBoxList>
            <br />
            with LINQ to SQL :
            <asp:CheckBoxList ID="CheckBoxList_LINQ" runat="server">
            </asp:CheckBoxList>
        </div>
    </form>
</body>
</html>
```

Code file :

CheckBoxListExample.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

using System.Data.SqlClient;
using System.Collections.Generic;

public partial class DataBinding_CheckBoxListExample : System.Web.UI.Page
{
    string connStr =
ConfigurationManager.ConnectionStrings["SQLExpressConnectionString"].ToString();

    protected void Page_Load(object sender, EventArgs e)
    {
        CheckBoxList_ADONET_DataBind();
        CheckBoxList_LINQ_DataBind();
    }
}
```

```

protected void CheckBoxList_ADONET_DataBind()
{
    string strSQL = "SELECT * FROM Organisasi";
    SqlConnection conn = new SqlConnection(connStr);
    SqlCommand cmd = new SqlCommand(strSQL, conn);
    ListItemCollection items = new ListItemCollection();

    conn.Open();
    SqlDataReader dr = cmd.ExecuteReader();

    while (dr.Read())
    {
        ListItem item = new ListItem();
        item.Value = dr["OrganisasiID"].ToString();
        item.Text = dr["OrganisasiNama"].ToString();

        items.Add(item);
    }

    CheckBoxList_ADONET.DataSource = items;
    CheckBoxList_ADONET.DataTextField = "Text";
    CheckBoxList_ADONET.DataValueField = "Value";
    CheckBoxList_ADONET.DataBind();
}

protected void CheckBoxList_LINQ_DataBind()
{
    TrainingDBDataContext db = new TrainingDBDataContext();

    var query = from o in db.Organisasisis select o;
    List<Organisasi> data = query.ToList();

    ListItemCollection items = new ListItemCollection();

    for (int i = 0; i < data.Count; i++)
    {
        Organisasi org = (Organisasi)data[i];

        ListItem item = new ListItem();
        item.Value = org.OrganisasiID;
        item.Text = org.OrganisasiNama;

        items.Add(item);
    }

    CheckBoxList_LINQ.DataSource = items;
    CheckBoxList_LINQ.DataTextField = "Text";
    CheckBoxList_LINQ.DataValueField = "Value";
    CheckBoxList_LINQ.DataBind();
}
}

```

Hasilnya adalah sebagai berikut :

with ADONET :

Kepgawaian Pidana Khusus Pidana Umum

with LINQ to SQL :

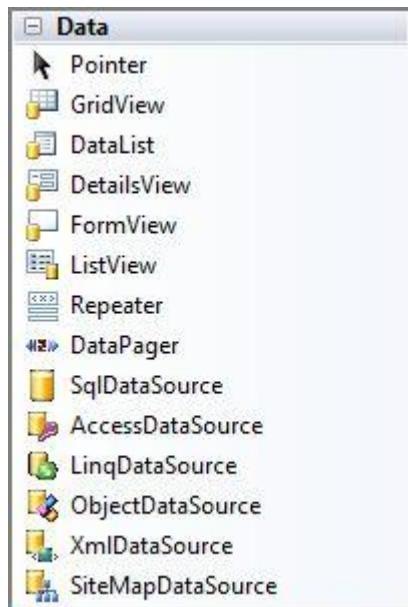
Kepgawaian

Pidana Khusus

Pidana Umum

Data Control

Pada toolbox di Visual Studio terdapat tab Data yang berisi control-control yang berhubungan dengan menampilkan data, menambah data, update data dan juga menghapus data. Berikut ini adalah control-control yang ada pada tab Data pada toolbox.



Control-control di atas dapat dikelompokkan menjadi berikut :

1. Sumber Data, yaitu control control yang menjadi sumber data yang akan digunakan control-control lain. Data source akan melakukan koneksi kepada database server, file MS Access, file XML dan lain-lain. yang termasuk dalam kelompok ini adalah :
 - a. SqlDataSource.
 - b. AccessDataSource.
 - c. LinqDataSource.
 - d. ObjectDataSource.
 - e. XmlDataSource.
 - f. SiteMapDataSource.
2. Operasi Data, yaitu control yang akan menampilkan data atau modifikasi data. Control-control yang termasuk dalam kelompok ini adalah :
 - a. GridView.
 - b. DropDownList.
 - c. Repeater.
 - d. ListView.
 - e. DetailView.
 - f. FormView.
3. Pager, yaitu control yang berfungsi sebagai pager yang dapat digunakan oleh control-control pada kelompok Operasi Data.

Untuk menampilkan data atau melakukan operasi data dengan menggunakan control-control ini dapat dilakukan dengan dua cara, yaitu :

1. Cara otomatis, menggunakan fasilitas yang ada pada Visual Studio.
2. Cara manual, membuat design control secara manual.

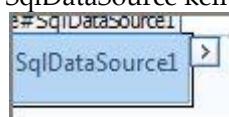
Cara Otomatis

Berikut ini akan diberikan contoh penggunaan sebagain control di atas dengan bantuan fasilitas yang dimiliki oleh Visual Studio.

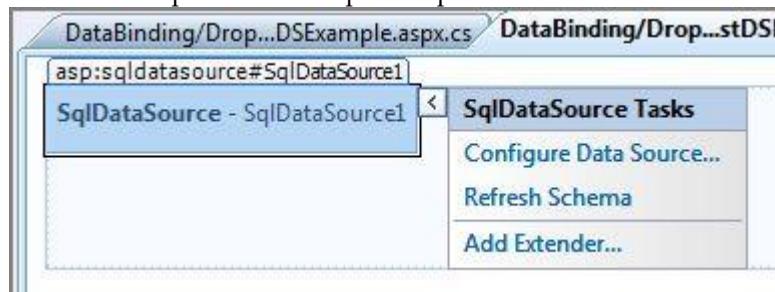
Menampilkan Data pada DropDownList dengan menggunakan SqlDataSource sebagai sumber data

Berikut adalah langkah-langkah yang harus diikuti untuk membuat dan mengkonfigurasi SqlDataSource :

1. Tambahkan halaman Web Form baru dengan nama DropDownListDSExample.aspx.
2. Pada editor ubah menjadi ke mode desing.
3. Drag control SqlDataSource dan drop pada Web Form. Klik pada control SqlDataSource kemudian pilih tanda berikut.



Maka akan dapat dilihat tampilan seperti berikut :

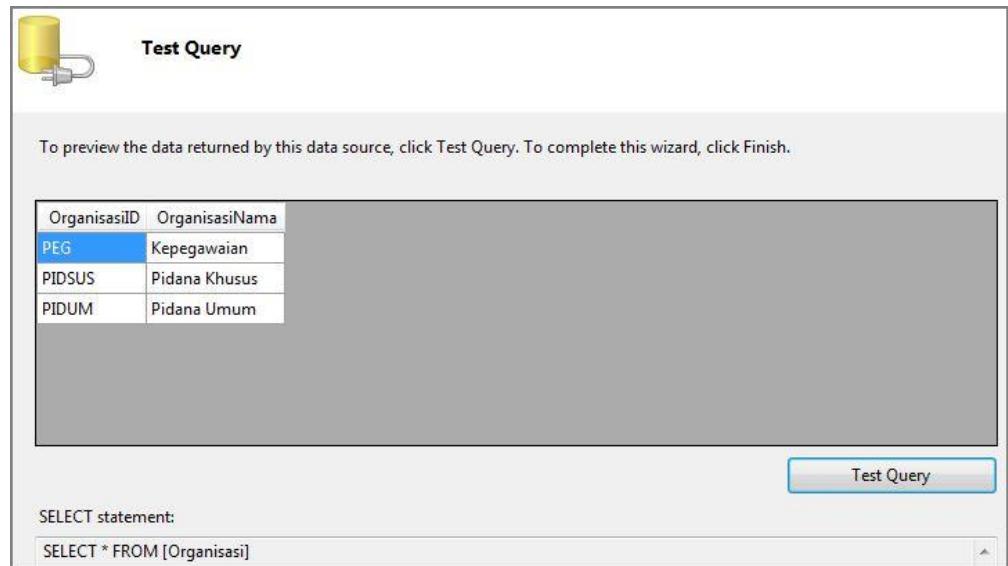


Pilih Configure Data Source.

4. Pada window Configure Data Source, koneksi pada database dapat dilakukan dengan menggunakan Connection String yang sudah ada atau dapat membuat Connection String baru dengan mengklik tombol New Connection.



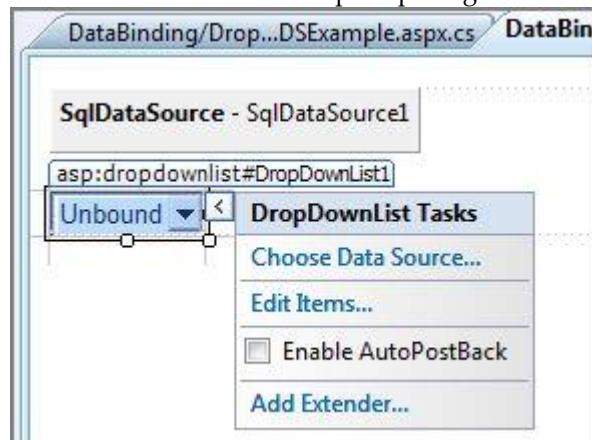
5. Kemudian klik tombol Next, dan berikutnya akan ditampilkan window untuk memilih tabel pada database yang akan digunakan. Pilih Next untuk melanjutkan.
6. Pada window Test Query berfungsi untuk menampilkan data dari query yang telah dibuat pada window sebelumnya.



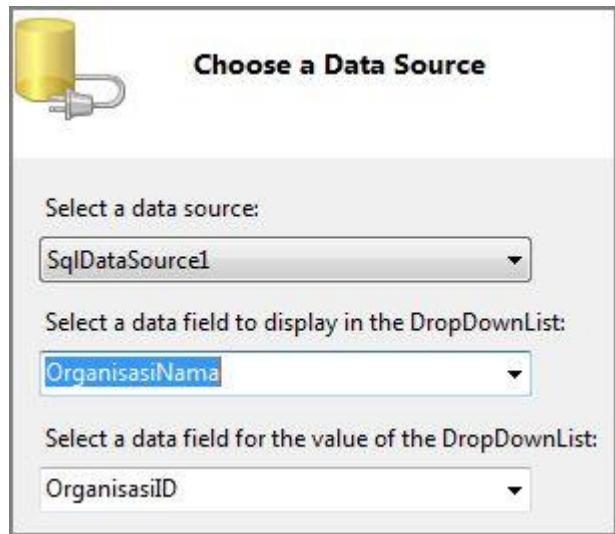
7. Klik tombol Finish.

Berikut ini langkah-langkah yang mesti diikuti untuk menampilkan data pada DropDownList dengan memanfaatkan SqlDataSource yang telah dibuat :

1. Drag control DropDownList dan drop pada Web Form.
2. Pilih Choose Data Source seperti pada gambar berikut.



3. Pada window Choose a Data Source, pilih nama SqlDataSource yang telah dibuat sebelumnya kemudian tentukan data field yang akan digunakan sebagai value dan text pada control DropDownList.



Menampilkan Data dengan GridView

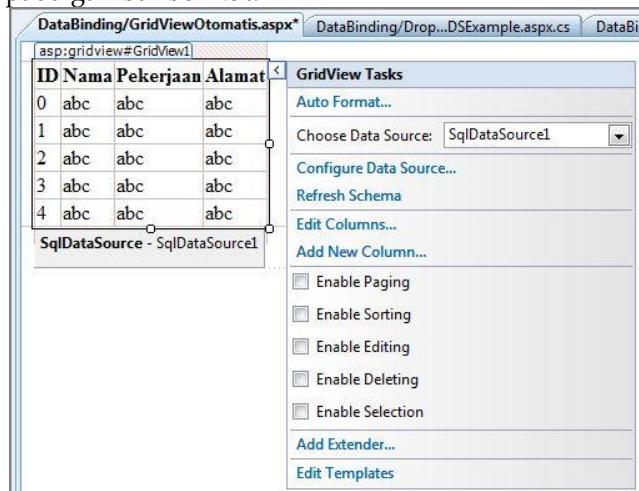
Berikut ini contoh menampilkan data dengan GridView dengan cara otomatis memanfaatkan fasilitas dari Visual Studio.

Berikut langkah-langkah yang mesti diikuti, yaitu :

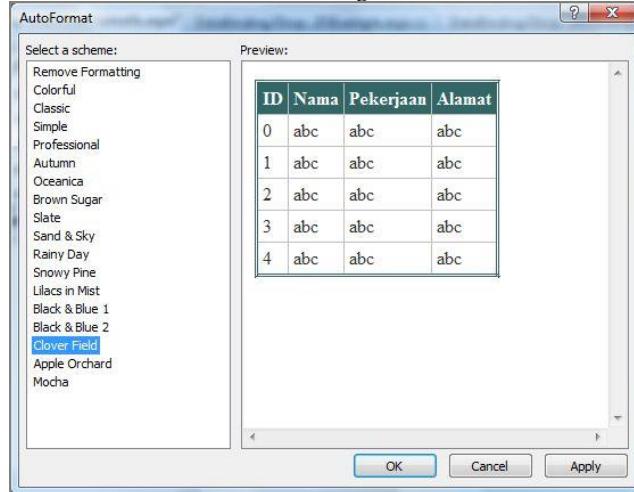
1. Tambahkan halaman Web Form dengan nama GridViewOtomatis.aspx, ubah mode editor ke mode Design.
2. Pada Visual Studio, aktifkan kolom Server Explorer menjadi seperti berikut.



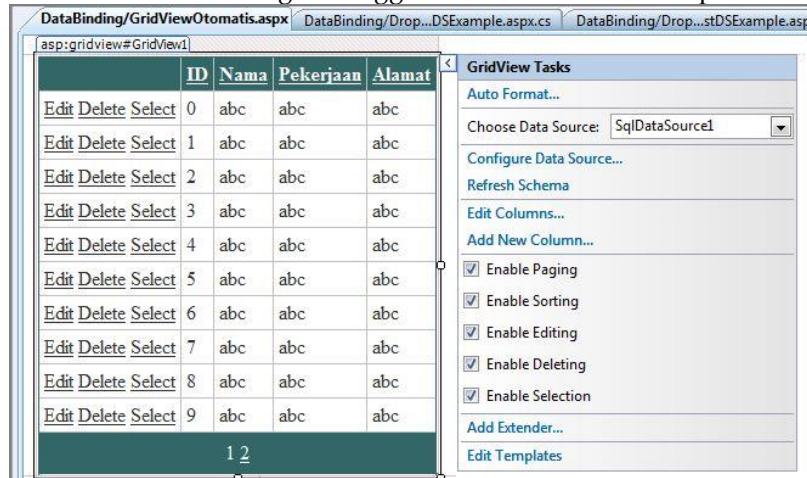
3. Drag Tabel PesertaTraining ke halaman Web Form. Maka akan ditampilkan seperti pada gambar berikut.



4. Pilih Auto Format untuk mengubah theme antarmuka GridView.



5. Centang checkbox Enable Paging, Enable Sorting, Enable Editing, Enable Deleting, Enable Selection bila ingin menggunakan fasilitas tersebut pada GridView.

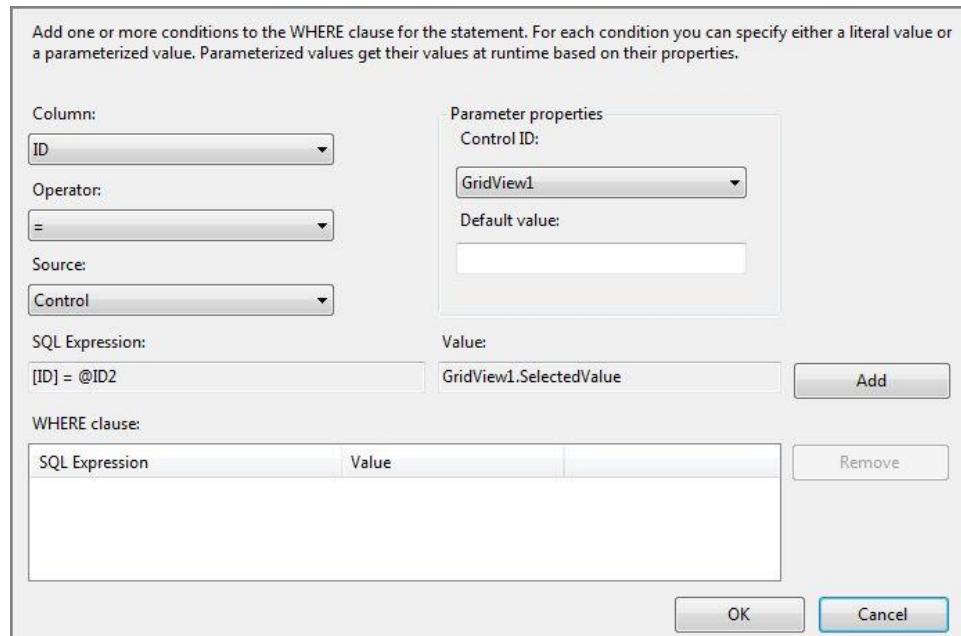


Menggunakan DetailView untuk menampilkan detail data dari record yang dipilih pada GridView

Kasus ini masih menggunakan Web Form yang telah dibuat sebelumnya, yaitu GridViewOtomatis.aspx.

Pada kasus ini akan ditambahkan control DetailView untuk menampilkan detail data dari record yang dipilih pada GridView. Berikut langkah-langkah yang mesti dilakukan, yaitu :

1. Tambahkan control SqlDataSource baru pada Web Form.
2. Pilih Configure Data Source. Pilih Connection String yang sama dengan control SqlDataSource yang digunakan oleh GridView. Klik Next.
3. Pada window Configure the Select Statement pilih tabel PesertaTraining. Klik tombol WHERE... untuk melakukan filter.



Isi nilai-nilai pada window Add Where Clause seperti pada gambar di atas, kemudian klik tombol Add, kemudian klik tombol OK.

4. Klik tombol Next dan akhiri dengan mengklik tombol Finish.

Hasilnya bisa dilihat seperti pada gambar berikut ini.

ID	Nama	Pekerjaan	Alamat
Edit Delete Select 1	Reza	Trainer	Jl. Cisitu Lama No. 13
Edit Delete Select 2	Faisal	Developer	Jl. Lama Cisitu No. 31

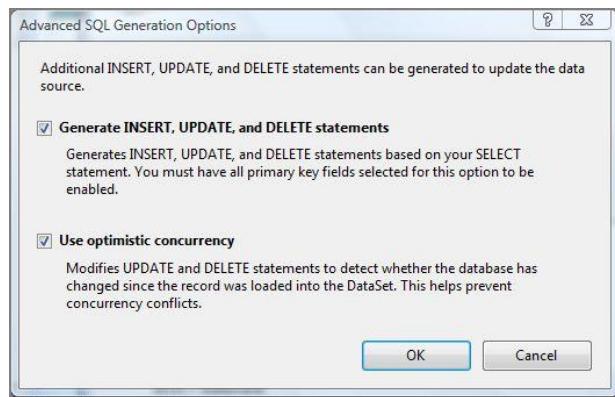
1	2	3
---	---	---

ID	2
Nama	Faisal
Pekerjaan	Developer
Alamat	Jl. Lama Cisitu No. 31

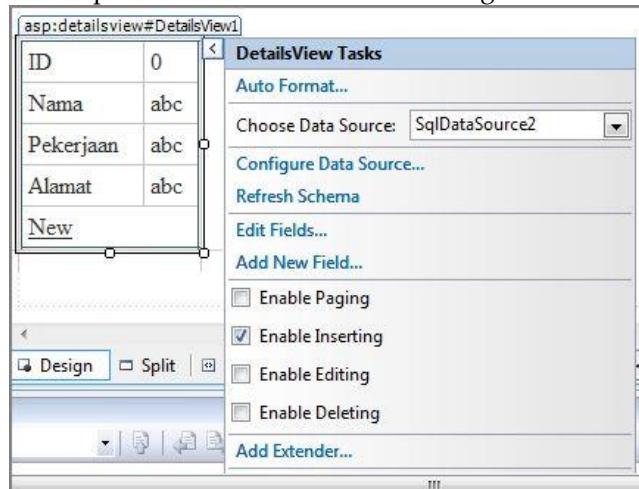
Menambah Data dengan DetailView

Untuk membuat contro DetailView pada contoh di atas dapat mempunyai fungsi menambah data maka langkah yang perlu dilakukan adalah sebagai berikut :

1. Klik control SqlDataSource1 yang ada pada Web Form dan pilih Configure Data Source.
2. Ketika pada window Configure the Select Statement klik tombol Advanced... Kemudian centang checkbox seperti pada gambar di bawah ini.



3. Dengan menambahkan feature di atas maka control DetailView mempunyai kemampuan untuk menambah data dengan cara mencantang Enable Inserting.



4. Maka pada control DetailView dapat dilihat tambahan LinkButton New untuk menambah data.

Repeater

Secara default control ini tidak mempunyai feature Pager seperti yang dilihat pada GridView. Selain itu control ini juga tidak mempunyai template antarmuka yang secara otomatis dibuat oleh Visual Studio.

Untuk membuat antarmuka dalam menampilkan data mesti dilakukan secara manual, sehingga antarmuka pada control Repeater dapat lebih bebas tergantung keinginan.

Berikut contoh penggunaan Repater.

```
RepeaterExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="RepeaterExample.aspx.cs"
Inherits="DataBinding_RepeaterExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Repeater ID="Repeater_PesertaTraining" runat="server">
                <HeaderTemplate>
                    <table width="100%" cellpadding="0" cellspacing="0">
                </HeaderTemplate>
                <ItemTemplate>
```

```

        <tr>
            <td colspan="3" style="font-size:23px;font-weight:bold">
                <%# Eval("Nama") %> [<%# Eval("NIP") %>]
            </td>
        </tr>
        <tr>
            <td style="width:100px">Organisasi</td>
            <td style="width:13px">:</td>
            <td style="font-weight:bold">
                <%# GetOrganisasiName(Convert.ToString(Eval("OrganisasiID")))>
            </td>
        </tr>
        <tr>
            <td colspan="3">
                <asp:Label ID="Label1" runat="server" Text='<%# Eval("Alamat")%>'>
            </asp:Label>
            </td>
        </tr>
    </ItemTemplate>
    <SeparatorTemplate>
        <tr>
            <td colspan="3"><hr /></td>
        </tr>
    </SeparatorTemplate>
    <FooterTemplate>
        </table>
    </FooterTemplate>
</asp:Repeater>
</div>
</form>
</body>
</html>

```

Code file :

```

RepeaterExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataBinding_RepeaterExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Repeater_PesertaTraining_Databind();
    }

    protected void Repeater_PesertaTraining_Databind()
    {
        TrainingDBDataContext db = new TrainingDBDataContext();

        var query = db.Pegawai.Select(p => p);

        Repeater_PesertaTraining.DataSource = query;
        Repeater_PesertaTraining.DataBind();
    }

    protected string GetOrganisasiName(string id)
    {
        try
        {
            TrainingDBDataContext db = new TrainingDBDataContext();
            Organisasi organisasi = db.Organisasies.Where(o => o.OrganisasiID ==
id).Single<Organisasi>();
        }
    }
}

```

```

        return organisasi.OrganisasiNama;
    }
    catch { }
    return "-";
}
}

```

DataList

Feature yang dimiliki oleh DataList lebih banyak bila dibandingkan yang dimiliki oleh control Repeater. DataList memiliki banyak feature yang sangat membantu seperti yang dimiliki oleh GridView dengan minus Pager.

Berikut ini contoh penggunaan DataList.

DataSetExample.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="DataSetExample.aspx.cs"
Inherits="DataBinding_DataSetExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:DataSet ID="ItemsList"
                GridLines="Both"
                CellPadding="3"
                CellSpacing="0"
                OnItemCommand="Item_Command"
                runat="server">
                <HeaderStyle BackColor="#aaaadd" %>
                <AlternatingItemStyle BackColor="Gainsboro" %>
                <SelectedItemStyle BackColor="Yellow" %>
                <HeaderTemplate>
                    NIP
                </HeaderTemplate>
                <ItemTemplate>
                    <asp:LinkButton ID="Button_Select" Text="Select" CommandName="Select"
                        runat="server" />
                    NIP
                    <%# DataBinder.Eval(Container.DataItem, "NIP") %>
                </ItemTemplate>
                <SelectedItemTemplate>
                    NIP:
                    <asp:Label ID="Label_NIP"
                        Text='<%# DataBinder.Eval(Container.DataItem, "NIP") %>' %>
                    <br />
                    Nama:
                    <asp:Label ID="Label_Nama"
                        Text='<%# DataBinder.Eval(Container.DataItem, "Nama") %>' %>
                    <br />
                    Organisasi:
                    <asp:Label ID="Label_Organisasi"
                        Text='<%# DataBinder.Eval(Container.DataItem, "OrganisasiID") %>' %>
                </SelectedItemTemplate>
            </asp:DataSet>
        </div>
    </form>
</body>
</html>

```

Code file :

DataSetExample.aspx.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataBinding_DataListExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            ItemsList.DataSource = CreatePegawaiDataSource();
            ItemsList.DataBind();
        }
    }

    protected IList CreatePegawaiDataSource()
    {
        TrainingDBDataContext db = new TrainingDBDataContext();
        var query = db.Pegawais.Select(p => p);

        return query.ToList();
    }

    protected void Item_Command(Object sender, DataListCommandEventArgs e)
    {
        ItemsList.SelectedIndex = e.Item.ItemIndex;

        ItemsList.DataSource = CreatePegawaiDataSource();
        ItemsList.DataBind();
    }
}

```

ListView

Control ini seperti DataList tetapi mempunyai fasilitas Pager dengan menggunakan control DataPager.

Visual Studio juga memberikan kemudahan untuk penggunaan control ini.

Berikut contoh sederhana penggunaan control ListView.

```

ListViewExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ListViewExample.aspx.cs"
Inherits="DataBinding_ListViewExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:SqlDataSource ID="SqlDataSource1" runat="server" />

            <asp:ListView ID="ListView_Pegawai" DataSourceID="SqlDataSource1"
                DataKeyNames="NIP" runat="server">
                <LayoutTemplate>

```

```

<asp:PlaceHolder ID="itemPlaceholder"
runat="server"></asp:PlaceHolder>
    <asp:DataPager ID="DataPager1" PageSize="3" runat="server">
        <Fields>
            <asp:NumericPagerField
                ButtonCount="5"
                PreviousPageText="---"
                NextPageText="-->" />
        </Fields>
    </asp:DataPager>
</LayoutTemplate>
<ItemTemplate>
    <%# Eval("Nama") %>
    <br />
</ItemTemplate>
</asp:ListView>
</div>
</form>
</body>
</html>

```

Code file :

```

ListViewExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataBinding_ListViewExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlDataSource1.ConnectionString =
ConfigurationManager.ConnectionStrings["SQLEXPRESSConnString"].ToString();
        SqlDataSource1.SelectCommand = "SELECT * FROM [Pegawai]";
    }
}

```

Contoh di atas adalah contol sederhana menampilkan data dengan fasilitas Pager.

Untuk membuat contoh di atas mempunyai fasilitas sorting cukup dengan menambahkan baris berikut pada LayOutTemplate.

```

<asp:LinkButton ID="LinkButton1"
    CommandName="Sort"
    CommandArgument="Nama"
    runat="server">
    Sort
</asp:LinkButton>

```

Sehingga kode ListViewExample.aspx menjadi sebagai berikut :

```

ListViewExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ListViewExample.aspx.cs"
Inherits="DataBinding_ListViewExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>

```

```

</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:SqlDataSource ID="SqlDataSource1" runat="server" />

            <asp:ListView ID="ListView_Pegawai" DataSourceID="SqlDataSource1"
                DataKeyNames="NIP" runat="server">
                <LayoutTemplate>
                    <asp:LinkButton ID="LinkButton1"
                        CommandName="Sort"
                        CommandArgument="Nama"
                        runat="server">
                        Sort
                    </asp:LinkButton>
                    <br />
                    <asp:PlaceHolder ID="itemPlaceholder"
                        runat="server"></asp:PlaceHolder>
                    <asp:DataPager ID="DataPager1" PageSize="13" runat="server">
                        <Fields>
                            <asp:NumericPagerField
                                ButtonCount="5"
                                PreviousPageText="---"
                                NextPageText="-->" />
                        </Fields>
                    </asp:DataPager>
                </LayoutTemplate>
                <ItemTemplate>
                    <%# Eval("Nama") %>
                    <br />
                </ItemTemplate>
            </asp:ListView>
        </div>
    </form>
</body>
</html>

```

ListView mempunyai fasilitas untuk grouping seperti pada contoh di bawah ini :

ListViewGroupExample.aspx

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ListViewGroupExample.aspx.cs"
Inherits="DataBinding.ListViewGroupExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:SqlDataSource ID="SqlDataSource1" runat="server" />

            <asp:ListView ID="ListView_Pegawai"
                DataSourceID="SqlDataSource1"
                DataKeyNames="NIP"
                GroupItemCount="2"
                runat="server">
                <LayoutTemplate>
                    <asp:PlaceHolder ID="groupPlaceholder"
                        runat="server"></asp:PlaceHolder>
                    <br /><br />
                    <asp:DataPager ID="DataPager1" PageSize="13" runat="server">
                        <Fields>
                            <asp:NumericPagerField
                                ButtonCount="5"
                                PreviousPageText="---"
                                NextPageText="-->" />
                        </Fields>
                    </asp:DataPager>
                </LayoutTemplate>
                <GroupTemplate>

```

```

        <asp:PlaceHolder ID="itemPlaceholder"
runat="server"></asp:PlaceHolder>
        </GroupTemplate>
        <GroupSeparatorTemplate>
            <hr />
        </GroupSeparatorTemplate>
        <ItemTemplate>
            <%# Eval("Nama") %>
        </ItemTemplate>
        <ItemSeparatorTemplate>
            <br />
        </ItemSeparatorTemplate>
    </asp:ListView>
</div>
</form>
</body>
</html>

```

Code file :

```

ListViewGroupExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataBinding_GroupExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlDataSource1.ConnectionString =
ConfigurationManager.ConnectionStrings["SQLEXPRESSConnString"].ToString();
        SqlDataSource1.SelectCommand = "SELECT * FROM [Pegawai]";
    }
}

```

Berikut ini contoh pembuatan fungsi pemilihan record.

```

ListViewExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ListViewExample.aspx.cs"
Inherits="DataBinding_GroupExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>

            <asp:SqlDataSource ID="SqlDataSource1" runat="server" />

            <asp:ListView ID="ListView_Pegawai" DataSourceID="SqlDataSource1"
                DataKeyNames="NIP" runat="server"
                onselectedindexchanged="ListView_Pegawai_SelectedIndexChanged">
                <LayoutTemplate>
                    <asp:LinkButton ID="LinkButton1"
                        CommandName="Sort"
                        CommandArgument="Nama"
                        runat="server">
                        Sort
                    </asp:LinkButton>
                    <br />
                <asp:PlaceHolder ID="itemPlaceholder">

```

```

    runat="server">></asp:PlaceHolder>
        <asp:DataPager ID="DataPager1" PageSize="13" runat="server">
            <Fields>
                <asp:NumericPagerField
                    ButtonCount="5"
                    PreviousPageText="---"
                    NextPageText="-->" />
            </Fields>
        </asp:DataPager>
    </LayoutTemplate>
    <SelectedItemTemplate>
        <asp:LinkButton ID="LinkButton2" CommandName="Select"
            runat="server">Select</asp:LinkButton>
        <b><%# Eval("Nama") %></b>
        <br />
    </SelectedItemTemplate>
    <ItemTemplate>
        <asp:LinkButton ID="LinkButton2" CommandName="Select"
            runat="server">Select</asp:LinkButton>
        <%# Eval("Nama") %>
        <br />
    </ItemTemplate>
</asp:ListView>
<br />

        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    </div>
</form>
</body>
</html>

```

Code file :

```

ListViewExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataBinding_ListViewExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        SqlDataSource1.ConnectionString =
ConfigurationManager.ConnectionStrings["SQLEXPRESSConnectionString"].ToString();
        SqlDataSource1.SelectCommand = "SELECT * FROM [Pegawai]";
    }

    protected void ListView_Pegawai_SelectedIndexChanged(object sender, EventArgs e)
    {
        Label1.Text = Convert.ToString(ListView_Pegawai.SelectedValue);
    }
}

```

GridView

Control ini merupakan control yang paling mudah dan paling lengkap untuk digunakan. Tetapi antarmuka control ini tidak sebebas control-control sebelumnya.

Berikut adalah contoh paling sederhana penggunaan control GridView.

```

GridViewExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="GridViewExample.aspx.cs"
Inherits="DataBinding_GridViewExample" %>

```

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView_Pegawai" runat="server">
            </asp:GridView>
        </div>
    </form>
</body>
</html>

```

Code file :

```

GridViewExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataBinding_GridViewExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        GridView_Pegawai_Databind();
    }

    protected void GridView_Pegawai_Databind()
    {
        TrainingDBDataContext db = new TrainingDBDataContext();

        var query = db.Pegawais.Select(p => p);

        GridView_Pegawai.DataSource = query;
        GridView_Pegawai.DataBind();
    }
}

```

Berikut ini contoh lain penggunaan control GridView.

```

GridViewExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="GridViewExample.aspx.cs"
Inherits="DataBinding_GridViewExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:LinqDataSource ID="LinqDataSource1" runat="server"/>

            <asp:GridView ID="GridView_Pegawai"
                DataSourceID="LinqDataSource1"
                DataKeyNames="NIP"
                AutoGenerateColumns="false"
                AllowSorting="true"
            >
            </asp:GridView>
        </div>
    </form>
</body>
</html>

```

```

        runat="server"
        onselectedindexchanged="GridView_Pegawai_SelectedIndexChanged">
        <SelectedRowStyle BackColor="Gray" Font-Bold="true" />
        <Columns>
            <asp:ButtonField DataTextField="NIP" CommandName="Select"
                SortExpression="NIP" HeaderText="NIP" />
            <asp:BoundField DataField="Nama"
                SortExpression="Nama" HeaderText="Nama" />
            <asp:TemplateField HeaderText="Organisasi"
                SortExpression="OrganisasiID">
                <ItemTemplate>
                    <%# GetOrganisasiName(Convert.ToString(Eval("OrganisasiID")))%>
                </ItemTemplate>
            </asp:TemplateField>
        </Columns>
    </asp:GridView>
    <br />
    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>

```

Code file :

```

GridViewExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class DataBinding_GridViewExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        LinqDataSource1.ContextTypeName = "TrainingDBDataContext";
        LinqDataSource1.TableName = "Pegawai";
    }

    protected void GridView_Pegawai_SelectedIndexChanged(object sender, EventArgs e)
    {
        Label1.Text = Convert.ToString(GridView_Pegawai.SelectedValue);
    }

    protected string GetOrganisasiName(string orgID)
    {
        try
        {
            TrainingDBDataContext db = new TrainingDBDataContext();
            Organisasi org = db.Organisasis.Where(p => p.OrganisasiID == orgID).Single<Organisasi>();

            return org.OrganisasiNama;
        }
        catch { }

        return "-";
    }
}

```


6

Design Antarmuka

Pada bagian ini akan diterangkan hal-hal yang berhubungan dengan design antarmuka yang meliputi design antarmuka halaman web site dan control-control yang ada didalamnya. Adapun bahasan pada bagian ini meliputi :

1. Master Page.
2. Theme.
3. Skin.
4. User Control.

Master Page

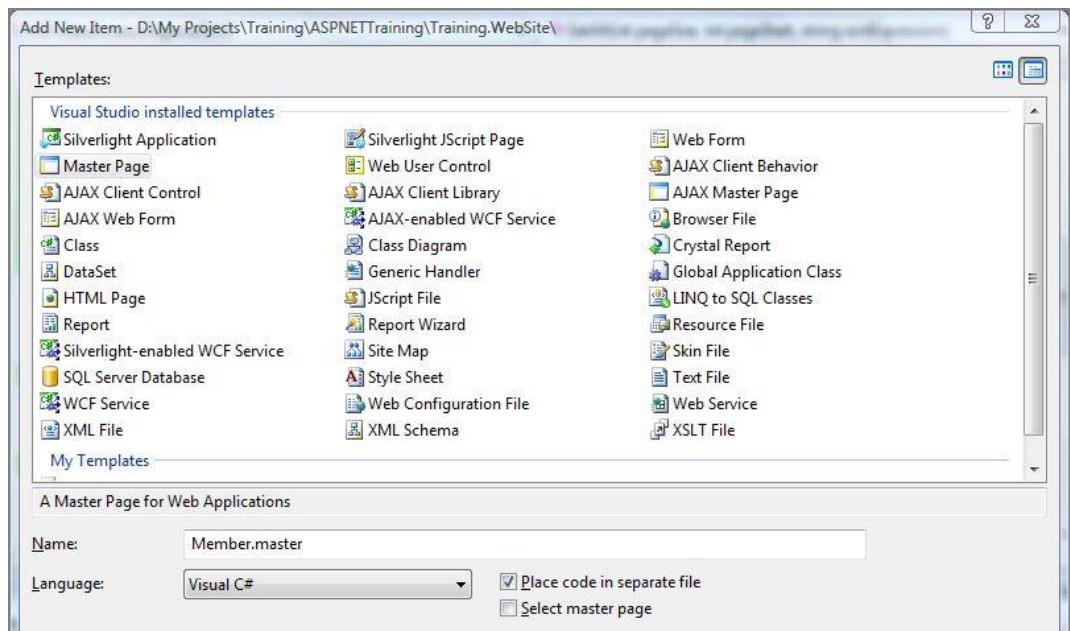
Master Page adalah halaman yang menjadi template antarmuka agar tampilan halaman-halaman pada Web Site dapat dibuat seragam. Selain itu dengan adanya Master Page, bila terjadi perubahan antarmuka website cukup mengubah halaman ini saja maka seluruh tampilan halaman-halaman pada Web Site akan ikut berubah.

File master ini akan digunakan oleh file subpage atau content page. Extension file Master Page adalah .master sedangkan halaman content akan menggunakan file dengan extension .aspx.

Membuat Master Page

Untuk menambahkan Master Page pada Web Site, klik kanan pada Web Site atau Web Application kemudian pilih Add New Item.

Pada window Add New Item pilih Master Page.



Berikan nama file pada kolom Name dan pilih Language yang akan digunakan.

Berikut isi file Master Page.

Member.master

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="MasterPage_Member" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceholder id="head" runat="server">
    </asp:ContentPlaceholder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ContentPlaceholder id="ContentPlaceholder1" runat="server">
                </asp:ContentPlaceholder>
            </div>
        </form>
    </body>
</html>
```

Code file :

Master.master.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;

public partial class MasterPage_Member : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
```

```
}
```

Pada file .master, dapat dilihat server control berikut :

```
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
            </asp:ContentPlaceHolder>
        </div>
    </form>
</body>
```

Control ContentPlaceHolder berfungsi sebagai area yang akan menampilkan isi dari file subpage atau content page (file .aspx).

Pada file .master, dapat ditambahkan tag HTML, server control seperti halnya file .aspx. berikut ini contoh antarmuka dari file .master.

Master.master

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="MasterPage_Member" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceHolder id="head" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table width="100%" cellpadding="0" cellspacing="0" border="1">
                <tr>
                    <td colspan="2">Header</td>
                </tr>
                <tr>
                    <td style="width:230px">Menu</td>
                    <td>
                        Content
                        <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
                        </asp:ContentPlaceHolder>
                    </td>
                </tr>
                <tr>
                    <td colspan="2">Footer</td>
                </tr>
            </table>
        </div>
    </form>
</body>
</html>
```

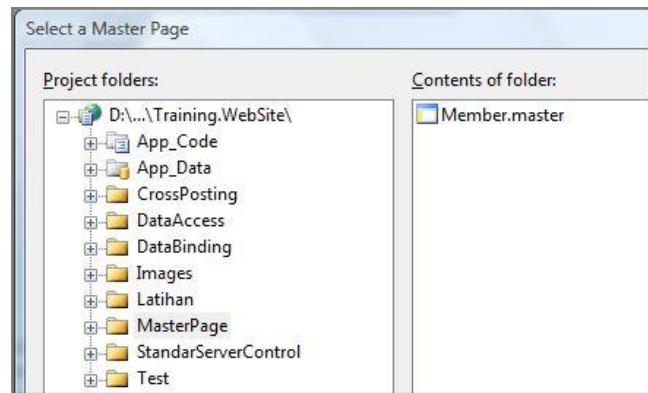
Menggunakan Master Page

File .master tidak bisa dilihat pada browser secara langsung, tetapi mesti digunakan terlebih dahulu oleh file .aspx. berikut ini cara mengaitkan file .aspx dengan file .master yang telah dibuat.

Pada saat menambahkan Web Form, centang checkbox Select master page.



Kemudian pilih file .master yang ingin digunakan.



Maka dapat dilihat kode file .aspx yang baru saja dibuat sebagai berikut :

```
<%@ Page Language="C#" MasterPageFile("~/MasterPage/Member.master"
AutoEventWireup="true" CodeFile="Default4.aspx.cs" Inherits="Test_Default4"
Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceholderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceholderID="ContentPlaceholder1" Runat="Server">
    Content ditulis disini
</asp:Content>
```

Dapat dilihat perbedaan file .aspx yang sebelumnya dibuat tanpa menggunakan Master Page.

Mengakses Server Control Pada Master Page

Pada bagian ini akan diterangkan bagaimana mengontrol server control yang ada pada file .master dari file .aspx.

Berikut ini adalah kode file .master :

```
Master.master
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="MasterPage_Member" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceholder id="head" runat="server">
        </asp:ContentPlaceholder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table width="100%" cellpadding="0" cellspacing="0" border="1">
                <tr>
                    <td colspan="2">
                        Header <br />
                        <asp:Label ID="Label_MasterPage" runat="server"
                            Text="Ditulis dari Master Page"></asp:Label>
                    </td>
                </tr>
            </table>
        </div>
    </form>
</body>
</html>
```

```

</tr>
<tr>
    <td style="width:230px">Menu</td>
    <td>
        Content
        <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
            </asp:ContentPlaceHolder>
        </td>
    </tr>
    <tr>
        <td colspan="2">Footer</td>
    </tr>
</table>
</div>
</form>
</body>
</html>

```

Pada kode di atas dapat dilihat control Label (yang dicetak tebal), label itu akan diakses oleh file .aspx dengan kode seperti berikut :

Default4.aspx.cs
<pre> using System; using System.Collections; using System.Configuration; using System.Data; using System.Linq; using System.Web; using System.Web.Security; using System.Web.UI; using System.Web.UI.HtmlControls; using System.Web.UI.WebControls; using System.Web.UI.WebControls.WebParts; using System.Xml.Linq; public partial class Test_Default4 : System.Web.UI.Page { protected void Page_Load(object sender, EventArgs e) { Label labelMasterPage = (Label)Master.FindControl("Label_MasterPage"); labelMasterPage.Text = "Ditulis dari file ASPX"; labelMasterPage.ForeColor = System.Drawing.Color.Blue; } } </pre>

Mengakses Server Control Pada File ASPX dari Master Page

Pada bagian ini akan diberikan contoh untuk mengakses server control yang ada pada file .aspx dari file .master.

Berikut ini adalah kode file .aspx

Default4.aspx
<pre> <%@ Page Language="C#" MasterPageFile="~/MasterPage/Member.master" AutoEventWireup="true" CodeFile="Default4.aspx.cs" Inherits="Test_Default4" Title="Untitled Page" %> <asp:Content ID="Content1" ContentPlaceholderID="head" Runat="Server"> </asp:Content> <asp:Content ID="Content2" ContentPlaceholderID="ContentPlaceHolder1" Runat="Server"> <asp:Label ID="Label_ContentPage" runat="server" Text="Label pada file ASPX"></asp:Label> </asp:Content> </pre>

Sedangkan berikut adalah kode dari code file file Master.master.

Master.master.cs
<pre> using System; using System.Collections; using System.Configuration; </pre>

```

using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;

public partial class MasterPage_Member : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        this.Page.Title = "Hello World";

        Label labelContentPage = (Label)
            ContentPlaceHolder1.FindControl("Label_ContentPage");

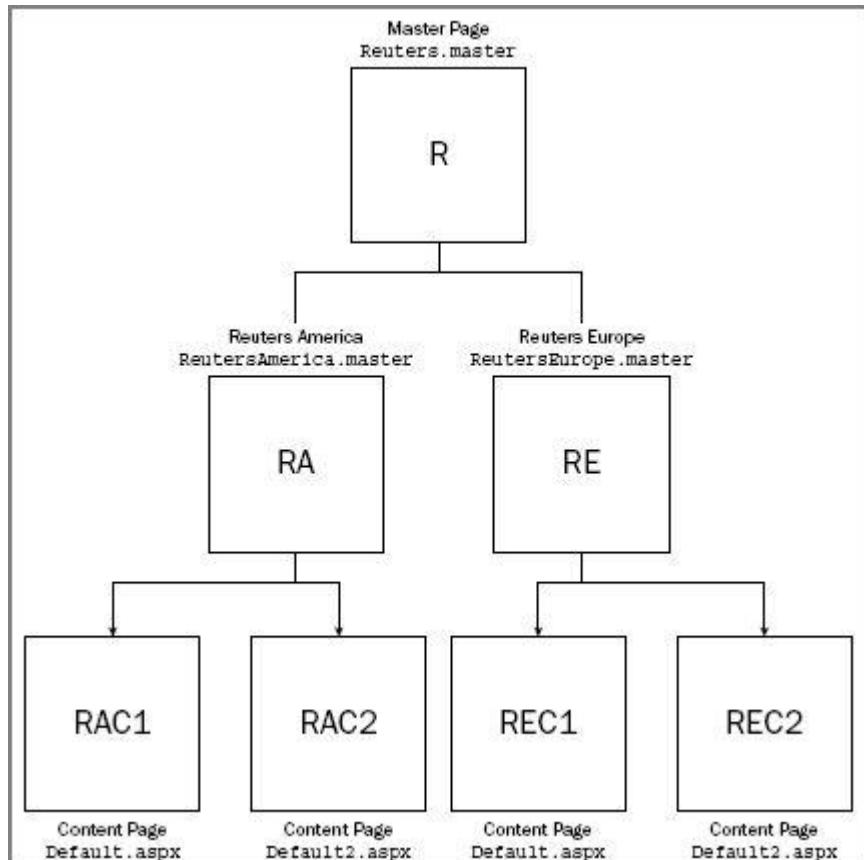
        labelContentPage.Text = "Di update dari Master Page";
    }
}

```

Nesting Master Page

Master Page dapat dimanfaatkan secara bertingkat, jadi dapat dibuat satu buah file R.master utama dimana dibawahnya terdapat file .master lain yaitu RA.master dan RE.master. setelah itu masing-masing file .master tersebut digunakan oleh file .aspx.

Deskripsinya dapat dilihat pada gambar di bawah ini.



Berikut ini adalah contoh kode file .master utama.

```

Utama.master
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Utama.master.cs"

```

```

Inherits="MasterPage_Utama" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceHolder id="head" runat="server">
        </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table width="100%" border="1" cellpadding="0" cellspacing="0">
                <tr>
                    <td>Header</td>
                </tr>
                <tr>
                    <td>
                        Content <br />
                        <asp:ContentPlaceHolder id="ContentPlaceHolder_Utama" runat="server">
                            </asp:ContentPlaceHolder>
                        </td>
                    </tr>
                    <tr>
                        <td>Footer</td>
                    </tr>
            </table>
        </div>
    </form>
</body>
</html>

```

Berikut ini kode sub master page. Pada halaman ini mesti dibuat control ContentPlaceHolder secara manual.

```

SubMasterPage1.master
<%@ Master Language="C#" MasterPageFile="~/MasterPage/Utama.master"
AutoEventWireup="false" CodeFile="SubMasterPage1.master.cs"
Inherits="MasterPage_SubMasterPage1" %>

<asp:Content ID="Content1" ContentPlaceholderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceholderID="ContentPlaceHolder_Utama"
Runat="Server">

    <table border="1" width="100%" cellpadding="0" cellspacing="0">
        <tr>
            <td style="width:300px">Menu</td>
            <td>
                Content<br />
                <asp:ContentPlaceHolder id="ContentPlaceHolder_Sub1" runat="server">
                    </asp:ContentPlaceHolder>
                </td>
        </tr>
    </table>
</asp:Content>

```

Sedangkan berikut adalah kode dari file .aspx.

```

Default6.aspx
<%@ Page Language="C#" MasterPageFile="~/MasterPage/SubMasterPage1.master"
AutoEventWireup="true" CodeFile="Default6.aspx.cs" Inherits="Test_Default6"
Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceholderID="ContentPlaceHolder_Sub1"
Runat="Server">
    Hello World
</asp:Content>

```

Theme dan Skin

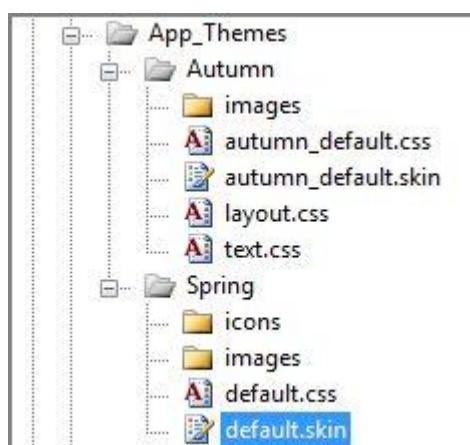
Theme mirip dengan Cascading Style Sheets (CSS) yang mempunyai fungsi untuk mengatur style antarmuka halaman web.

Theme pada ASP.NET, selain berpengaruh untuk mengatur style tag-tag HTML juga dapat digunakan untuk mengatur style antarmuka dari server control.

Pada Theme terdiri atas :

1. File CSS.
2. File Skin.
3. Gambar.

Berikut ini contoh Theme yang dapat diimplementasikan pada Web Site atau Web Application.



Pada gambar di atas dapat dilihat terdapat dua theme yaitu :

1. Autumn.
2. Spring.

Pada masing-masing theme terdapat file CSS, Skin dan gambar. File CSS yang ada pada theme tidak perlu didaftarkan secara manual pada halaman web.

Implementasi Theme

Berikut ini contoh implementasi Theme pada Web Site atau Web Application.

Berikut langkah-langkah yang mesti diikuti :

1. Klik kanan pada Web Site atau Web Application.
2. Pilih Add ASP.NET Folder > App_Themes.
3. Klik kanan pada App_Themes > Add ASP.NET Folder > Theme. Berikan nama theme yang baru dibuat.

- Pada folder theme yang baru dibuat (misal Autumn atau Spring) tambahkan file-file CSS, Skin atau folder gambar untuk menyimpan gambar yang akan digunakan pada halaman web.
- Untuk percobaan, tambahkan file autumn_default.css pada folder theme Autumn dengan kode seperti berikut.

```
autumn_default.css
```

```
body
{
    font-family:Verdana, Arial;
    font-size:10px;
    background-color:#000;
    color:#fff;
}
```

- Untuk menggunakan theme Autumn maka pada file.aspx gunakan perintah seperti berikut ini.

```
Default6.aspx
```

```
<%@ Page Language="C#"
    Theme="Autumn"
    MasterPageFile("~/MasterPage/SubMasterPage1.master"
    AutoEventWireup="true"
    CodeFile="Default6.aspx.cs"
    Inherits="Test_Default6"
    Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder_Sub1"
Runat="Server">
    Hello World
</asp:Content>
```

Pada gambar berikut dapat dilihat secara otomatis Visual Studio mengetahui theme yang ada pada Web Site atau Web Application.

```
<%@ Page Language="C#"
    Theme=
    MasterPageFile "~/MasterPage/SubMasterPage1.master"
    AutoEventWireup="true"
    CodeFile="Default6.aspx.cs"
    Inherits="Test_Default6"
    Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="ContentPlaceHolder_Sub1"
Runat="Server">
    Hello World
</asp:Content>
```

Dengan cara ini setiap file .aspx yang ada mesti ditambahkan kode seperti di atas.

- Agar pemilihan theme dapat berlaku untuk semua halaman web dan mudah diubah apabila ingin menggunakan theme yang lain maka dapat dilakukan dengan melakukan deklarasi pada file web.config.

Ubah baris berikut :

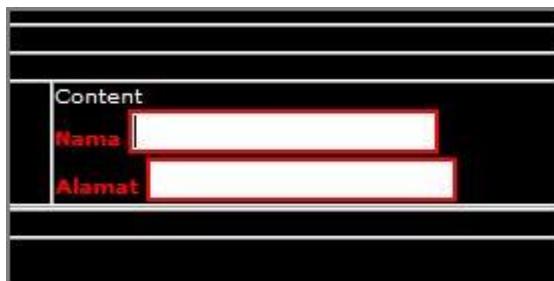
```
...
<pages>
...
</pages>
...
```

Menjadi seperti berikut :

```
...
<pages theme="Autumn">
...
</pages>
...
```

Implementasi Skin

Server control dapat dibuat antarmukanya sesuai keinginan. Misalnya dibuat form seperti berikut :



Dengan kode seperti berikut :

```
<asp:Label ID="Label_Form_Nama"  
    Text="Nama"  
    Font-Bold="true"  
    Font-Size="10px"  
    Font-Names="Verdana"  
    ForeColor="Red"  
    runat="server" >  
</asp:Label>  
<asp:TextBox ID="TextBox_Form_Nama"  
    BorderColor="Red"  
    BorderWidth="2px"  
    runat="server">  
</asp:TextBox>  
<br />  
<asp:Label ID="Label_Form_Alamat"  
    Text="Alamat"  
    Font-Bold="true"  
    Font-Size="10px"  
    Font-Names="Verdana"  
    ForeColor="Red"  
    runat="server" >  
</asp:Label>  
<asp:TextBox ID="TextBoxForm_Alamat"  
    BorderColor="Red"  
    BorderWidth="2px"  
    runat="server">  
</asp:TextBox>
```

Misalnya suatu saat style antarmuka dari control Label dan TextBox ingin diubah, maka yang mesti dilakukan adalah mengubah setiap control tersebut. Bayangkan misalnya control-control tersebut juga terdapat pada halaman lain, maka mesti dilakukan proses modifikasi style pada setiap control pada setiap halaman yang menggunakannya.

Untuk membantu memecahkan masalah ini, pada ASP.NET dikenal dengan istilah Skin.

Berikut ini langkah yang dapat diikuti untuk implementasi Skin :

1. Pada folder theme (misalnya folder theme Autumn) tambahkan file .skin, misalnya nama file adalah autumn_default.skin
2. Pada file autumn_default.skin tambahkan kode berikut :

```
autumn_default.skin  
<asp:Label Font-Bold="true"  
    Font-Size="10px"  
    Font-Names="Verdana"  
    ForeColor="Red"  
    runat="server" >  
</asp:Label>  
  
<asp:TextBox BorderColor="Red"
```

```

        BorderWidth="2px"
        runat="server">
</asp:TextBox>

```

Perhatikan kode di atas, tidak ada atribut ID. Dan pada file .aspx di atas cukup ditulis kode seperti berikut :

```

<asp:Label ID="Label_Form_Nama"
    Text="Nama"
    runat="server" >
</asp:Label>

<asp:TextBox ID="TextBox_Form_Nama"
    runat="server">
</asp:TextBox>

<br />

<asp:Label ID="Label_Form_Alamat"
    Text="Alamat"
    runat="server" >
</asp:Label>

<asp:TextBox ID="TextBoxForm_Alamat"
    runat="server">
</asp:TextBox>

```

Dapat dilihat tidak adalagi atribut untuk mengubah antarmuka pada control-control di atas, apabila ingin mengubah style dari control di atas cukup lakukan perubahan pada file .skin.

3. Dekrasi style control yang ada pada file .skin dapat diberi atribut SkinID apabila style tersebut hanya akan bisa digunakan bila dipanggil oleh control yang ada pada halaman web. Sebagai contoh bisa dilihat pada kode berikut :

```

autumn_default.skin
<asp:TextBox SkinID="TextBoxBlue"
    BorderColor="Blue"
    BorderWidth="1px"
    runat="server">
</asp:TextBox>

```

Dapat dilihat pada kode di atas terdapat atribut SkinID dengan nilai TextBoxBlue. Maka untuk menggunakanya dapat dilihat kode berikut di bawah ini.

```

<asp:TextBox ID="TextBoxForm_Alamat"
    SkinID="TextBoxBlue"
    runat="server">
</asp:TextBox>

```

Pada kode di atas dapat dilihat terdapat atribut SkinID untuk memanggil style yang telah dideklarasikan pada file .skin.

Web User Control

Web user control adalah suatu halaman kecil yang dapat ditempelkan atau digunakan pada halaman web, baik itu pada file .master ataupun .aspx.

Sebagai contoh kasus adalah, terdapat form pencarian dan link seperti pada gambar berikut ini:



Apabila form seperti di atas ingin dapat digunakan pada semua halaman dengan kemudahan pengelolaan apabila terjadi perubahan maka cukup dibuat sebuah Web User Control.

Web User Control mempunyai file dengan extension .ascx dan mempunyai sifat reuseable.

Membuat dan Menggunakan Web User Control

Berikut ini contoh membuat dan menggunakan Web User Control.

1. Klik kanan pada Web Site atau Web Application dan pilih Add New Item.
2. Pada window Add New Item pilih Web User Control, beri nama file sebagai berikut WelcomeControl.ascx.
3. Tulis kode berikut pada file WelcomeControl.ascx.

```
WelcomeControl.ascx
<%@ Control Language="C#" AutoEventWireup="true"
CodeFile="WelcomeControl.ascx.cs" Inherits="Control_WelcomeControl" %>
<table>
  <tr>
    <td>Search</td>
    <td>
      <asp:TextBox ID="TextBox_Keyword" runat="server"></asp:TextBox>
    </td>
    <td>
      <asp:Button ID="Button_Search" runat="server" Text="Search" />
    </td>
  </tr>
  <tr>
    <td colspan="3">
      <asp:HyperLink ID="HyperLink_SignIn" runat="server">
        Sign In
      </asp:HyperLink> |
      <asp:HyperLink ID="HyperLink_Join" runat="server">
        Join
      </asp:HyperLink> |
      <asp:HyperLink ID="HyperLink_Help" runat="server">
        Help
      </asp:HyperLink>
    </td>
  </tr>
</table>
```

4. Berikut ini contoh kode pada file .aspx untuk mendaftarkan dan menampilkan web user control di atas.

```
WebUserControlExample.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="WebUserControlExample.aspx.cs"
Inherits="Control_WebUserControlExample" %>
<%@ Register Src="~/Control/WelcomeControl.ascx" TagName="Welcome"
TagPrefix="MSPControl" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
```

```

<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <MSPControl:Welcome ID="WelcomeMessage" runat="server" />
        </div>
    </form>
</body>
</html>

```

Berikut ini contoh lain Web User Control yang dapat dibuat.

Berikut ini adalah kode dari Web User Control :

TwoOptionControl.ascx

```

<%@ Control Language="C#" AutoEventWireup="true" CodeFile="TwoOptionControl.aspx.cs"
Inherits="Control_TwoOptionControl" %>
<asp:RadioButtonList ID="RadioButtonListTwoOption" RepeatDirection="Horizontal"
Width="200px" runat="server">
    <asp:ListItem Text="Option1" Value="1"></asp:ListItem>
    <asp:ListItem Text="Option2" Value="2"></asp:ListItem>
</asp:RadioButtonList>

```

Code file :

TwoOptionControl.aspx.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Control_TwoOptionControl : System.Web.UI.UserControl
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    public string Option1
    {
        set
        {
            if (!String.IsNullOrEmpty(value))
            {
                RadioButtonListTwoOption.Items.FindByValue("1").Text = value;
            }

            ViewState["Option1"] = value;
        }
    }

    public string Option2
    {
        set
        {
            if (!String.IsNullOrEmpty(value))
            {
                RadioButtonListTwoOption.Items.FindByValue("2").Text = value;
            }

            ViewState["Option2"] = value;
        }
    }

    public string Value

```

```

    {
        get
        {
            if (!String.IsNullOrEmpty(RadioButtonListTwoOption.SelectedValue))
            {
                if (RadioButtonListTwoOption.SelectedValue.Equals("1"))
                {
                    ViewState["Value"] = "1";
                }
                else if (RadioButtonListTwoOption.SelectedValue.Equals("2"))
                {
                    ViewState["Value"] = "2";
                }
                else
                {
                    ViewState["Value"] = "9";
                }
            }
            else
            {
                ViewState["Value"] = "9";
            }

            return Convert.ToString(ViewState["Value"]);
        }
        set
        {
            if (!String.IsNullOrEmpty(value))
            {
                if (value.Equals("1"))
                {
                    RadioButtonListTwoOption.SelectedValue = "1";
                }
                else if (value.Equals("0"))
                {
                    RadioButtonListTwoOption.SelectedValue = "0";
                }
                else
                {
                    RadioButtonListTwoOption.SelectedIndex = -1;
                }
            }
            else
            {
                RadioButtonListTwoOption.SelectedIndex = -1;
            }

            ViewState["Value"] = value;
        }
    }
}

```

Dan berikut adalah contoh penggunaannya pada file .aspx.

WebUserControlExample.aspx

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="WebUserControlExample.aspx.cs" Inherits="Control_WebUserControlExample" %>
<%@ Register Src("~/Control/TwoOptionControl.ascx" TagName="TwoOptionControl"
TagPrefix="MSPControl" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <MSPControl:TwoOptionControl ID="TwoOptionControl1" Option1="Yes"
Option2="No" runat="server" />
            <br />
            <asp:Button ID="Button1" runat="server" Text="Button" onclick="Button1_Click" />
            <br />
            <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>

```

```
</div>
</form>
</body>
</html>
```

Code file :

WebUserControlExample.aspx.cs

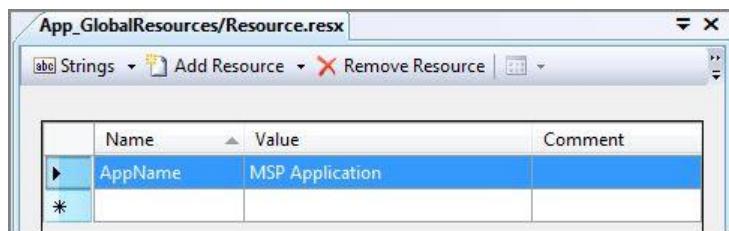
```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Control_WebUserControlExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = TwoOptionControl1.Value;
    }
}
```

Resource & Globalization

ASP.NET mempunyai feature resource yang memungkinkan text dapat disimpan dalam suatu file .resx yang merupakan tabel berisi Name, Value dan Comment seperti pada gambar berikut.



	Name	Value	Comment
▶	AppName	MSP Application	
*			

Data yang disimpan pada file ini dapat digunakan pada halaman web (file .master, .aspx dan .ascx).

Dengan feature ini memungkinkan halaman web dibuat dalam berbagai bahasa dan dapat diubah dengan mudah akan menggunakan bahasa yang diinginkan.

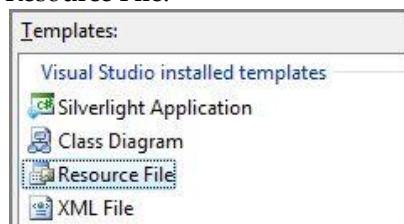
File Resource mempunyai dua tipe, yaitu :

1. Global, yang artinya dapat digunakan pada semua halaman web.
2. Local, yang artinya hanya dapat digunakan pada sebuah halaman web saja.

Global Resource

Berikut ini langkah-langkah membuat global resource dan cara penggunaanya.

1. Klik kanan pada Web Site atau Web Application, pilih Add ASP.NET Folder > App_GlobalResources.
2. Klik kanan pada folder App_GlobalResources > Add New Item. Pada template pilih Resource File.



Berikan nama MSP.resx.

3. Berikan data seperti pada gambar berikut.

Name	Value	Comment
AppName	MSP Web Site	
Label_Name	Name	
Label_Address	Address	
*		

4. Buat file .aspx dengan isi seperti berikut.

```

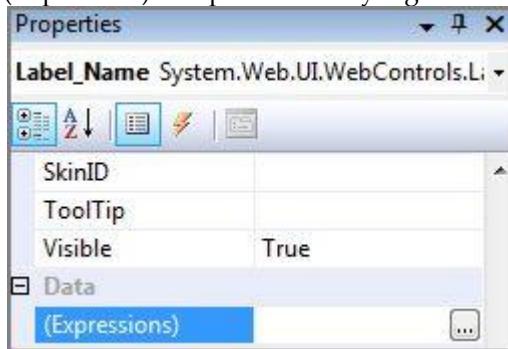
GlobalResourceExample.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="GlobalResourceExample.aspx.cs"
Inherits="Resource_GlobalResourceExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

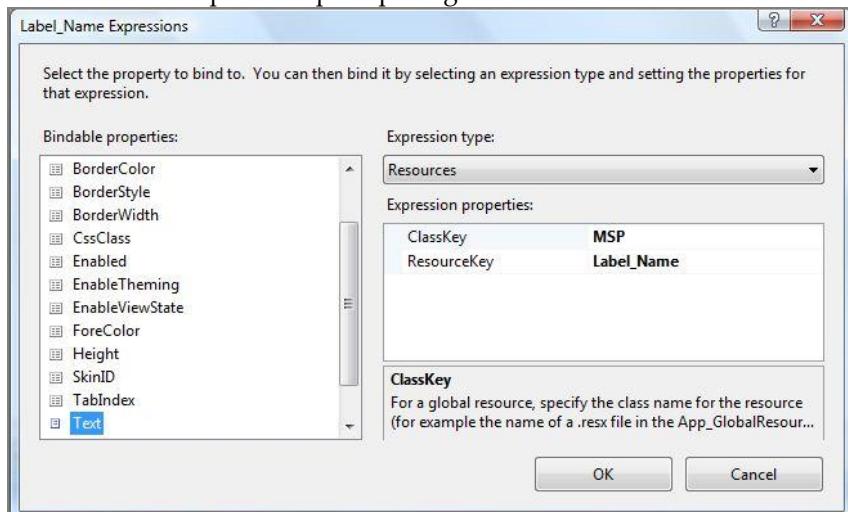
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table>
                <tr>
                    <td>
                        <asp:Label ID="Label_Name" runat="server" Text="Label">
                            </asp:Label>
                    </td>
                    <td>:</td>
                    <td>
                        <asp:TextBox ID="TextBox_Name" runat="server">
                            </asp:TextBox>
                    </td>
                </tr>
                <tr>
                    <td>
                        <asp:Label ID="Label_Address" runat="server" Text="Label">
                            </asp:Label>
                    </td>
                    <td>:</td>
                    <td>
                        <asp:TextBox ID="TextBox_Address" runat="server">
                            </asp:TextBox>
                    </td>
                </tr>
            </table>
        </div>
    </form>
</body>
</html>

```

5. Ubah ke mode Design, klik pada Label_Name, pada kolom Properties klik (Expression) dan pilih tombol yang akan muncul seperti pada gambar berikut.



Maka akan ditampilkan seperti pada gambar berikut.



Isikan nilai seperti pada gambar di atas dan klik OK. Nilai ClassKey diambil dari nama file yang ada pada App_GlobalResource yang telah dibuat. Sedangkan nilai ResourceKey adalah nilai field Name yang nilainya akan digunakan.

Hasil dari proses ini dapat dilihat terdapat perubahan pada control Label_Name seperti pada kode berikut ini :

```
<asp:Label ID="Label_Name" runat="server"
    Text="<%$ Resources:MSP, Label_Name %>">
</asp:Label>
```

6. Lakukan hal yang sama pada control Label_Address dengan nilai seperti berikut.

```
<asp:Label ID="Label_Address" runat="server"
    Text="<%$ Resources:MSP, Label_Address %>">
</asp:Label>
```

Hasilnya dapat dilihat pada gambar berikut.

Berikut ini adalah membuat file .resx dengan nilai-nilai dalam bahasa Indonesia.

1. Klik kanan pada folder App_GlobalResource pilih Add New Item, pilih Resource File pada kolom template dan berikan file tersebut dengan nama MSP.id.resx. id pada nama file MSP.id.resx menandakan kode bahasa dari suatu negara atau lebih tepatnya adalah nilai dari uiCulture pada globalization.
2. Isi file ini sama seperti pada file MSP.resx tetapi dengan Value dalam bahasa Indonesia. Seperti pada gambar berikut ini.

Name	Value	Comment
AppName	Web Site MSP	
Label_Address	Alamat	
Label_Name	Nama	

3. Secara default resource yang digunakan pada halaman web adalah file MSP.resx, apabila ingin menggunakan file MSP.id.resx sebagai default maka perlu ditambahkan baris berikut pada file web.config.

```
Web.config
...
<system.web>
    <globalization uiCulture="id" culture="id-ID" />
...

```

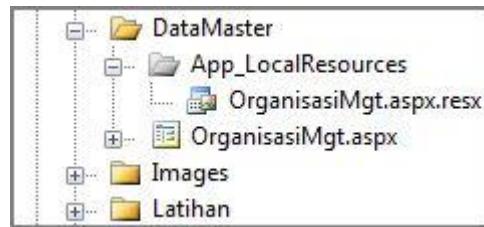
4. Refresh halaman GlobalResourceExample.aspx maka dapat dilihat hasil seperti berikut.

Local Resource

Berbeda dengan Global Resource yang berlaku untuk seluruh halaman web maka Local resource hanya berlaku untuk satu halaman saja.

Berikut langkah-langkah yang mesti diikuti untuk melakukan implementasi Local Resource :

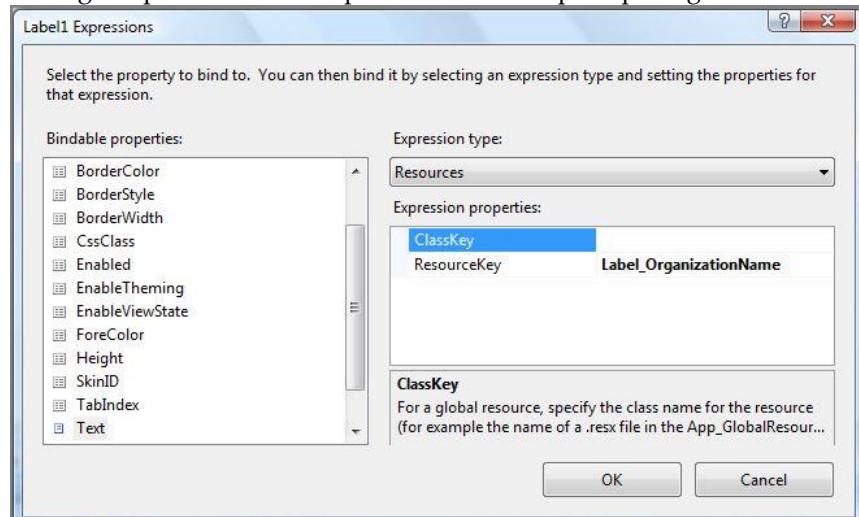
1. Misalnya terdapat modul DataMaster yang disimpan dalam direktori DataMaster. Di dalamnya terdapat file OrganisasiMgt.aspx.
2. Pada direktori DataMaster tambahkan folder App_LocalResources dengan cara klik kanan pada folder DataMaster > Add ASP.NET Folder > App_LocalResources.
3. Tambahkan file .resx pada folder tersebut dengan nama OrganisasiMgt.aspx.resx. Hasilnya bisa dilihat pada gambar berikut.



4. Berikut isi file dari Organisasi.aspx.resx.

	Name	Value	Comment
▶	Label_Organization	Organization Name	
*			

5. Sedangkan pada window Expression dibuat seperti pada gambar berikut ini.



Berbeda dengan Global Resource, pada Local Resource nilai ClassKey tidak perlu diisi. Maka dapat dilihat kode control akan seperti berikut ini.

```
<asp:Label ID="Label1" runat="server"
    Text="<%$ Resources:Label_OrganizationName %>">
</asp:Label>
```

8

Validasi

Validasi diperlukan untuk memeriksa nilai-nilai pada form sebelum disimpan ke database. Misalnya form berikut ini.



Misalnya ketika tombol Simpan diklik maka nilai-nilai Nama, Umur dan Email akan diperiksa dengan prosedur seperti berikut :

1. Setiap control TextBox harus diisi.
2. Umur harus diisi dengan angka.
3. Email harus valid.

Dengan sarat seperti di atas maka saat tombol Simpan diklik sebelum dilakukan proses penyimpanan data perlu ada proses pemeriksaan nilai-nilai TextBox.

Berikut contoh kode untuk pemeriksaan nilai-nilai pada TextBox.

```
ValidationExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ValidasiExample.aspx.cs"
Inherits="Validasi_ValidateExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

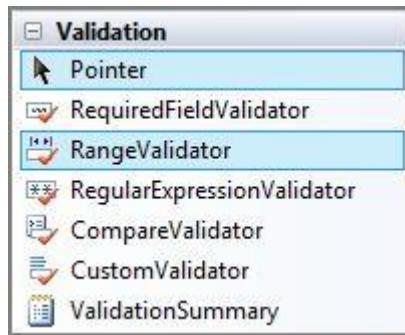
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table>
                <tr>
                    <td colspan="3">
                        <asp:ValidationSummary ID="ValidationSummary1"
                            ValidationGroup="DefaultForm" runat="server" />
                    </td>
                </tr>
                <tr>
                    <td>
                        <asp:Label ID="Label1" runat="server" Text="Nama"></asp:Label>
                    </td>
                    <td></td>
                    <td>
                        <asp:TextBox ID="TextBox1" ValidationGroup="DefaultForm"
                            runat="server">
                        </asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
                            ControlToValidate="TextBox1"
                            ValidationGroup="DefaultForm"
                            runat="server" />
                    </td>
                </tr>
            </table>
        </div>
    </form>
</body>
</html>
```

```

        ToolTip="Name is required."
        ErrorMessage="Name is required.">
        *
    </asp:RequiredFieldValidator>
</td>
</tr>
<tr>
    <td>
        <asp:Label ID="Label2" runat="server" Text="Age"></asp:Label>
    </td>
    <td></td>
    <td>
        <asp:TextBox ID="TextBox2" ValidationGroup="DefaultForm"
            runat="server">
        </asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
            ControlToValidate="TextBox2"
            ValidationGroup="DefaultForm"
            runat="server"
            ToolTip="Age is required."
            ErrorMessage="Age is required.">
            *
        </asp:RequiredFieldValidator>
        <asp:RangeValidator ID="RangeValidator1"
            runat="server"
            ControlToValidate="TextBox2"
            ErrorMessage="Age must number."
            ToolTip="Age must number."
            MaximumValue="200"
            ValidationGroup="DefaultForm"
            MinimumValue="1">*</asp:RangeValidator>
    </td>
</tr>
<tr>
    <td>
        <asp:Label ID="Label3" runat="server" Text="Email"></asp:Label>
    </td>
    <td></td>
    <td>
        <asp:TextBox ID="TextBox3" ValidationGroup="DefaultForm"
            runat="server">
        </asp:TextBox>
        <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
            ControlToValidate="TextBox3"
            ValidationGroup="DefaultForm"
            runat="server"
            ToolTip="Email is required."
            ErrorMessage="Email is required.">
            *
        </asp:RequiredFieldValidator>
        <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
            runat="server"
            ToolTip="Email is not valid."
            ValidationGroup="DefaultForm"
            ErrorMessage="Email is not valid."
ControlToValidate="TextBox3"
ValidationExpression="\w+([-_.']\w+)*@(\w+([-_.']\w+)*\.\w+([-_.']\w+)*">*</asp:RegularExpressionValidator>
    </td>
</tr>
<tr>
    <td colspan="3">
        <asp:Button ID="Button1" runat="server" Text="Simpan"
            ValidationGroup="DefaultForm" />
    </td>
</tr>
</table>
</div>
</form>
</body>
</html>

```

Untuk melakukan validasi pada control-control yang ada pada form dapat digunakan control-control yang terdapat pada kelompok Validation pada Toolbox.



RequiredFieldValidator

Control ini membuat control yang akan divalidasi harus mempunyai isi.

Berikut contohnya.

```
<asp:ValidationSummary ID="ValidationSummary1" ValidationGroup="DefaultForm"
runat="server" />

<asp:TextBox ID="TextBox1" ValidationGroup="DefaultForm" runat="server">
</asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator1"
ControlToValidate="TextBox1"
ValidationGroup="DefaultForm"
runat="server"
ToolTip="Name is required."
ErrorMessage="Name is required.">
*
</asp:RequiredFieldValidator>
```

Keterangan :

1. Control TextBox1 adalah control yang akan divalidasi.
2. Control RequiredFieldValidator1 adalah control validator.
3. Atribut ValidationGroup berfungsi untuk mengelompokkan control yang akan divalidasi dengan control yang berfungsi sebagai validator.
4. Atribut ControlToValidate pada control validator berfungsi untuk menentukan control yang akan divalidasi.
5. ErrorMessage adalah error yang akan ditampilkan pada control ValidationSummary1.

RangeValidator

Control RangeValidator berfungsi untuk memeriksa nilai pada input apakah memenuhi sarat berada pada range nilai yang sudah ditentukan. Nilai dapat berupa angka atau tanggal.

Berikut ini contoh penggunaan control RangeValidator.

```
<asp:ValidationSummary ID="ValidationSummary1" ValidationGroup="DefaultForm"
runat="server" />

<asp:TextBox ID="TextBox2" ValidationGroup="DefaultForm" runat="server">
</asp:TextBox>
<asp:RangeValidator ID="RangeValidator1"
runat="server"
ControlToValidate="TextBox2"
ErrorMessage=" Age should be between 1 - 200."
ToolTip="Age should be between 1 - 200.">
```

```
    MaximumValue="200"
    ValidationGroup="DefaultForm"
    MinimumValue="1">>*</asp:RangeValidator>
```

Keterangan :

1. MaximumValue : nilai maksimum.
2. MinimumValue : nilai minimum.

RegularExpressionValidator

Control RegularExpressionValidator berfungsi melakukan validasi dengan sarat mencocokkan nilai pada regular expression yang ditentukan.

Berikut ini contoh penggunaan control RegularExpressionValidator.

```
<asp:ValidationSummary ID="ValidationSummary1" ValidationGroup="DefaultForm"
runat="server" />

<asp:TextBox ID="TextBox3" ValidationGroup="DefaultForm" runat="server">
</asp:TextBox>
<asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server"
ToolTip="Email is not valid."
ValidationGroup="DefaultForm"
ErrorMessage="Email is not valid." ControlToValidate="TextBox3"

ValidationExpression="\w+([-.\'])\w+*@(\w+([-.\'])\w+)*\.\w+([-.
])\w+)*"></asp:RegularExpressionValidator>
```

Keterangan :

1. ValidationExpression : nilai regular expression.

CompareValidator

Control CompareValidator berfungsi untuk mencocokkan nilai antara dua control, misalnya terdapat dua control TextBox yang berfungsi menyimpan password. Maka kedua control TextBox tersebut harus mempunyai isi yang sama. Maka dapat digunakan control ini untuk memeriksanya.

```
<asp:ValidationSummary ID="ValidationSummary1" ValidationGroup="DefaultForm"
runat="server" />
<br />
Password 1 : <asp:TextBox ID="TextBox1" ValidationGroup="DefaultForm"
TextMode="Password" runat="server"></asp:TextBox>
<br />
Password 2 : <asp:TextBox ID="TextBox2" ValidationGroup="DefaultForm"
TextMode="Password" runat="server"></asp:TextBox>
<br />
<asp:CompareValidator ID="CompareValidator1"
runat="server"
ErrorMessage="Password tidak sama"
ControlToCompare="TextBox1"
ControlToValidate="TextBox2"
Type="String"
Operator="Equal"
ValidationGroup="DefaultForm">
*
</asp:CompareValidator>
<br />
<asp:Button ID="Button1"
runat="server"
ValidationGroup="DefaultForm"
Text="Button" />
```

Control ini dapat digunakan untuk memeriksa nilai berdasarkan operator-operator seperti berikut :

1. Equal.
2. NotEqual.
3. GreaterThan.
4. GreaterThanEqual.
5. LessThan.
6. LessThanEqual.
7. DataTypeCheck.

Berikut contoh penggunaan CompareValidator agar hanya menerima nilai berupa angka saja.

```
<asp:ValidationSummary ID="ValidationSummary1" ValidationGroup="DefaultForm"
runat="server" />
<br />

Angka :
<asp:TextBox ID="TextBox3"
    ValidationGroup="DefaultForm"
    runat="server"></asp:TextBox>

<asp:CompareValidator ID="CompareValidator2"
    runat="server"
    ControlToValidate="TextBox3"
    Operator="DataTypeCheck"
    Type="Integer"
    ErrorMessage="Isi harus Integer."
    ToolTip="Isi harus Integer."
    ValidationGroup="DefaultForm">
    *
</asp:CompareValidator>

<br />
<asp:Button ID="Button1"
    runat="server"
    ValidationGroup="DefaultForm"
    Text="Button" />
```

Penggunaan control validator dapat digunakan lebih dari satu. Sebagai contoh, pada kode berikut ini terdapat kolom umur yang harus diisi, dan batasan umurnya adalah lebih dari 18 tahun.

```
Umur :
<asp:TextBox ID="TextBox3"
    ValidationGroup="DefaultForm"
    runat="server"></asp:TextBox>
<asp:RequiredFieldValidator ID="RequiredFieldValidator1"
    runat="server"
    ControlToValidate="TextBox3"
    ValidationGroup="DefaultForm"
    ToolTip="Umur harus diisi."
    ErrorMessage="Umur harus diisi.">
    *
</asp:RequiredFieldValidator>

<asp:CompareValidator ID="CompareValidator2"
    runat="server"
    ControlToValidate="TextBox3"
    Operator="DataTypeCheck"
    Type="Integer"
    ErrorMessage="Isi harus Integer."
    ToolTip="Isi harus Integer."
    ValidationGroup="DefaultForm">
    *
</asp:CompareValidator>

<asp:CompareValidator ID="CompareValidator3"
    runat="server"
    ControlToValidate="TextBox3"
```

```

        Operator="GreaterThan"
        ValueToCompare="18"
        ErrorMessage="Umur harus lebih 18 tahun."
        ToolTip="Umur harus lebih 18 tahun."
        ValidationGroup="DefaultForm">
    *
</asp:CompareValidator>

<br />

<asp:Button ID="Button1"
    runat="server"
    ValidationGroup="DefaultForm"
    Text="Button" />

```

CustomValidator

Control CustomValidator memungkinkan untuk membuat validator dengan sarat validasi yang lebih bebas sesuai keinginan.

Validasi Client-Side

Berikut ini contoh penggunaan fungsi pada client-side dengan menggunakan JavaScript untuk fungsi validasi.

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="CustomValidatorExample.aspx.cs" Inherits="Validasi_CustomValidatorExample"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <script language="javascript" type="text/javascript">
        function validateNumber(oSrc, args)
        {
            args.IsValid = (args.Value % 5 == 0);
        }
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ValidationSummary ID="ValidationSummary1" ValidationGroup="DefaultForm"
                runat="server" />
            <br />
            <asp:TextBox ID="TextBox1" ValidationGroup="DefaultForm" runat="server">
            </asp:TextBox>
            <asp:CustomValidator ID="CustomValidator1"
                runat="server"
                ControlToValidate="TextBox1"
                ToolTip="Number must be divisible by 5."
                ErrorMessage="Number must be divisible by 5."
                ClientValidationFunction="validateNumber"
                ValidationGroup="DefaultForm"
                Display="Dynamic"
                >
            *
        </asp:CustomValidator>
        <br />

        <asp:Button ID="Button1"
            runat="server"
            ValidationGroup="DefaultForm"
            Text="Button" />
    </div>
    </form>
</body>

```

```
</html>
```

Keterangan :

1. ClientValidationFunction : atribut untuk memanggil fungsi JavaScript yang akan menjadi fungsi untuk proses validasi.

Validasi Server-Side

Pada proses validasi server-side, method yang akan digunakan pada proses validasi disimpan pada code file dari file .aspx.

Berikut contoh penggunaan control CustomValidator untuk kasus validasi server-side.

CustomValidatorServerExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="CustomValidatorServerExample.aspx.cs"
Inherits="Validasi_CustomValidatorServerExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ValidationSummary ID="ValidationSummary1" ValidationGroup="DefaultForm"
runat="server" />
            <br />
            NIP :
            <asp:TextBox ID="TextBox1"
                ValidationGroup="DefaultForm"
                runat="server"></asp:TextBox>
            <asp:CustomValidator ID="CustomValidator1"
                runat="server"
                ControlToValidate="TextBox1"
                ValidationGroup="DefaultForm"
                OnServerValidate="ValidasiNIP"
                ErrorMessage="NIP sudah ada."*></asp:CustomValidator>
            <br />
            <asp:Button ID="Button1"
                runat="server"
                ValidationGroup="DefaultForm"
                Text="Simpan" onclick="Button1_Click" />
        </div>
    </form>
</body>
</html>
```

Code file :

CustomValidatorServerExample.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Validasi_CustomValidatorServerExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
```

```
{  
}  
  
protected void ValidasiNIP(object sender, ServerValidateEventArgs args)  
{  
    args.IsValid = false;  
  
    try  
    {  
        TrainingDBDataContext db = new TrainingDBDataContext();  
        var query = db.Pegawais.Where(p => p.NIP == args.Value);  
        int jumlahData = query.ToList().Count;  
  
        args.IsValid = !(jumlahData > 0);  
    }  
    catch (Exception ex)  
    {  
        args.IsValid = false;  
    }  
}  
  
protected void Button1_Click(object sender, EventArgs e)  
{  
    if (!isValid)  
    {  
        return;  
    }  
}
```

ASP.NET Ajax dan Ajax Control Toolkit

ASP.NET Ajax

Untuk implementasi ASP.NET Ajax pada Web Site atau Web Application dapat digunakan control group AJAX Extensions yang ada pada Toolbox.



Implementasi ASP.NET Ajax

Untuk memulai menggunakan ASP.NET Ajax, tambahkan control ScriptManager pada halaman web (file .aspx). Control ScriptManager dapat juga ditambahkan pada file .master.

```
<asp:ScriptManager ID="ScriptManager1" EnablePartialRendering="true" runat="server">
</asp:ScriptManager>
```

Keterangan :

1. EnablePartialRendering : secara default control ScriptManager menggunakan nilai true untuk atribut EnablePartialRendering. Dengan nilai true maka feature ASP.NET Ajax digunakan pada halaman tersebut. Bila tidak ingin menggunakan ASP.NET Ajax maka cukup berikan nilai false pada atribut ini.

Setelah control ScriptManager ditambahkan pada halaman web, maka perlu dibuat area yang akan di render secara partial. Area tersebut dapat dibuat dengan menggunakan control UpdatePanel.

Berikut contoh penggunaan control UpdatePanel.

```
<asp:ScriptManager ID="ScriptManager1"
    EnablePartialRendering="true"
    AsyncPostBackTimeout="600"
    runat="server">
</asp:ScriptManager>

<asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        <asp:Label ID="Label1" runat="server" Text="Label">
        </asp:Label>

        <asp:Button ID="Button1"
            runat="server"
            Text="Refresh"
            onclick="Button1_Click" />
    </ContentTemplate>
</asp:UpdatePanel>
```

```

</ContentTemplate>
</asp:UpdatePanel>

Code file :
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Ajax_AjaxExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
    }
}

```

Dari dilihat pada contoh di atas, area partial rendering dapat dibuat dengan control UpdatePanel dengan sintaks berikut :

```

<asp:UpdatePanel ID="UpdatePanel1" runat="server">
    <ContentTemplate>
        // control-control
    </ContentTemplate>
</asp:UpdatePanel>

```

Dengan meletakkan control Label1 dan Button1 di dalam control UpdatePanel maka ketika tombol Button1 diklik dapat dilihat nilai pada Label1 akan berubah tetapi tidak terjadi full postback, yang terjadi hanya area di dalam UpdatePanel saja yang berubah.

Pengenal UpdateMode pada UpdatePanel

Pada control UpdatePanel terdapat atribut UpdateMode yang dapat mempunyai nilai :

1. Always.
2. Conditional.

Pada contoh kode berikut, pada file .aspx dibuat dua buah area yang menggunakan control UpdatePanel.

```

UpdateModeExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="UpdateModeExample.aspx.cs"
Inherits="Ajax_UpdateModeExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
        </asp:ScriptManager>

```

```

<table width="500" border="1" cellpadding="1" cellspacing="1">
    <tr>
        <td style="width:250px">
            UpdatePanel 1
            <br />
            <asp:UpdatePanel ID="UpdatePanel1" UpdateMode="Always"
                runat="server">
                <ContentTemplate>
                    <asp:Label ID="Label1" runat="server" Text="Label">
                    </asp:Label>
                </ContentTemplate>
            </asp:UpdatePanel>
        </td>
        <td style="width:250px">
            UpdatePanel 2
            <br />
            <asp:UpdatePanel ID="UpdatePanel2" UpdateMode="Always"
                runat="server">
                <ContentTemplate>
                    <asp:Label ID="Label2" runat="server" Text="Label">
                    </asp:Label>
                    <br />
                    <asp:Button ID="Button1"
                        runat="server"
                        Text="Refresh"
                        onclick="Button1_Click" />
                </ContentTemplate>
            </asp:UpdatePanel>
        </td>
    </tr>
</table>
</div>
</form>
</body>
</html>

```

Code file :

```

UpdateModeExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Ajax_UpdateModeExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
        Label2.Text = DateTime.Now.ToString();
    }
}

```

Hasilnya dapat dilihat pada gambar berikut ini.



Dapat dilihat kode pada file .aspx, control UpdatePanel1 dan UpdatePanel2 mempunyai nilai UpdateMode=Always. Dengan nilai tersebut, bila tombol Refresh diklik maka nilai label yang ada pada UpdatePanel1 dan UpdatePanel2 akan sama-sama berubah.

Misalnya nilai UpdateMode pada UpdatePanel1 diubah menjadi Conditional seperti berikut.

```
<asp:UpdatePanel ID="UpdatePanel1" UpdateMode="Conditional" runat="server">
    <ContentTemplate>
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    </ContentTemplate>
</asp:UpdatePanel>
```

Maka bila tombol Refresh ditekan nilai label pada UpdatePanel1 tidak berubah secara otomatis. Untuk dapat mengubah nilai label pada UpdatePanel1 maka perlu dipanggil baris seperti berikut pada code file.

```
protected void Button1_Click(object sender, EventArgs e)
{
    Label1.Text = DateTime.Now.ToString();
    Label2.Text = DateTime.Now.ToString();

    UpdatePanel1.Update();
}
```

Trigger pada UpdatePanel

Pada contoh-contoh di atas control-control seperti Label dan Button berada pada area UpdatePanel. Pada contoh berikut ini misalnya control Button berada diluar area UpdatePanel, maka dapat digunakan Trigger untuk mendaftarkan Button tersebut agar ketika control Button diklik proses partial rendering tetap terjadi.

```
TriggerExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="TriggerExample.aspx.cs"
Inherits="Ajax_TriggerExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>
            <asp:UpdatePanel ID="UpdatePanel1" runat="server">
                <ContentTemplate>
                    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
                </ContentTemplate>
                <Triggers>
                    <asp:AsyncPostBackTrigger ControlID="Button1" />
                </Triggers>
            </asp:UpdatePanel>
            <asp:Button ID="Button1" runat="server" Text="Button" onclick="Button1_Click" />
        </div>
    </form>
</body>
```

```
</html>
```

Code file :

```
TriggerExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Ajax_TriggerExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

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

Pada file .aspx dapat dilihat baris berikut :

```
<Triggers>
    <asp:AsyncPostBackTrigger ControlID="Button1" />
</Triggers>
```

Pada tag Triggers dapat didaftarkan control yang akan men-trigger secara asynchronus dengan control `<asp:AsyncPostBackTrigger ... />`. Dengan control ini control-control yang berada diluar area UpdatePanel dapat didaftarkan seperti bila control-control berada pada area UpdatePanel.

Pada tag Trigger juga terdapat control lain yaitu : `<asp:PostBackTrigger ... />` yang berfungsi mendaftarkan control (biasanya control yang berada pada area UpdatePanel) agar melakukan proses full postback.

Berikut contoh penggunaannya.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="TriggerExample.aspx.cs"
Inherits="Ajax_TriggerExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>
            <asp:UpdatePanel ID="UpdatePanel1" runat="server">
                <ContentTemplate>
                    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
                    <asp:Button ID="Button1" runat="server" Text="Button"
                        onclick="Button1_Click" />
                </ContentTemplate>
                <Triggers>
                    <asp:PostBackTrigger ControlID="Button1" />
                </Triggers>
            </asp:UpdatePanel>
        </div>
    </form>
</body>
</html>
```

```

</div>
</form>
</body>
</html>

```

Code file :

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Ajax_TriggerExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
    }
}

```

UpdateProgress

Control ini berfungsi untuk sebagai penanda bila proses sedang asynchronous postback sedang dilakukan. Penanda ini dapat berupa tulisan dan gambar, sebagai contoh :

1. Tulisan : Loading...silakan tunggu.
2. Gambar : 

Berikut contoh penggunaan control UpdateProgress.

```

UpdateProgressExample.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="UpdateProgressExample.aspx.cs" Inherits="Ajax_UpdateProgressExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>

            <asp:UpdateProgress ID="UpdateProgress1" runat="server">
                <ProgressTemplate>
                    Loading...silakan tunggu.
                </ProgressTemplate>
            </asp:UpdateProgress>

            <asp:UpdatePanel ID="UpdatePanel1" runat="server">
                <ContentTemplate>
                    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
                    <asp:Button ID="Button1" runat="server" Text="Button"
                        onclick="Button1_Click" />
                </ContentTemplate>
            </asp:UpdatePanel>
        </div>
    </form>

```

```

</asp:UpdatePanel>
</div>
</form>
</body>
</html>

```

Code file :

```

UpdateProgressExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Ajax_UpdateProgressExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
        for (int i = 0; i < 1000000000; i++)
        {
        }
    }
}

```

Bila terdapat lebih dari satu control UpdatePanel, maka control UpdateProgress akan berlaku untuk semua UpdatePanel tersebut. Kecuali pada control UpdateProgress disebutkan control UpdatePanel yang mana yang akan ditangani. Atribut yang menentukan UpdatePanel mana yang akan ditangani oleh control UpdateProgress dapat dilihat pada contoh berikut ini.

```

<asp:UpdateProgress AssociatedUpdatePanelID="UpdatePanel1"
    ID="UpdateProgress1" runat="server">
    <ProgressTemplate>
        Loading....silakan tunggu.
    </ProgressTemplate>
</asp:UpdateProgress>

```

Timer

Control Timer memungkinkan suatu proses dapat dilakukan secara terjadwal dalam rentang waktu tertentu. Proses dapat dilakukan secara full postback atau asynchronous tergantung dari letak control UpdateProgress. Bila berada di dalam UpdatePanel atau di daftarkan di dalam tag Triggers sebagai control yang bersifat Asynchronous maka proses akan dilakukan secara Asynchronous seperti pada contoh berikut ini.

```

TimerExample.aspx
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="TimerExample.aspx.cs"
Inherits="Ajax_TimerExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>

```

```

</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>
            <asp:UpdatePanel ID="UpdatePanel1" runat="server">
                <ContentTemplate>
                    <asp:Timer ID="Timer1" Interval="60" OnTick="Timer1_Tick"
                        runat="server">
                    </asp:Timer>

                    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
                </ContentTemplate>
            </asp:UpdatePanel>
        </div>
    </form>
</body>
</html>

```

Code file :

```

TimerExample.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Ajax_TimerExample : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void Timer1_Tick(object sender, EventArgs e)
    {
        Label1.Text = DateTime.Now.ToString();
    }
}

```

Ajax Control Toolkit

Ajax Control Toolkit merupakan contro-control yang berhubungan dengan antarmuka dinamis dengan memanfaatkan ASP.NET Ajax. Dengan Ajax Control Toolkit dapat dibuat antarmuka yang menarik dari segi tampilan dan animasi.

Berikut adalah control-control yang ada pada Ajax Control Toolkit :

1. Accordion
2. AlwaysVisibleControl
3. Animation
4. AutoComplete
5. Calendar
6. CascadingDropDown
7. CollapsiblePanel
8. ConfirmButton
9. DragPanel

10. DropDownList
11. DropShadow
12. DynamicPopulate
13. FilteredTextBox
14. HoverMenu
15. ListSearch
16. MaskedEdit
17. ModalPopup
18. MutuallyExclusiveCheckBox
19. NoBot
20. NumericUpDown
21. PagingBulletedList
22. PasswordStrength
23. PopupControl
24. Rating
25. ReorderList
26. ResizableControl
27. RoundedCorners
28. Slider
29. SlideShow
30. Tabs
31. TextBoxWatermark
32. ToggleButton
33. UpdatePanelAnimation
34. ValidatorCallout

Mendaftarkan Library Ajax Control Toolkit pada Web.config

Apabila ingin menggunakan Ajax Control Toolkit pada Web Site atau Web Application, maka dapat di daftarkan pada Web.config dengan baris seperti berikut ini.

```
...
<pages>
    <controls>
        <add namespace="AjaxControlToolkit"
            assembly="AjaxControlToolkit"
            tagPrefix="ajaxToolkit"/>

        ...
    </controls>
</pages>
...
```

Tambahkan AjaxControlToolkit.dll pada folder Bin pada Web Site. Bila menggunakan Web Application, dapat menambahkan AjaxControlToolkit.dll sebagai Reference.

Control Penggunaan

Pada bagian ini akan diberikan beberapa contoh penggunaan Ajax Control Toolkit.

Calendar

CalendarExample.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="CalendarExample.aspx.cs"
Inherits="ACT_CalendarExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>

            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

            <ajaxToolkit:CalendarExtender TargetControlID="TextBox1" runat="server">
            </ajaxToolkit:CalendarExtender>
        </div>
    </form>
</body>
</html>

```

FilteredTextBox

```

<%@ Page Language="C#" AutoEventWireup="true" CodeFile="FilterTextExample.aspx.cs"
Inherits="ACT_FilterTextExample" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:ScriptManager ID="ScriptManager1" runat="server">
            </asp:ScriptManager>

            <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

            <ajaxToolkit:FilteredTextBoxExtender ID="ftbe" runat="server"
                TargetControlID="TextBox1"
                FilterType="Custom, Numbers"
                ValidChars="+-=/*(). ." />
        </div>
    </form>
</body>
</html>

```

10

Membership, Profile dan Navigasi

Dalam membuat aplikasi terdapat beberapa hal yang mesti ada, yaitu :

1. Pengelolaan Role dan User yang dapat mengakses aplikasi.
2. Pengelolaan informasi User yang mudah sehingga dapat diubah tanpa harus mengubah banyak kode dan struktur database.
3. Navigasi yang mudah dibuat dan dapat terkait dengan Role, sehingga menu navigasi dapat ditampilkan sesuai dengan Role dari User yang sedang membuka aplikasi.

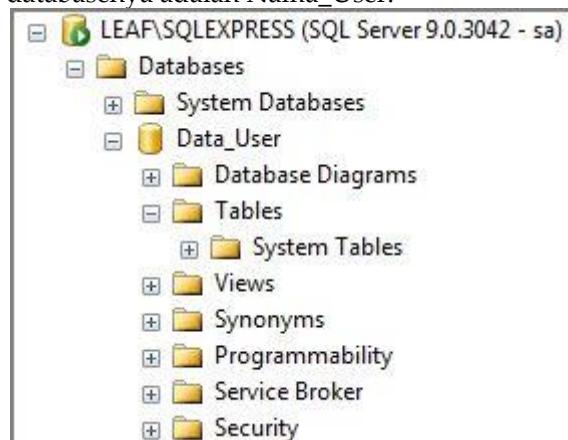
Pembuatan item-item di atas tidak lepas dari pembuatan database untuk mendukung itu semua.

Membership

Menyiapkan Database untuk Membership

Berikut langkah-langkah yang mesti diikuti untuk membuat database yang akan mendukung feature Membership pada ASP.NET.

1. Buat database pada SQL Server 2005 atau SQL Server Express Edition, misal nama databasenya adalah Nama_User.

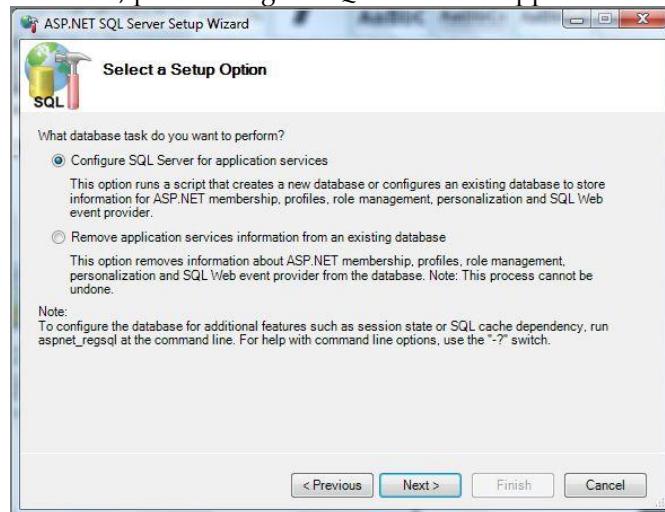


Dapat dilihat pada database Data_User masih terlihat berlum terdapat tabel.

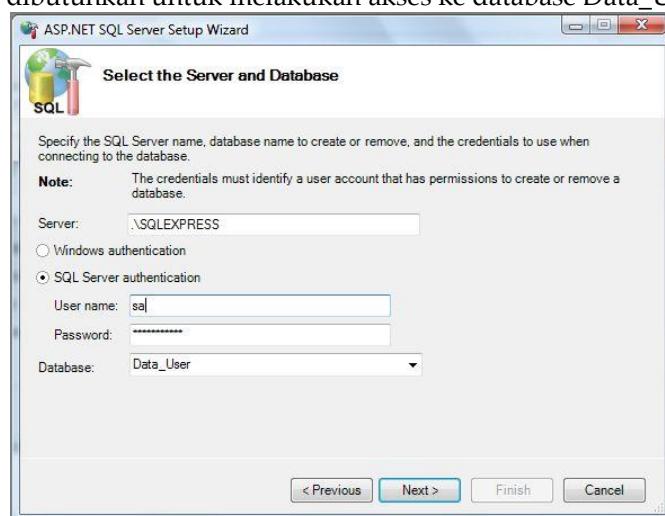
2. Eksekusi file aspnet_regsql.exe yang berada pada direktori C:\Windows\Microsoft.NET\Framework\v2.0.50727. Maka akan ditampilkan window ASP.NET SQL Server Setup Wizard.



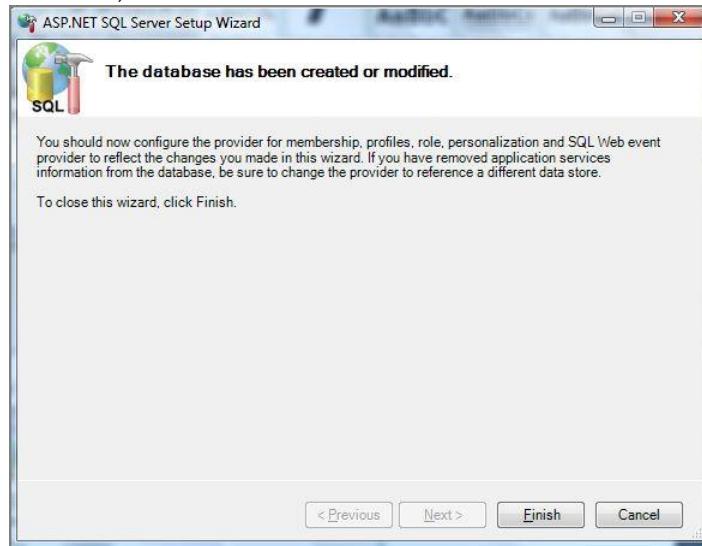
Klik Next, pilih Configure SQL Server for application services.



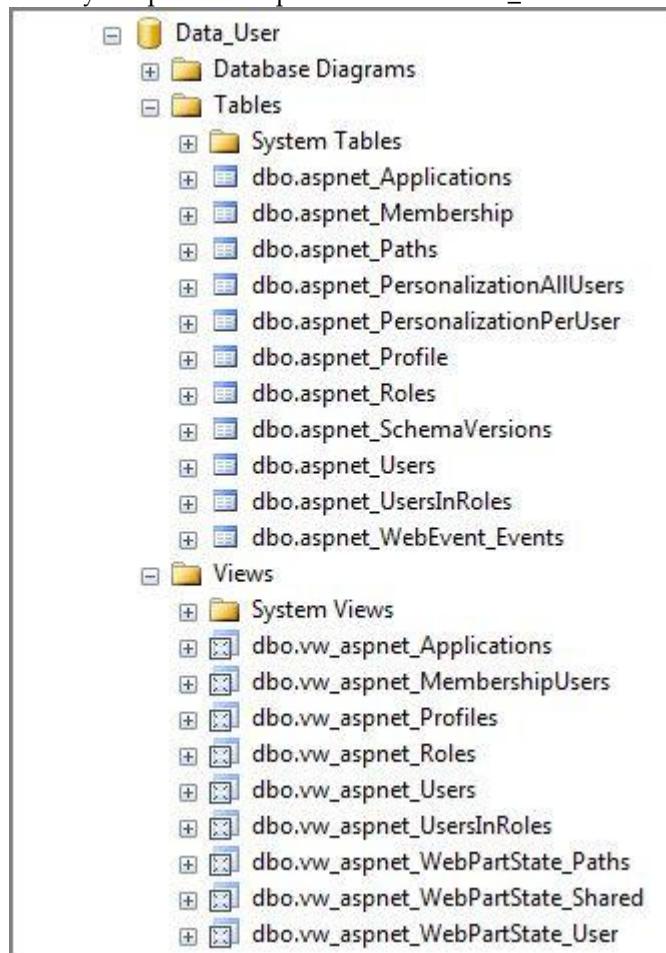
3. Klik Next dan pada bagian Select the Server and Database isikan nilai-nilai yang dibutuhkan untuk melakukan akses ke database Data_User.



4. Klik Next, kembali Next dan terakhir tombol Finish.



5. Hasilnya dapat dilihat pada database Data_User.



Proses yang baru saja dilakukan akan membuat tabel, view bahkan store procedure yang dibutuhkan untuk digunakan saat feature membership digunakan.

Menyiapkan Web.config

Setelah database dibuat, agar dapat digunakan pada Web Site atau Web Application maka perlu penambahan konfigurasi pada Web.config.

Berikut ini baris kode yang mesti ditambahkan pada Web.config.

1. Membuat Connection String untuk melakukan koneksi pada database Data_User.

```
...
<connectionStrings>
    <add name="DataUserConnectionString"
        connectionString="Data Source=.\SQLEXPRESS;
                          Initial Catalog=Data_User;
                          User ID=sa;Password=rahasia"
        providerName="System.Data.SqlClient"/>
</connectionStrings>
...
```

Control di atas merupakan Connection String untuk koneksi ke SQL Server Express, jadi bila ingin melakukan koneksi ke SQL Server 200x maka Connection String yang digunakan adalah seperti berikut.

```
...
<connectionStrings>
    <add name="DataUserConnectionString"
        connectionString="Data Source=.;
                          Initial Catalog=Data_User;
                          User ID=sa;Password=rahasia"
        providerName="System.Data.SqlClient"/>
</connectionStrings>
...
```

2. Selanjutnya menambahkan baris berikut.

```
...
<system.web>
    ...
<roleManager enabled="true"
    defaultProvider="AspNetSqlRoleProvider">
    <providers>
        <clear/>
        <add name="AspNetSqlRoleProvider"
            connectionStringName="DataUserConnectionString"
            applicationName="MSP"
            type="System.Web.Security.SqlRoleProvider,
                  System.Web, Version=2.0.0.0, Culture=neutral,
                  PublicKeyToken=b03f5f7f11d50a3a"/>
    </providers>
</roleManager>
<membership defaultProvider="AspNetSqlMembershipProvider">
    <providers>
        <clear/>
        <add name="AspNetSqlMembershipProvider"
            type="System.Web.Security.SqlMembershipProvider,
                  System.Web, Version=2.0.0.0, Culture=neutral,
                  PublicKeyToken=b03f5f7f11d50a3a"
            connectionStringName="DataUserConnectionString"
            enablePasswordRetrieval="false"
            enablePasswordReset="true"
            requiresQuestionAndAnswer="false" applicationName="MSP"
            requiresUniqueEmail="true" passwordFormat="Hashed"
            maxInvalidPasswordAttempts="5" minRequiredPasswordLength="1"
            minRequiredNonalphanumericCharacters="0" passwordAttemptWindow="10"
            passwordStrengthRegularExpression="" />
    </providers>
</membership>
    ...
</system.web>
...
```

Pada kode di atas yang mesti diperhatikan adalah atribut :

1. connectionStringName, yang merupakan nilai atribut name yang ada pada Connection String untuk koneksi ke database Data_User.
2. applicationName, adalah identitas dari aplikasi web. Dapat satu Web Site atau Web Application nama ini harus sama dan ketika sudah digunakan satu kali maka tidak bisa diubah lagi, karena sangat erat hubungannya dengan database pada Data_User.
3. Modifikasi nilai atribut mode pada bagian authentication dari .

```
<authentication mode="Windows" />
```

Menjadi :

```
<authentication mode="Forms" />
```

4. Untuk melakukan testing keberhasilan koneksi dan konfigurasi tersebut di atas dapat dilakukan dengan menggunakan ASP.NET Configuration yang dapat dieksekusi dengan menekan tombolnya pada Solution Explorer (dalam lingkaran merah).



Pilih Web Site atau Web Application pada Solution Explorer kemudian klik tombol ASP.NET Configuration.

5. Bila koneksi dan konfigurasi yang dilakukan sudah benar maka setelah tombol ASP.NET Configuration diklik dapat dilihat halaman berikut ini pada web browser.



Keberhasilan koneksi dan konfigurasi dapat dilihat dengan masuk ke bagian Security. Bila berhasil maka dapat dilihat halaman seperti berikut.

You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

[Use the security Setup Wizard to configure security step by step.](#)

Click the links in the table to manage the settings for your application.

Users	Roles	Access Rules
Existing users: 0 Create user Manage users Select authentication type	Existing roles: 0 Disable Roles Create or Manage roles	Create access rules Manage access rules

6. Pada bagian Security ini ada 3 hal yang dilakukan yaitu :
 - a. Membuat Role.
 - b. Membuat User.
 - c. Membuat Access Rule.

Mengelola Role

Untuk membuat Role atau mengedit Role langkah yang dilakukan adalah sebagai berikut :

1. Klik Create or Manage roles.
2. Isi nama role yang diinginkan pada textbox New role name dan klik tombol Add Role.

Role Name	Add/Remove Users
Administrator	Manage Delete

Mengelola User

Untuk membuat user langkah yang dilakukan adalah :

1. Klik Create user.

2. Isi informasi user pada form.

Add a user by entering the user's ID, password, and e-mail address on this page.

Create User	Roles
Sign Up for Your New Account User Name: <input type="text" value="reyza"/> Password: <input type="password"/> Confirm Password: <input type="password"/> E-mail: <input type="text" value="reza.faisal@gmail.com"/>	Select roles for this user: <input checked="" type="checkbox"/> Administrator <input checked="" type="checkbox"/> Member
<input type="button" value="Create User"/>	
<input checked="" type="checkbox"/> Active User	

3. Satu user dapat memiliki lebih dari satu role, seperti pada contoh di atas, user reyza dapat memiliki role sebagai Administrator dan Member.
4. Klik tombol Create User.
5. Untuk melihat daftar user dapat dilakukan dengan memilih Manage users, maka dapat dilihat daftar user seperti berikut.

Click a row to select a user and then click **Edit user** to view or change the user's password or other properties.
To assign roles to the selected user, select the appropriate check boxes on the right.

To prevent a user from logging into your application but retain his or her information in your database, set the status to inactive by clearing the check box.

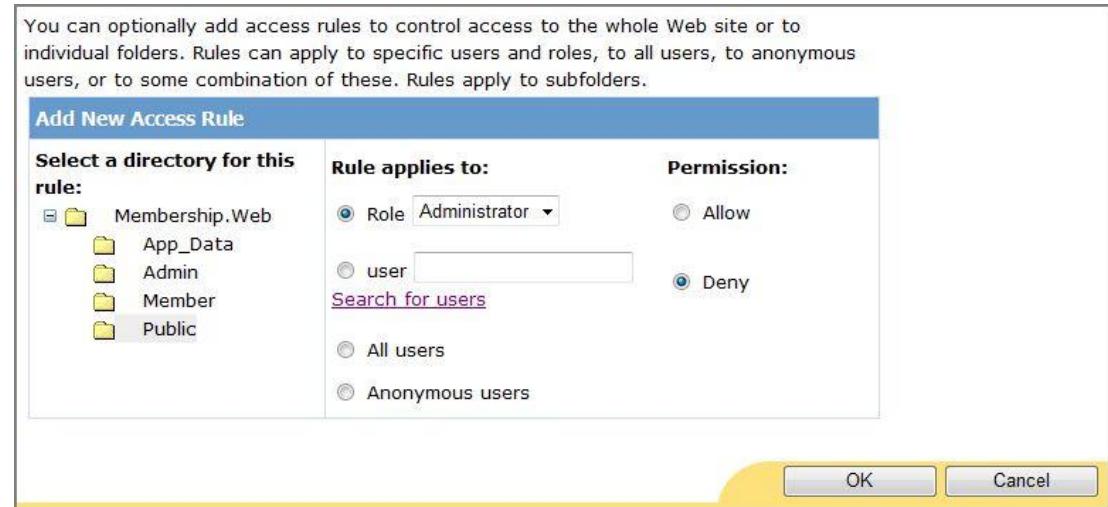
Search for Users		
Search by: <input type="text" value="User name"/> for: <input type="text"/> Wildcard characters * and ? are permitted. A B C D E F G H I J K L M N O P Q R S T U V W X Y Z All		
Active	User name	Roles
<input checked="" type="checkbox"/>	reyza	Edit user Delete user Edit roles
Create new user Back		

Mengelola Access Rule

Misal pada Web Site atau Web Application terdapat folder-folder, maka dapat diatur aturan pengaksesannya dari bagian Security > Access Rule.

Access Rules
Create access rules Manage access rules

Dengan memilih Crate access rules atau Manage access rules maka dapat dilihat antarmuka seperti berikut ini.

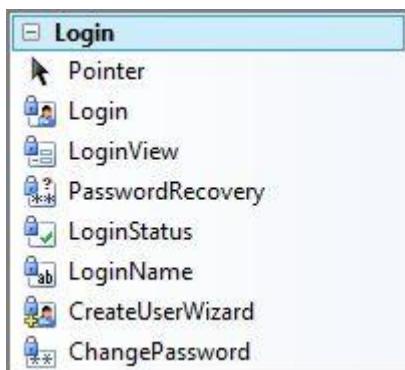


Dengan fasilitas ini dapat diatur hak akses folder atau file-file yang di dalam folder dapat di akses oleh role atau user apa saja.

Login Control

Pada toolbox dapat dilihat control-control di dalam group Login. Control-control ini memudahkan pemanfaatan penggunaan feature membership, dari keperluan untuk membuat antarmuka untuk login sampai proses authentifikasi.

Berikut ini adalah control-control yang terdapat pada group Login.



Untuk mengenal dan mengerti penggunaan control-control tersebut maka pada bagian ini akan dibuat beberapa kasus.

Authentifikasi User

Kasus pertama adalah akan dibuat halaman login dan halaman yang hanya akan dapat diakses ketika user sudah berhasil melewati proses authentifikasi.

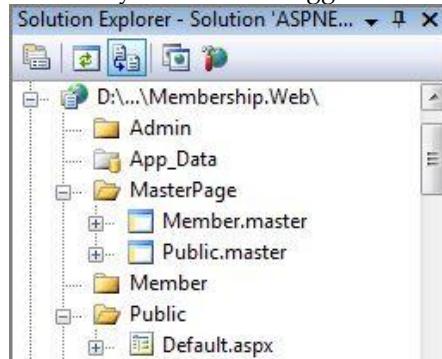
Berikut langkah-langkah yang mesti diikuti.

1. Membuat file .master untuk Public (yang dapat dapat diakses oleh secara anonymous) dan Member. File-file tersebut adalah.
 - a. Public.master.
 - b. Member.master.

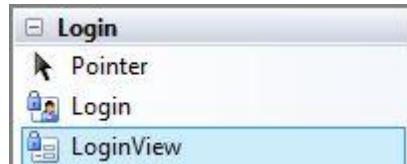
Kedua file tersebut disimpan pada direktori MasterPages.



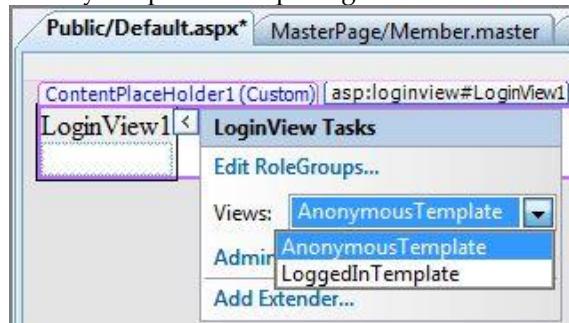
2. Tambahkan folder Public pada Web Site dan tambahkan file Default.aspx didalamnya. File ini menggunakan Public.master sebagai master page.



Pada file Default.aspx, ubah mode ke Design dan drag-n-drop control LoginView dari toolbox ke dalam halaman Default.aspx.



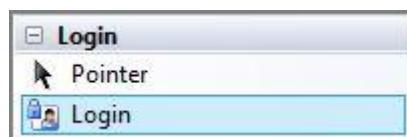
Hasilnya dapat dilihat pada gambar berikut ini.



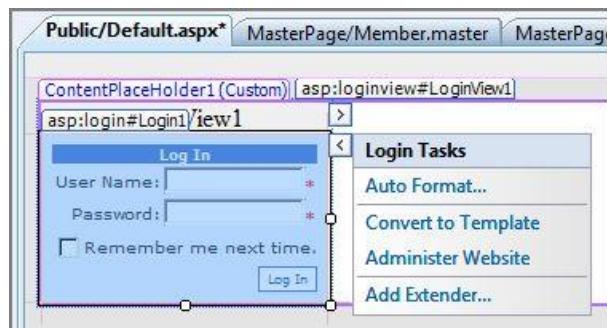
Control LoginView mempunyai dua template yaitu :

- AnonymousTemplate, yaitu antarmuka yang dapat dilihat oleh user anonymous atau user yang belum login.
- LoggedInTemplate, yaitu antarmuka yang dapat dilihat setelah user login.

Pada AnonymousTemplate tambahkan control Login.

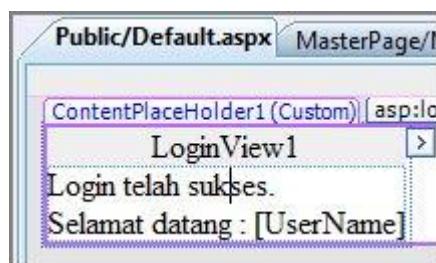


Maka hasilnya dapat dilihat seperti berikut.

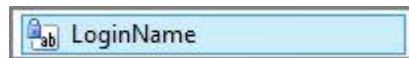


Untuk mengubah style control Login dapat diubah dengan memilih Auto Format.

Selanjutnya ubah view LoginView ke LoggedInTemplate, ketikkan text seperti yang terlihat pada gambar.



Setelah itu tambahkan control LoginName.



Control ini berfungsi untuk menampilkan nama user yang berhasil login.

3. Simpan dan View in browse.

Pada awal halaman di liat pada browser akan dilihat form login seperti pada gambar.



Setelah dimasukkan user name dan password yang sesuai dengan data yang ada maka tampilan halaman ini akan berubah seperti di atas.

Dari contoh di atas maka sudah diperkenalkan tiga control dari group Login, yaitu :

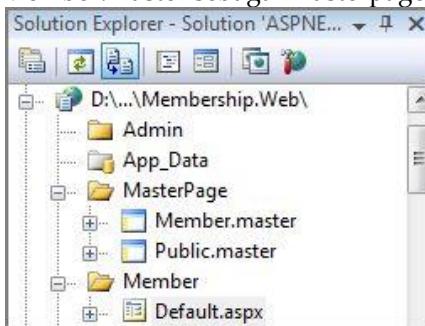
1. LoginView.
2. Login.
3. LoginName.

Mengamankan Halaman Member

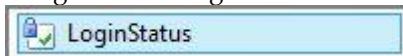
Halaman member disini adalah halaman yang hanya dapat diakses oleh user yang telah login. Misalnya halaman member akan disimpan pada folder Member dan menggunakan file Member.master sebagai master page. Bila user anonymous langsung mengakses file pada folder Member akan ditolak dan diantarkan pada halaman Public/Default.aspx agar login terlebih dahulu. Selain itu pada akan dibuat tombol logout.

Solusi dari contoh kasus ini adalah sebagai berikut.

1. Buat folder Member dan tambahkan file Default.aspx yang menggunakan file Member.master sebagai masterpage.



2. Pada file Member.master tambahkan control LoginStatus yang akan berfungsi sebagai tombol logout.



Dan berikut kode file Member.master yang telah menggunakan control LoginStatus.

```
Member.master
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="MasterPage__Member" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
    <asp:ContentPlaceHolder id="head" runat="server">
        </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:LoginStatus ID="LoginStatus1"
                LogoutText="Logout"
                LogoutPageUrl="~/Public/Default.aspx"
                runat="server" />
            <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

            </asp:ContentPlaceHolder>
        </div>
    </form>
</body>
</html>
```

3. Kemudian pada code file dari file Member.master tambahkan beberapa baris untuk pemeriksaan status user yang sedang akses halaman.

```
Member.master.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
```

```

using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Xml.Linq;

public partial class MasterPage_Member : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        bool IsLogin = HttpContext.Current.User.Identity.IsAuthenticated;

        if (!IsLogin)
        {
            Response.Redirect("~/Public/Default.aspx");
        }
    }
}

```

Penambahan baris kode di atas akan membuat user yang belum login akan diantar ke halaman Default.aspx yang berada pada folder Public.

Mengatur Akses Control Pada Halaman

Pada kasus berikut, dimisalkan dalam satu halaman ada dua bagian aplikasi kecil yang akan ditampilkan dalam suatu kondisi tertentu berdasarkan role dari user yang sedang mengakses halaman tersebut.

Misalnya pada halaman Default.aspx yang ada pada folder Member, terdapat fungsi Create User dan Ganti Password User Sendiri. Berikut adalah hak akses yang diinginkan berdasarkan role.

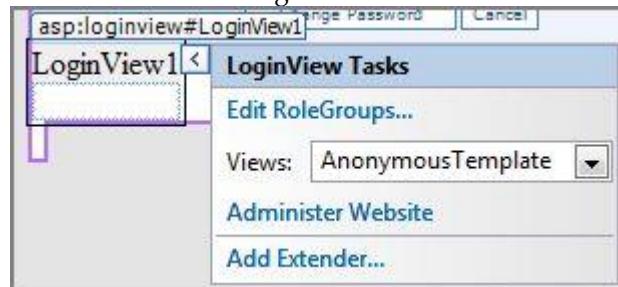
1. Role Administrator akan dapat mengakses kedua aplikasi tersebut.
2. Role Member hanya akan dapat mengakses fungsi Ganti Password User Sendiri, tetapi tidak dapat mengakses fungsi Create User.

Untuk kasus tersebut maka berikut adalah langkah-langkah yang harus dilakukan.

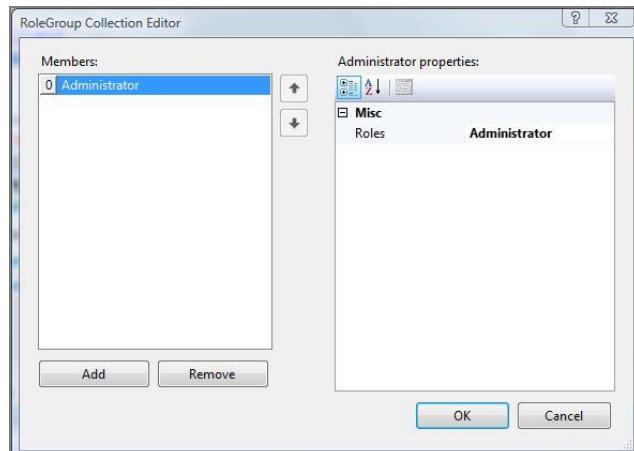
1. Pada file Member\Default.aspx tambahkan control ChangePassword, ubah style control ini dengan memilih Auto format.



2. Tambahkan control LoginView di bawah control ChangePassword.

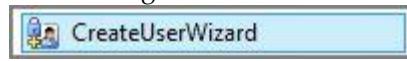


Klik EditRoleGroups, maka akan ditampilkan window RoleGroup Collection Editor.

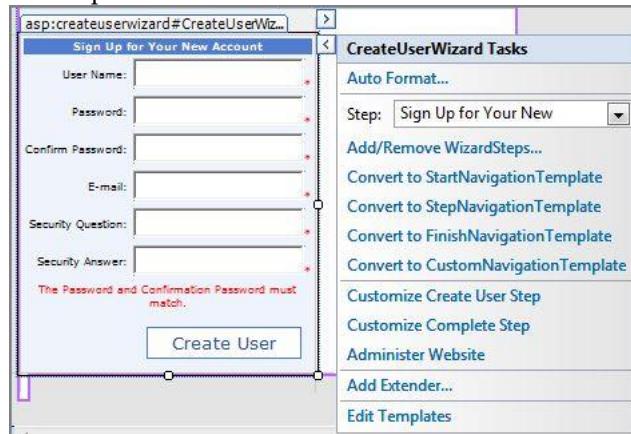


Klik tombol Add dan iskan nilai Administrator pada Roles. Kemudian klik OK. Maka pada View, selain AnonymousTemplate dan LoggedInTemplate terdapat tambahan RoleGroup[0] – Administrator.

3. Pilih RoleGroup[0] – Administrator, dan tambahkan control CreateUserWizard didalam LoginView.



Dan dapat dilihat antarmuka control CreateUserWizard seperti berikut.



4. Simpan dan View in browser.

Bila user yang mengakses halaman ini hanya memiliki role Member maka berikut ini tampilan yang dapat dilihat.



Bila user tersebut adalah user dengan role Administrator atau Administrator dan Member maka antarmuka seperti berikut yang dapat dilihat.



Berikut ini kode dari file Member/Default.aspx.

Default.aspx

```
<%@ Page Language="C#" MasterPageFile="~/MasterPage/Member.master"
AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="Member_Default"
Title="Untitled Page" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">

    <asp:ChangePassword ID="ChangePassword1" runat="server" BackColor="#EFF3FB"
        BorderColor="#B5C7DE" BorderPadding="4" BorderStyle="Solid" BorderWidth="1px"
        Font-Names="Verdana" Font-Size="0.8em">
        <CancelButtonStyle BackColor="White" BorderColor="#507CD1"
            BorderStyle="Solid"
            BorderWidth="1px" Font-Names="Verdana" Font-Size="0.8em"
            ForeColor="#284E98" />
        <PasswordHintStyle Font-Italic="True" ForeColor="#507CD1" />
        <ContinueButtonStyle BackColor="White" BorderColor="#507CD1"
            BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-
            Size="0.8em"
            ForeColor="#284E98" />
        <ChangePasswordButtonStyle BackColor="White" BorderColor="#507CD1"
            BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana" Font-
            Size="0.8em"
            ForeColor="#284E98" />
        <TitleTextStyle BackColor="#507CD1" Font-Bold="True" Font-Size="0.9em"
            ForeColor="White" />
        <TextBoxStyle Font-Size="0.8em" />
        <InstructionTextStyle Font-Italic="True" ForeColor="Black" />
    </asp:ChangePassword>
    <asp:LoginView ID="LoginView1" runat="server">
        <RoleGroups>
            <asp:RoleGroup Roles="Administrator">
                <ContentTemplate>
                    <asp:CreateUserWizard ID="CreateUserWizard1" runat="server"
                        BackColor="#EFF3FB"
                        BorderColor="#B5C7DE" BorderStyle="Solid" BorderWidth="1px"
                        Font-Names="Verdana" Font-Size="0.8em">
                        <SideBarStyle BackColor="#507CD1" Font-Size="0.9em"
                            VerticalAlign="Top" />
                        <SideBarButtonStyle BackColor="#507CD1" Font-Names="Verdana"
                            ForeColor="White" />
                        <ContinueButtonStyle BackColor="White" BorderColor="#507CD1"
                            BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana"
                            ForeColor="#284E98" />
                        <NavigationButtonStyle BackColor="White" BorderColor="#507CD1"
                            BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana"
                            ForeColor="#284E98" />
                    </asp:CreateUserWizard1>
                </ContentTemplate>
            </asp:RoleGroup>
        </RoleGroups>
    </asp:LoginView>
</asp:Content>
```

```

<HeaderStyle BackColor="#284E98" BorderColor="#EFF3FB"
    BorderStyle="Solid"
    BorderWidth="2px" Font-Bold="True" Font-Size="0.9em"
    ForeColor="White"
    HorizontalAlign="Center" />
<CreateUserButtonStyle BackColor="White" BorderColor="#507CD1"
    BorderStyle="Solid" BorderWidth="1px" Font-Names="Verdana"
    ForeColor="#284E98" />
<TitleTextStyle BackColor="#507CD1" Font-Bold="True"
    ForeColor="White" />
<StepStyle Font-Size="0.8em" />
<WizardSteps>
    <asp:CreateUserWizardStep runat="server" />
    <asp:CompleteWizardStep runat="server" />
</WizardSteps>
</asp:CreateUserWizard>
</ContentTemplate>
</asp:RoleGroup>
</RoleGroups>
</asp:LoginView>
<br />
<br />
</asp:Content>

```

Profile

Pada saat membuat user sebelumnya, tidak banyak informasi yang disimpan bila dilihat dari form berikut.



Dalam aplikasi sebenarnya user dapat mempunyai informasi lain seperti :

1. Nama Lengkap
2. Organisasi.
3. Jabatan.
4. Golongan.
5. Dan lain-lain.

Untuk menambahkan informasi di atas solusinya adalah dengan memanfaatkan Profile. Dengan menggunakan Profile, tidak perlu dibuat tabel khusus pada database sehingga bila dalam perjalanan pembuatan aplikasi ternyata ingin ditambahkan informasi lain maka proses penambahan tersebut tidak akan mengganggu struktur database dan juga tidak terlalu banyak memakan energi untuk menambahkannya dalam baris program.

Menyiapkan Web.config

Seperti penambahan feature membership di atas, maka saat melakukan penambahan feature Profile juga harus dilakukan modifikasi pada file Web.config.

Berikut ini adalah kode yang harus ditambahkan pada Web.config.

```
 . . .
<system.web>
. . .
<profile>
  <providers>
    <clear/>
    <add name="AspNetSqlProfileProvider"
      connectionStringName=" DataUserConnectionString "
      applicationName="MSP"
      type="System.Web.Profile.SqlProfileProvider, System.Web, Version=2.0.0.0,
      Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"/>
  </providers>
  <properties>
    <add name="NamaLengkap" type="System.String" serializeAs="String"
      allowAnonymous="false" defaultValue="" readOnly="false"/>
    <add name="Organisasi" type="System.String" serializeAs="String"
      allowAnonymous="false" defaultValue="" readOnly="false"/>
    <add name="Jabatan" type="System.String" serializeAs="String"
      allowAnonymous="false" defaultValue="" readOnly="false"/>
    <add name="Golongan" type="System.String" serializeAs="String"
      allowAnonymous="false" defaultValue="" readOnly="false"/>
  </properties>
</profile>
. . .
</system.web>
. . .
```

Pada kode di atas terdapat bagian properties yang dapat digunakan untuk menambahkan informasi yang ingin disimpan untuk suatu user. Dapat dilihat pada contoh di atas property yang akan dimiliki oleh user adalah :

1. NamaLengkap.
2. Organisasi.
3. Jabatan.
4. Golongan.

Operasi Profile

Pada bagian ini akan dipaparkan bagaimana cara menyimpan property yang telah didaftarkan pada Web.config selain itu juga akan ditunjukkan dimana data tersebut disimpan dan bagaimana mengambilnya kembali untuk ditampilkan.

Berikut ini contoh kode untuk operasi update data profile user.

```
ProfileCommon profileCommon = Profile.GetProfile("reyza");
profileCommon.NamaLengkap = "M Reza Faisal";
profileCommon.Organisasi = "INDC";
profileCommon.Golongan = "Pemula";
profileCommon.Jabatan = "Junior Programmer";
profileCommon.Save();
```

Data profile user ini disimpan pada tabel aspnet_Profile.

The screenshot shows the Object Explorer on the left with 'Databases' expanded, showing 'System Databases', 'Data_User', 'Database Diagrams', and 'Tables'. Under 'Tables', 'System Tables' is expanded, showing 'dbo.aspnet_Applications', 'dbo.aspnet_Membership', 'dbo.aspnet_Paths', 'dbo.aspnet_PersonalizationAll', 'dbo.aspnet_PersonalizationPer', and 'dbo.aspnet_Profile'. On the right, the 'aspnet_Profile' table is selected, displaying a single row of data:

	UserId	PropertyNames	PropertyValues...	PropertyValues...	LastUpdatedDate
▶	979c-6dc278ac93db	Jabatan:S:0:17:...	Junior Programm...	<Binary data>	7/19/2008 8:39:...
*	NULL	NULL	NULL	NULL	NULL

Berikut ini contoh untuk mengambil nilai profile dari user dan menampilkannya pada halaman web.

ShowProfile.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ShowProfile.aspx.cs"
Inherits="Profile_ShowProfile" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <table>
                <tr>
                    <td>Username</td>
                    <td>:</td>
                    <td>
                        <asp:Label ID="Label1" runat="server" Text="Label">
                            </asp:Label>
                    </td>
                </tr>
                <tr>
                    <td>Nama Lengkap</td>
                    <td>:</td>
                    <td>
                        <asp:Label ID="Label2" runat="server" Text="Label">
                            </asp:Label>
                    </td>
                </tr>
                <tr>
                    <td>Organisasi</td>
                    <td>:</td>
                    <td>
                        <asp:Label ID="Label3" runat="server" Text="Label">
                            </asp:Label>
                    </td>
                </tr>
                <tr>
                    <td>Jabatan</td>
                    <td>:</td>
                    <td>
                        <asp:Label ID="Label4" runat="server" Text="Label">
                            </asp:Label>
                    </td>
                </tr>
                <tr>
                    <td>Golongan</td>
                    <td>:</td>
                    <td>
                        <asp:Label ID="Label5" runat="server" Text="Label">
                            </asp:Label>
                    </td>
                </tr>
            </table>
        </div>
    </form>
</body>
</html>
```

```

        </td>
    </tr>
</table>
</div>
</form>
</body>
</html>

```

Code file :

```

ShowProfile.aspx.cs
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

public partial class Profile_ShowProfile : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            ProfileCommon profileCommon = Profile.GetProfile("reyza");
            Label1.Text = "reyza";
            Label2.Text = profileCommon.NamaLengkap;
            Label3.Text = profileCommon.Organisasi;
            Label4.Text = profileCommon.Jabatan;
            Label5.Text = profileCommon.Golongan;
        }
        catch (Exception ex)
        {

        }
    }
}

```

Navigasi

Navigasi adalah hal yang sangat dibutuhkan dalam suatu aplikasi. Navigasi dapat berupa menu dan site map path.

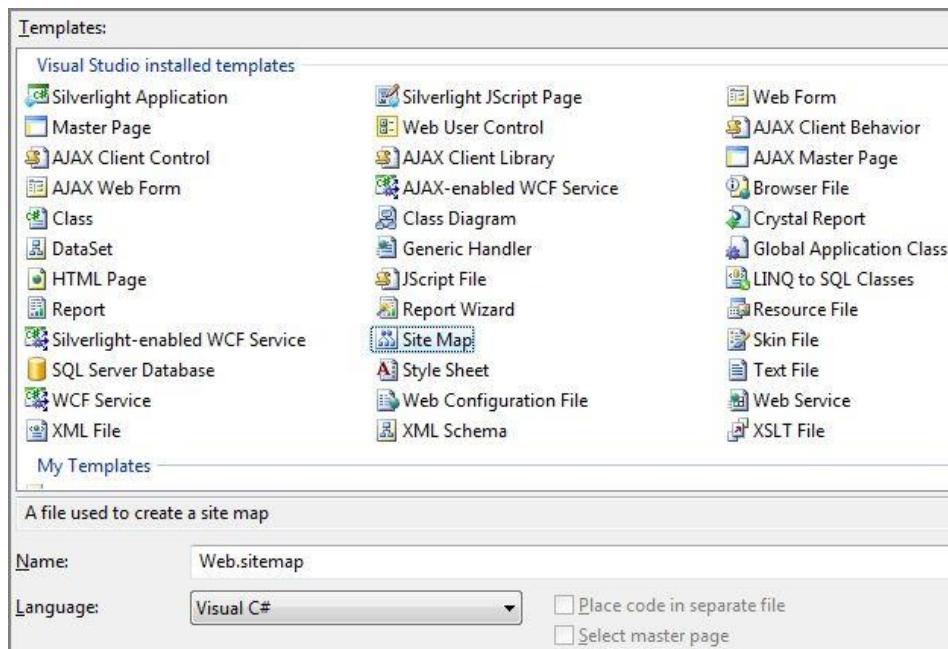
Menu berfungsi untuk berpindah ke halaman yang dituju. Sedangkan site map path berfungsi mengetahui lokasi file.

Untuk mengetahui control-control apa saja yang digunakan dan juga hal-hal lain yang diperlukan dapat mengikuti langkah-langkah berikut di bawah ini.

Sitemap

Sitemap adalah file dengan format XML yang berfungsi untuk menyimpan daftar file-file yang dapat di akses pada suatu Web Site. Pada Sitemap setiap file dapat didaftarkan role yang dapat mengakses halaman tersebut.

Untuk menambahkan file Sitemap pada Web Site atau Web Application dapat dilakukan dengan cara, klik kanan pada Web Site atau Web Application kemudian pilih Add New Item pada window Add New Item pilih Site Map dan beri nama dengan Web.sitemap.



Berikut ini contoh isi dari file Web.sitemap

```
Web.sitemap
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
    <siteMapNode url "~/Dashboard/Default.aspx" title="Home" description="Home"
        roles="Administrator, Member">
        <siteMapNode url "~/Admin/Default.aspx" title="Admin" description="Admin"
            roles="Administrator" >
            <siteMapNode url "~/Admin/Default1.aspx" title="Pengelolaan User"
                description="Pengelolaan User" roles="Administrator" />
            <siteMapNode url "~/Admin/Default2.aspx" title="Pengeloaan Role"
                description="Pengeloaan Role" roles="Administrator" />
            <siteMapNode url "~/Admin/Default3.aspx" title="Pengeloaan Akses"
                description="Pengeloaan Akses" roles="Administrator" />
        </siteMapNode>
        <siteMapNode url "~/Member/Default.aspx" title="Member" description="Member"
            roles="Member" >
            <siteMapNode url "~/Member/Default1.aspx" title="Blog" description="Blog"
                roles="Member" />
            <siteMapNode url "~/Member/Default2.aspx" title="Forum" description="Forum"
                roles="Member" />
            <siteMapNode url "~/Member/Default3.aspx" title="File" description="File"
                roles="Member" />
        </siteMapNode>
    </siteMapNode>
</siteMap>
```

Yang mesti diperhatikan dalam mengisi data pada Sitemap adalah nilai url pada setiap note tidak boleh sama.

Untuk menggunakan Sitemap ini sebagai menu pada Web Site atau Web Application maka perlu dilakukan konfigurasi pada Web.config.

Berikut adalah kode yang harus digunakan untuk melakukan konfigurasi tersebut.

```
...
<system.web>
...
<siteMap enabled="true">
    <providers>
        <add name="MSPSSiteMap" siteMapFile="~/Web.sitemap"
```

```

        type="System.Web.XmlSiteMapProvider" securityTrimmingEnabled="true"/>
    </providers>
  </siteMap>
  . . .
</system.web>
. . .

```

Menu dan TreeView

Menu dapat ditampilkan dengan control Menu dan TreeView yang ada pada group Navigation pada Toolbox.

Selain itu juga dibutuhkan control SiteMapDataSource yang ada pada group Data sebagai data source dari control Menu dan TreeView.

Menu

Berikut contoh menampilkan menu dengan control Menu.

Member.master

```

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="MasterPage_Member" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Untitled Page</title>
  <asp:ContentPlaceHolder id="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:SiteMapDataSource ID="SiteMapDataSource1" ShowStartingNode="false"
        SiteMapProvider="MSPSiteMap" runat="server" />
      <asp:Menu ID="Menu1" DataSourceID="SiteMapDataSource1"
        Orientation="Horizontal" runat="server">
      </asp:Menu>
      <asp:LoginStatus ID="LoginStatus1" LogoutText="Logout"
        LogoutPageUrl "~/Public/Default.aspx" runat="server" />
      <br />
      <br />
      <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
    </div>
  </form>
</body>
</html>

```

TreeView

Berikut adalah contoh penggunaan TreeView.

Member.master

```

<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="MasterPage_Member" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Untitled Page</title>
  <asp:ContentPlaceHolder id="head" runat="server">

```

```
</asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:SiteMapDataSource ID="SiteMapDataSource1" ShowStartingNode="false"
                SiteMapProvider="MSPSiteMap" runat="server" />

            <asp:TreeView ID="TreeView1" DataSourceID="SiteMapDataSource1"
                runat="server">
            </asp:TreeView>
            <asp:LoginStatus ID="LoginStatus1" LogoutText="Logout"
                LogoutPageUrl="~/Public/Default.aspx" runat="server" />
            <br />
            <br />
            <asp:ContentPlaceHolder id="ContentPlaceHolder1" runat="server">

                </asp:ContentPlaceHolder>
            </div>
        </form>
    </body>
</html>
```

11

Reporting

Reporting merupakan salah satu bagian yang penting dalam aplikasi sistem informasi, di dalam Visual studio 2008 disediakan tool atau komponen untuk pembuatan report, diantaranya Microsoft Report Viewer dan Crystal Report. Sepanjang bab ini kami akan membahas hanya Microsoft Report Viewer untuk pembuatan report-report. Seperti apa Microsoft Report Viewer itu, berikut pembahasannya :

Microsoft ReportViewer

ReportViewer adalah suatu control yang dapat didistribusikan secara bebas, yang memungkinkan report ditambahkan ke aplikasi yang dibangun menggunakan framework .NET. report-report di rancang dengan kemudahan tool drag-and-drop menggunakan report designer yang sudah termasuk dalam Visual Studio 2008 (standard edition ke atas).

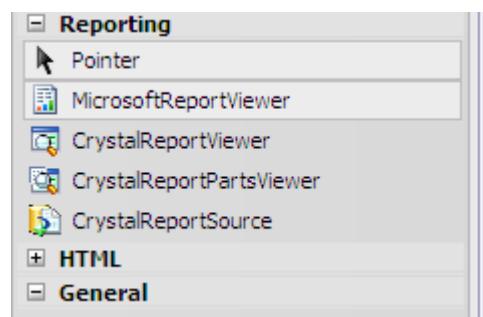
Pada ASP.NET reportviewer dapat ditambahkan pada project web dengan extention file (.rdlc), seperti gambar berikut :



Seperti apa report designer yang ditawarkan report viewer untuk kemudahan design report, berikut contoh tampilan report viewer designer :

Peralatan Keselamatan (Dilisi oleh PPR)			Tindak Pencegahan Khusus (Dilisi oleh PPR) (Aspek Keselamatan, Kesehatan & Lingkungan dan Kualitas) <i>#Fields#PRECAUTION Value</i>								
PENGESAHAN											
SIC Pekerjaan = PIG = <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">PIC</td> <td style="width: 33%;">Ass. SM</td> <td style="width: 33%;">SM</td> </tr> </table>			PIC	Ass. SM	SM	SIC Area = PIG = <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">PIC</td> <td style="width: 33%;">Ass. SM</td> <td style="width: 33%;">SM</td> </tr> </table> <p>Cat.: Kolom dasar ditandatangani setelah mendapatkan safety treatment selesai dilakukan</p>			PIC	Ass. SM	SM
PIC	Ass. SM	SM									
PIC	Ass. SM	SM									
Beksi Terkait = PIG = <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">PIC</td> <td style="width: 33%;">Ass. SM</td> <td style="width: 33%;">SM</td> </tr> </table>			PIC	Ass. SM	SM	Beksi Terkait = PIG = <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">PIC</td> <td style="width: 33%;">Ass. SM</td> <td style="width: 33%;">SM</td> </tr> </table>			PIC	Ass. SM	SM
PIC	Ass. SM	SM									
PIC	Ass. SM	SM									
SERAH TERIMA SELESAINYA PEKERJAAN											
PERIKSA HAL-HAL BERIKUT <ul style="list-style-type: none"> <input type="checkbox"/> Areal dan peralatan sudah bersih <input type="checkbox"/> Alat sudah dikembalikan <input type="checkbox"/> 			SIC Pekerjaan SIC Area SIC Area <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 33%;">PIC</td> <td style="width: 33%;">Ass. SPV/GL</td> <td style="width: 33%;">PIC/Ass. SM/SM</td> </tr> </table> <p>Telepon : - CCR TPA 850, 351, 352 - CCR PET 850, 551 - Clinic 230, 281, 282 </p>			PIC	Ass. SPV/GL	PIC/Ass. SM/SM			
PIC	Ass. SPV/GL	PIC/Ass. SM/SM									
Ijin DIBATALKAN jika SIRINE keadaan darurat berbunyi											
<small>Asli (Putih) : Teknisi atau Subkontraktor (dipasang di lokasi pekerjaan). Form ini dikembalikan ke SIC pekerjaan setelah pekerjaan selesai dan disimpan.</small>											
MITSUBISHI CHEMICAL INDONESIA		RADIATION WORK PERMIT (IJIN KERJA RADIASI)		HARIAN No.: #Parameters!NoSwp Value Periode : #Parameters!Tgl Val							
Nama Pekerjaan (Dilisi oleh SIC Pekerjaan) <i>#Fields#WORK_TITLE Value</i>		Pelaksana <input checked="" type="checkbox"/> Teknisi <input type="checkbox"/> Subkontraktor #Fields#SUBCONTRACTOR_NAME Value									

Untuk menampilkan report yang sudah di-design, kita memerlukan page ASP.NET yang didalamnya terdapat control Microsoft Report Viewer. Tambahkan control Microsoft Report Viewer control yang terdapat pada Toolbox. Terlihat pada gambar dibawah ini :



Drag control MicrosoftReportViewer kedalam page ASP.NET.

Berikut code yang terlihat pada halaman ASP.NET :

```

<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportOrders.aspx.cs" Inherits="ReportOrders" %>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms,
Version=9.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <rsweb:ReportViewer ID="ReportViewer1" runat="server">
                </rsweb:ReportViewer>
            </div>
        </form>
    </body>
</html>

```

Ada beberapa keuntungan jika kita menggunakan control reportviewer diantaranya :

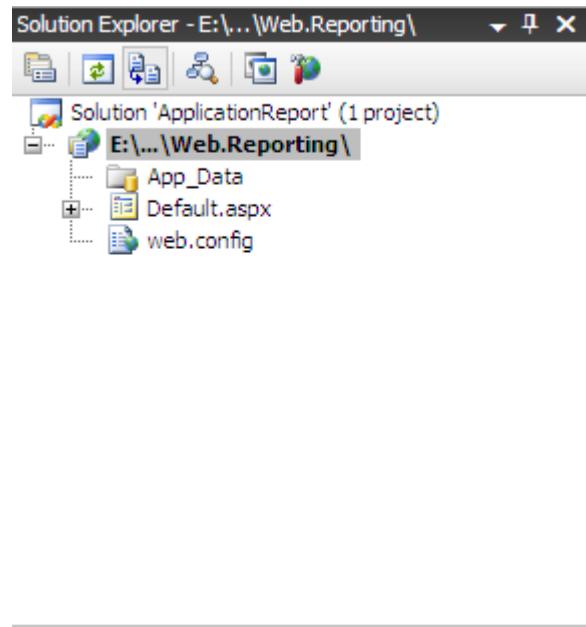
1. Efisiensi pemrosesan data, engine report yang dibangun dengan reportviewer dapat menampilkan beberapa operasi seperti filter, sort, group dan agregasi.
2. Mendukung bermacam bentuk dalam penampilan data. Kita dapat menampilkan data dalam bentuk list, tabel, grafik dan matriks (atau bisa disebut crosstab).
3. Report visual yang menarik. Kita dapat menentukan font, warna, border style, gambar background dan lain-lain untuk membuat tampilan report lebih menarik.
4. Mendukung format tertentu. Kita dapat menambahkan expresi-expresi kedalam report untuk mengubah model tampilan secara dinamis berdasarkan nilai-nilai dari data.
5. Mendukung atau menyediakan pencetakan dan preview pencetakan.
6. Mendukung atau menyediakan export ke Excel dan PDF.

Control ReportViewer ini dapat memproses dan me-render report sendiri menggunakan sebuah build-in engine ('local mode') atau dapat juga menampilkan report yang di proses dan dirender pada reportserver ('remote mode').

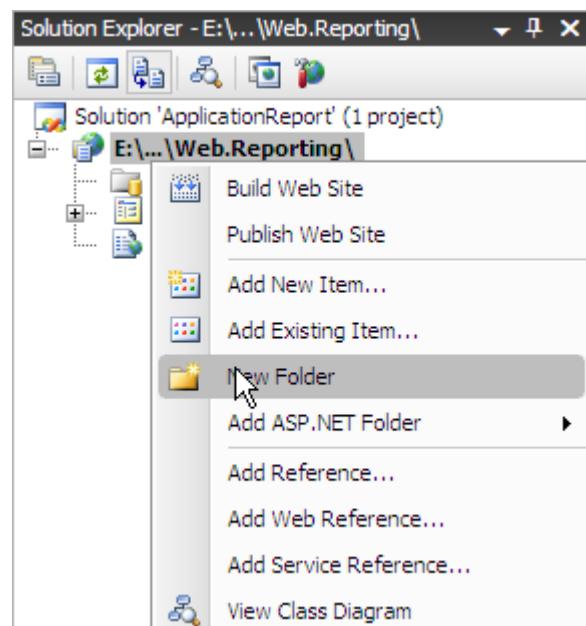
Bagaimana cara untuk membuat report mulai dari menambahkan report designer kedalam suatu project web site.

Berikut langkah-langkahnya :

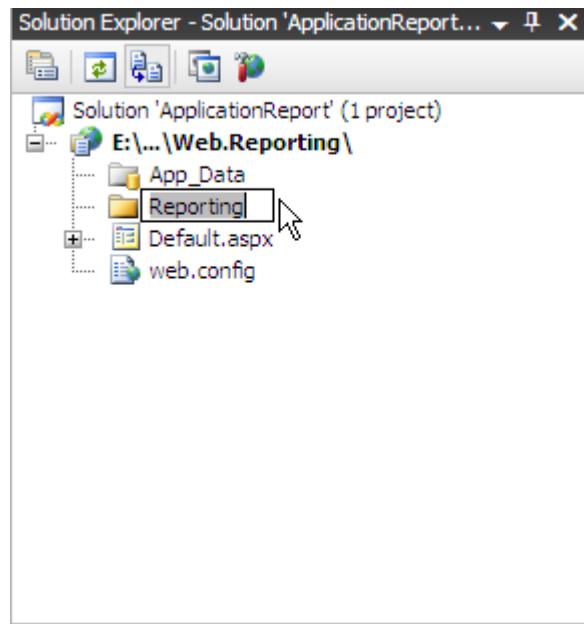
1. Pastikan dalam solution yang kita buat terdapat project web site



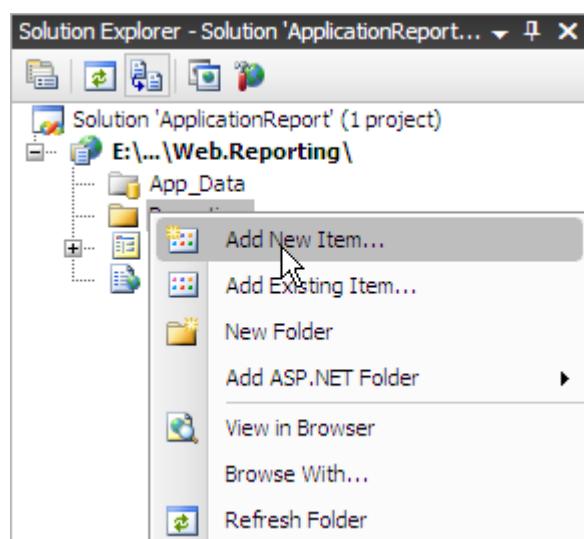
2. Di dalam project web site kita bisa tambahkan folder bernama Reporting, dengan cara click kanan pada project web → lalu pilih New Folder



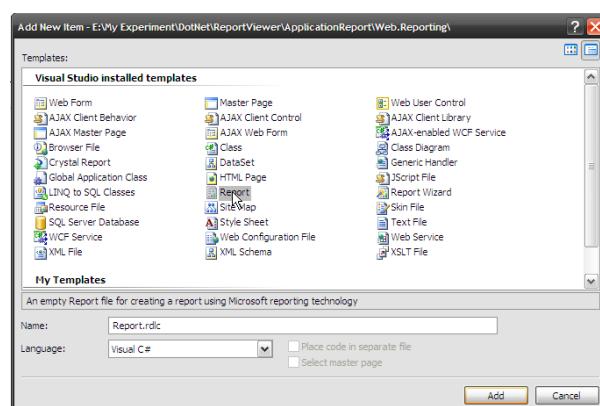
3. Pada project web site, folder tersebut kita berinama Reporting



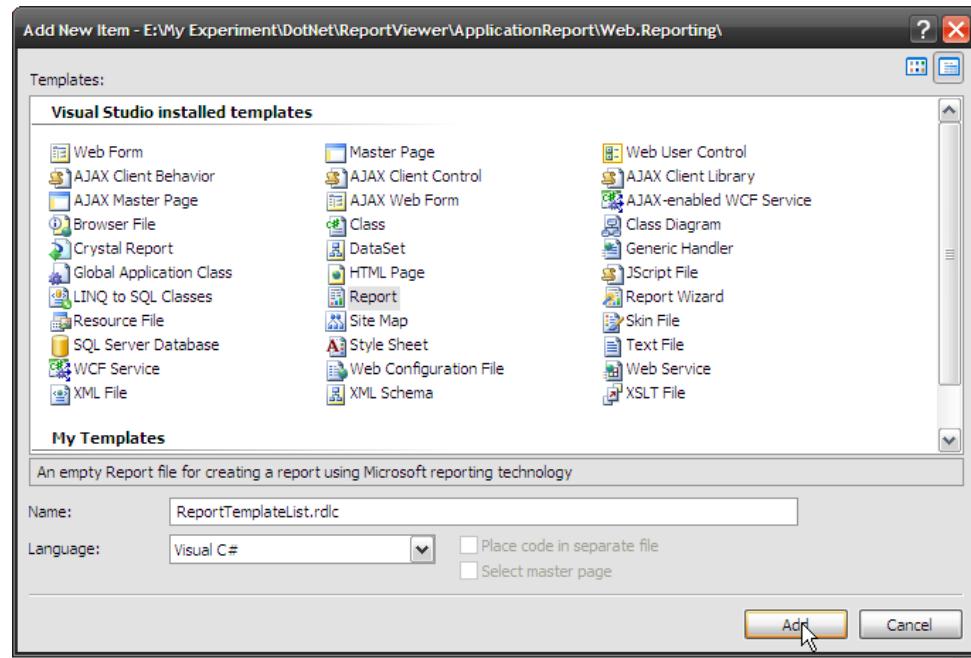
4. Tambahkan pada folder Reporting page report designer (.rdlc), caranya click kanan pada folder Reporting → Pilih Add New Item..



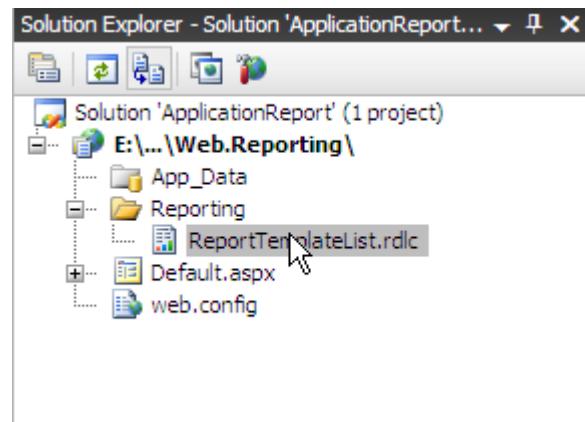
Lalu akan muncul dialog seperti dibawah ini, pilih Report.



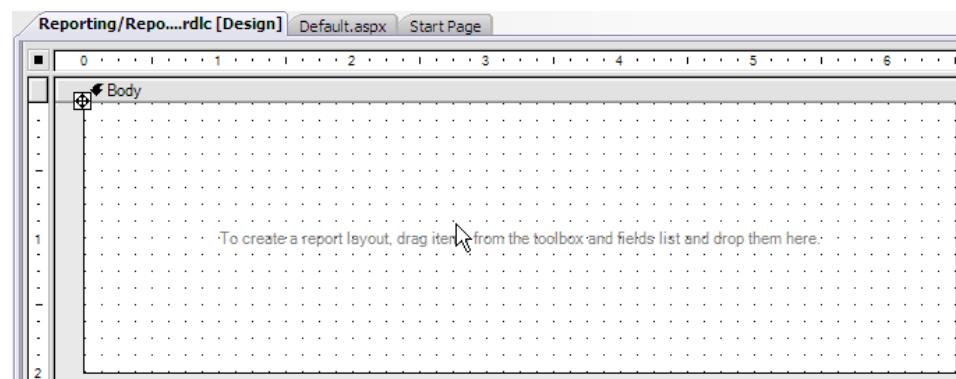
5. Beri nama report kita menjadi ReportTemplateList.rdlc → lalu click Add



Maka akan terlihat dalam Solution kita seperti berikut :

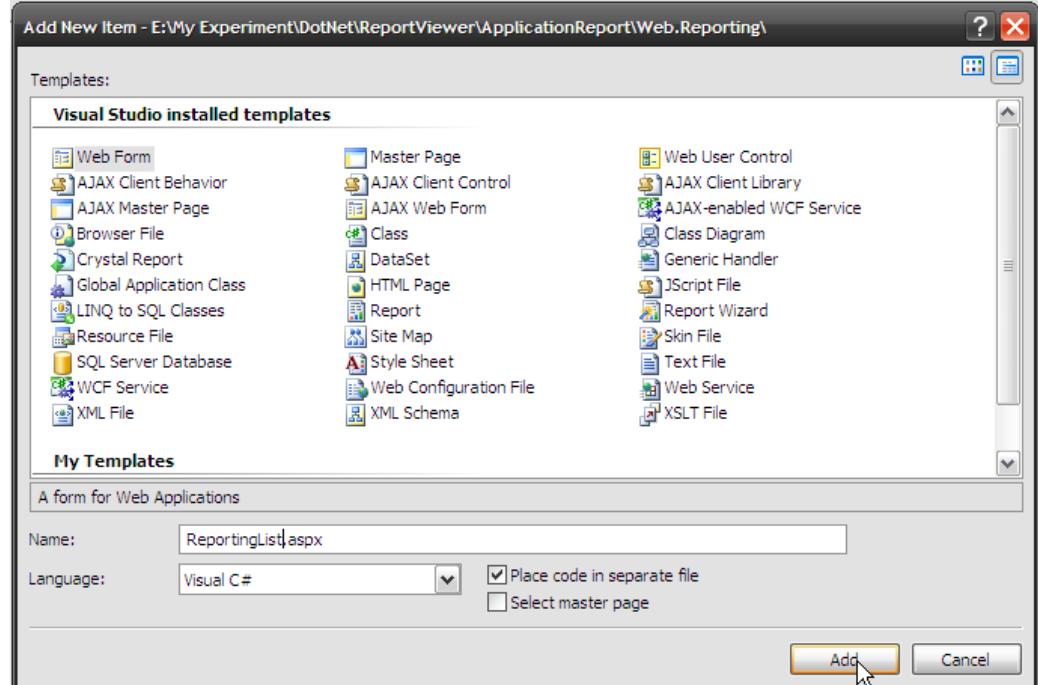


Dan kita bisa melihat area designer untuk report seperti berikut :

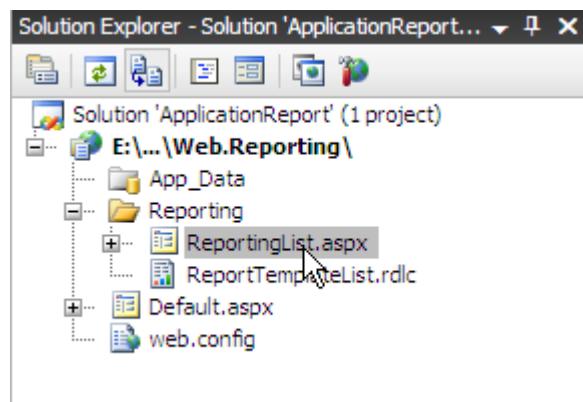


- Setelah menambahkan report designer langkah selanjutnya agar hasil designer dapat di tampilkan kita tambahkan pake aspx pada folder Reporting. Caranya click

kanan pada folder Reporting → pilih Add New Item → Pilih Web Form → beri nama page tersebut dengan nama ReportingList.aspx → lalu click Add



Maka akan terlihat pada Solution kita seperti berikut :



Berikut bisa dilihat code dari page aspx :

```
ReportingList.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportingList.aspx.cs"
Inherits="Reporting_ReportingList" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
```

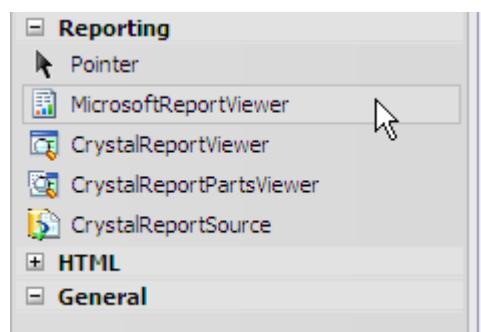
```

<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>

</div>
</form>
</body>
</html>

```

7. Setelah page aspx di tambahkan, tambahkan control MicrosoftReportViewer pada tollbox



drag-and-drop control tersebut kedalam page ReportingList.aspx, sehingga akan terlihat code seperti berikut :

```

ReportingList.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportingList.aspx.cs"
Inherits="Reporting_ReportingList" %>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms",
Version=9.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
    Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <rsweb:ReportViewer ID="ReportViewer1" runat="server">
                </rsweb:ReportViewer>
            </div>
        </form>
    </body>

```

```
</html>
```

8. Setelah proses diatas selesai, maka untuk tahap pertama pembuatan report sudah selesai dilakukan tanpa ada data yang bisa ditampilkan.

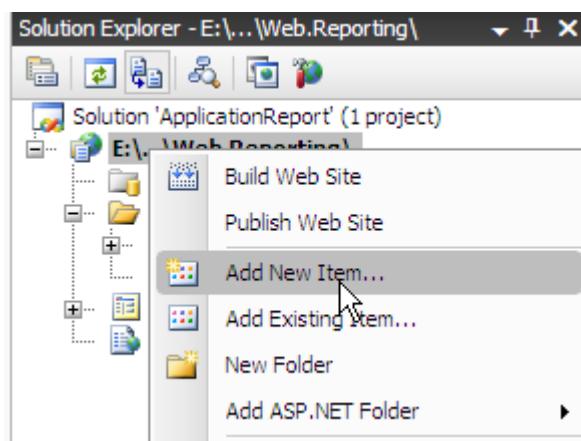
Untuk lebih detail seperti apa cara memasukan data dan menampilkannya, dan control apa saja yang bisa dipake di reportviewer designer, berikut penjelasannya :

Report Data Source

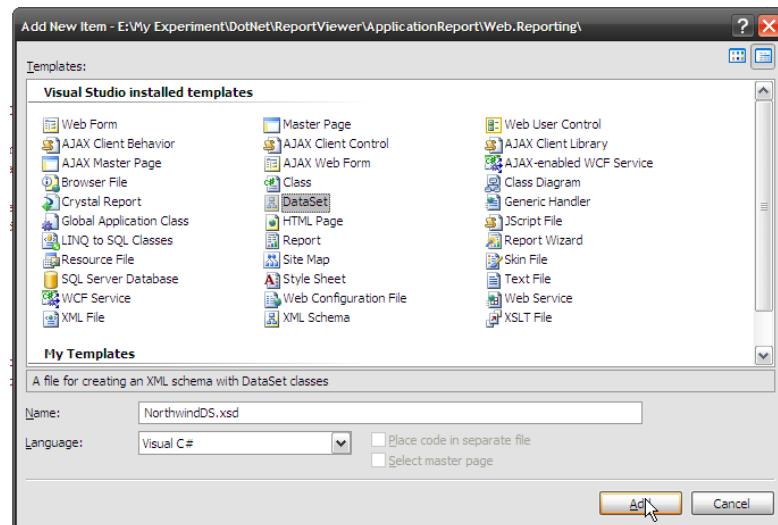
Untuk menampilkan data pada suatu report, yang dibutuhkan oleh report tersebut adalah sumber data yang akan ditampilkan. Pada report viewer sumber data (data source) bisa di suplay dari dua jenis, 1. Object 2. DataTable. Pada pembahasan sekarang kita akan membahas bagaimana menambahkan data source dari data table atau menggunakan DataSet.

Berikut cara menambahkan DataSet yang nantinya akan digunakan sebagai data source bagi report :

1. Pada Solution ApplicationReport terdapat project Web.Reporting → Click kanan pada project web site tersebut → lalu pilih Add New Item..

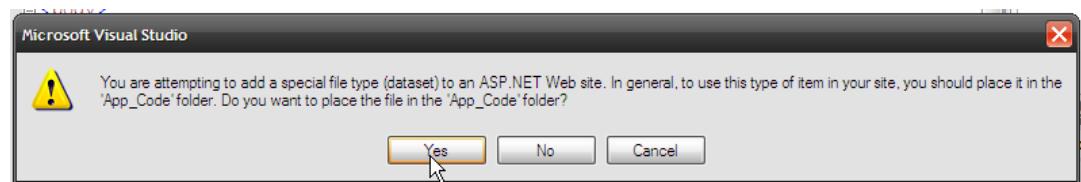


Kemudian akan muncul dialog seperti berikut :

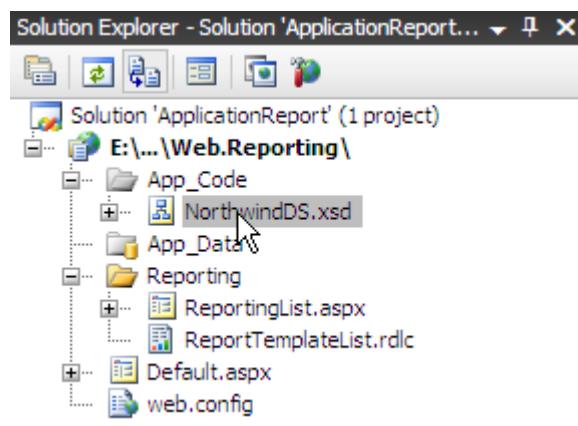


Pilih DataSet kemudian beri nama dataset tersebut dengan nama NorthwindDS.xsd, kemudian click Add.

Ketika Add di Click maka akan muncul dialog seperti dibawah ini, click Yes



Setelah di click Yes, maka project web akan secara otomatis membuat folder App_Code dimana folder tersebut akan memuat DataSet yang sudah ditambahkan tadi.

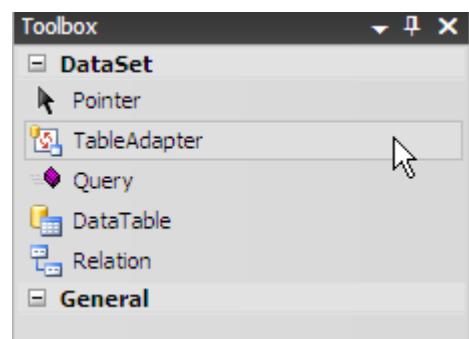


Dan akan terlihat layout design dari dataset tersebut :

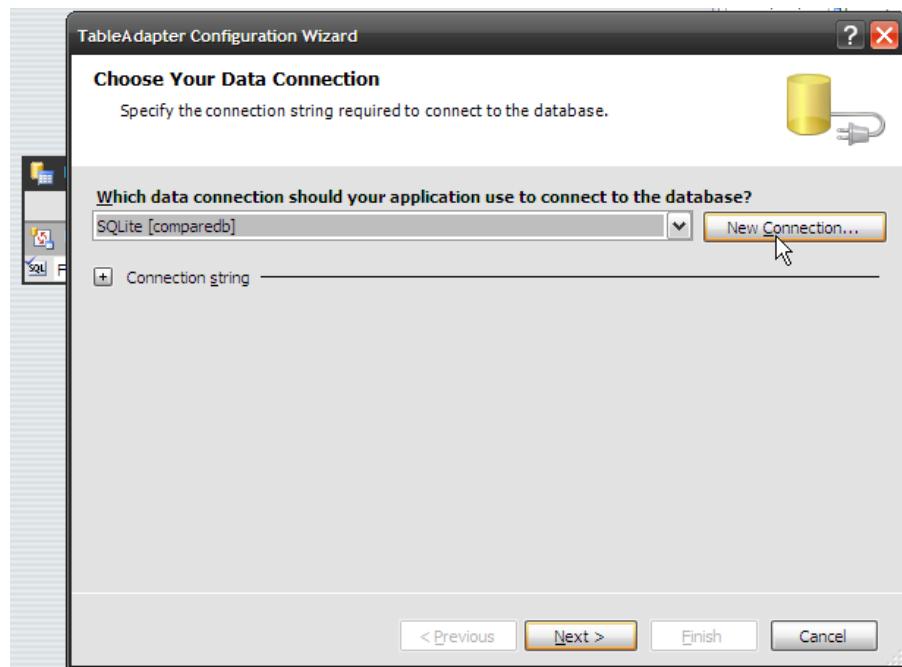


Layout diatas akan digunakan untuk membuat design datatable yang akan digunakan sebagai sumber data bagi report.

2. Dari toolbox tambahkan control TableAdapter kedalam layout design dataset

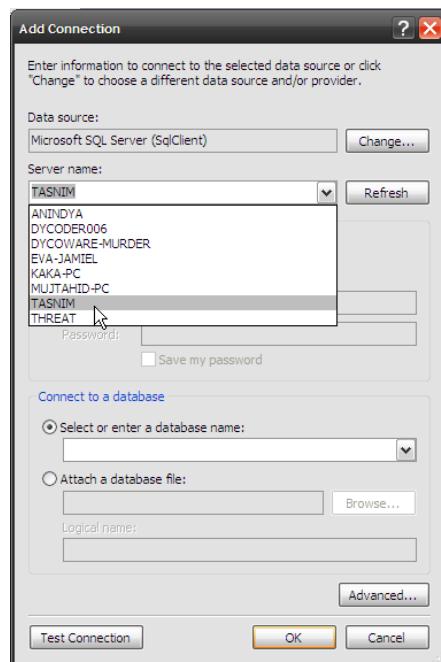


drag-and-drop TableAdapter sehingga terlihat tampilan seperti berikut :

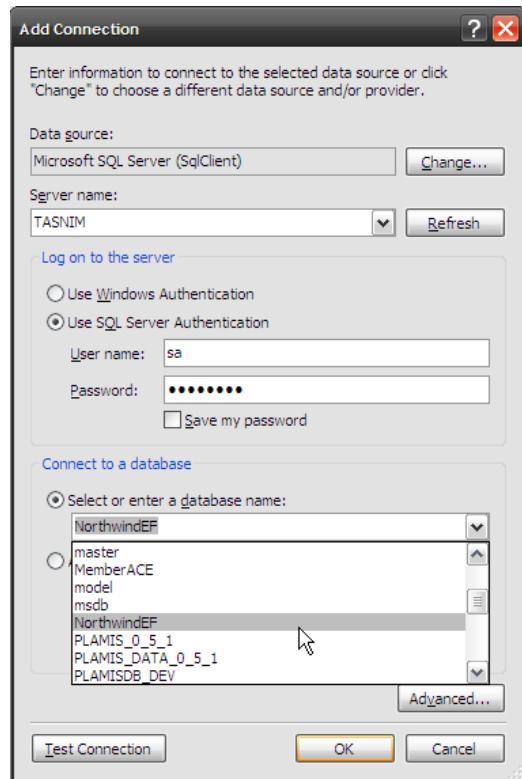


3. Dari dialog diatas pilih New Connection jika koneksi belum dibuat atau pilih dari dropdown di samping connection yang sudah dibuat.

Jika koneksi belum dibuat pilih New Connection maka akan muncul dialog seperti berikut :

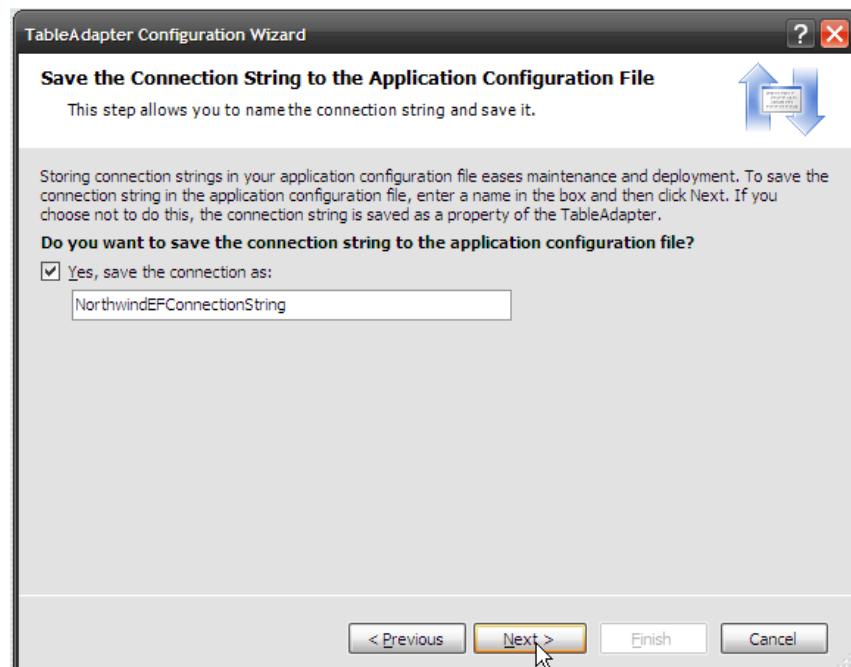


Pilih server name, dimana dalam server tersebut ada database yang akan digunakan sebagai data source. Tambahkan SQL Server Authentication jika diperlukan, pastikan koneksi berhasil dengan meng-click Button Test Connection, setelah itu pilih database yang akan digunakan seperti berikut :



Pilih database Northwind untuk digunakan dalam contoh ini. Setelah itu click OK

4. Click Next maka akan muncul dialog seperti berikut :

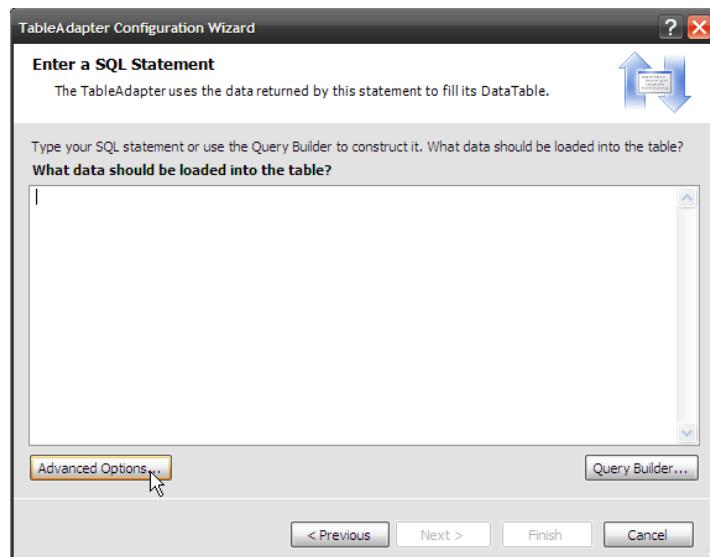


Dialog tersebut menyatakan bahwa akan ditambahkan connection string pada file Web.config jika dibutuhkan maka click Next, anda pun bisa merubah nama dari Connection Stringnya.

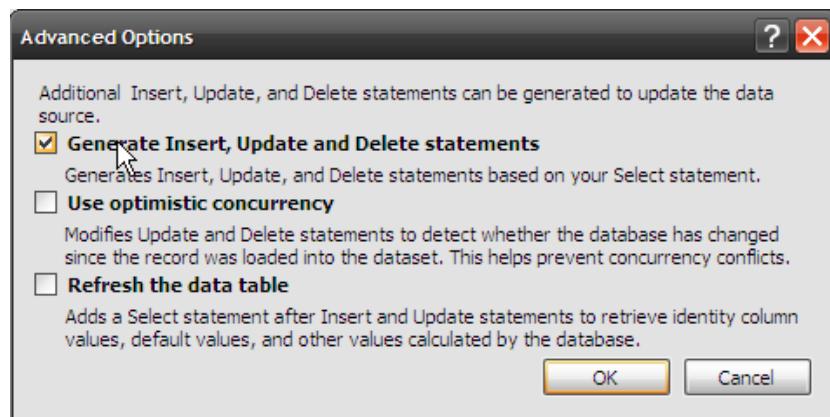
Code nya seperti berikut :

```
Web.config
<connectionStrings>
<add name="NorthwindEFConnectionString" connectionString="Data
Source=tasnim;Initial Catalog=NorthwindEF;Persist Security
Info=True;User ID=sa;Password=nutasnim"
providerName="System.Data.SqlClient" />
</connectionStrings>
```

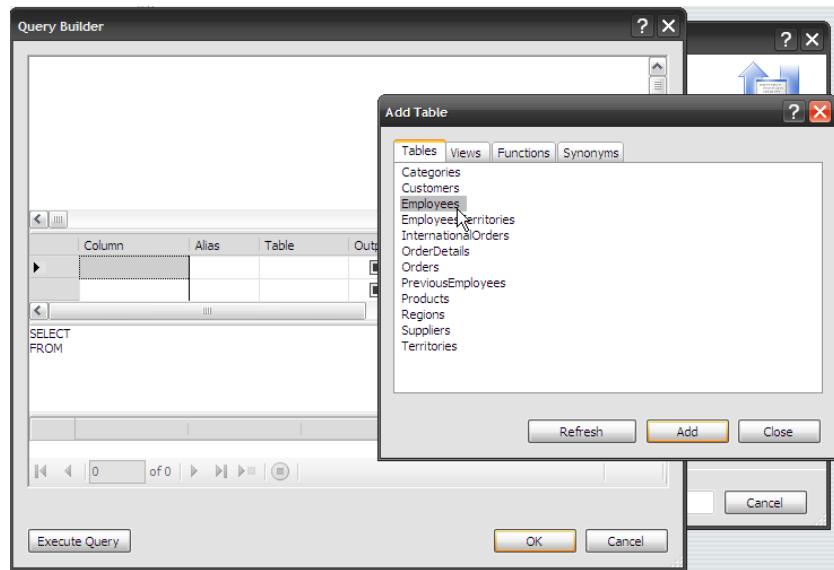
5. Setelah click Next secara default terpilih Use SQL Statement → selanjutnya click Next



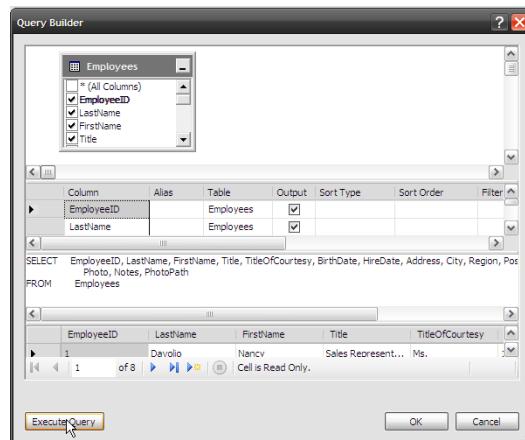
6. Pilih button Advanced Option untuk men-uncheck seperti dialog dibawah, setelah itu pilih OK.



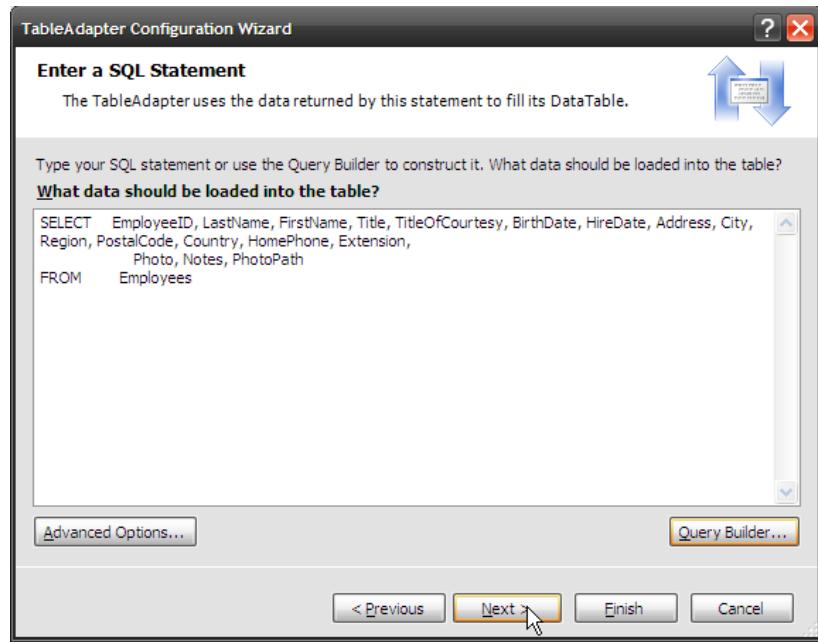
7. Click button Query Builder maka akan muncul dialog seperti dibawah :



Pilih table satu atau lebih untuk digunakan dalam query, pada kasus sekarang coba pilih satu table Employees, setelah itu click button Add → lalu select all data dari table Employees



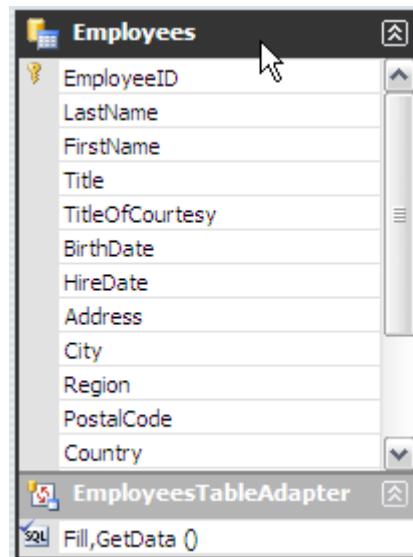
8. Kemudian click OK → click Next untuk melanjutkan proses



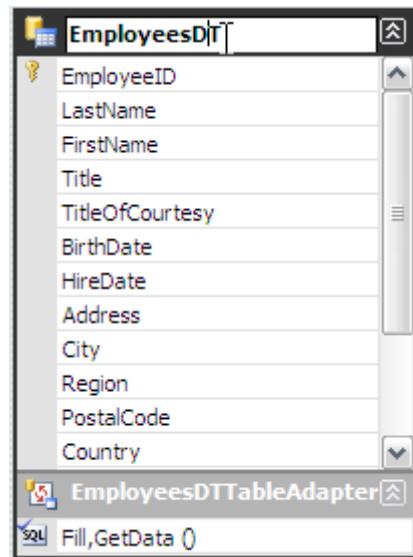
9. Setelah button Next di click maka akan terlihat calon dari method yang di generate oleh datatable



10. Kemudian click Next → Finish



11. Rubah nama data table Employees menjadi EmployeesDT, untuk menandakan bahwa EmployeesDT itu datatable, dengan cara double click di nama data tablenya.



12. Kemudian click Save All di toolbar Visual Studio 2008



13. Proses pembuatan datatabel yang akan digunakan sebagai data source untuk report telah selesai.

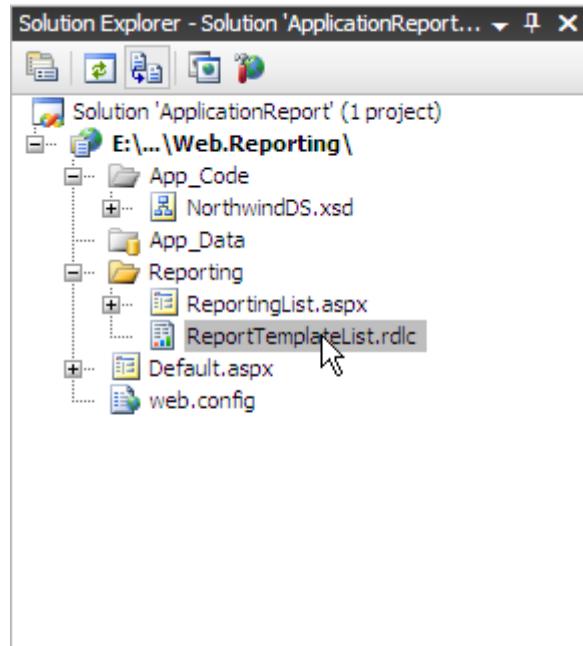
Bagaimana cara menggunakan datasource untuk di suply ke report, akan dibahas lebih lanjut dengan menggunakan control-control pada reportviewer designer.

List

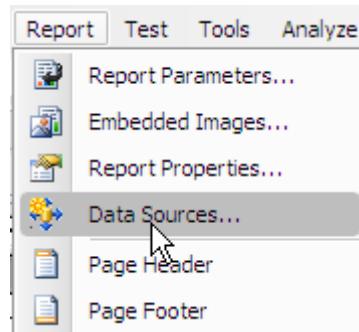
List salah satu control yang ada di ReportViewer yang digunakan untuk menampilkan data berbentuk list. Penggunaan control list sangat mudah hanya dengan men-drag-and-drop control tersebut dan men-drag juga data yang akan ditampilkan.

Bagaimana implementasi penggunaan control List ?, kita lihat cara pakainya seperti berikut :

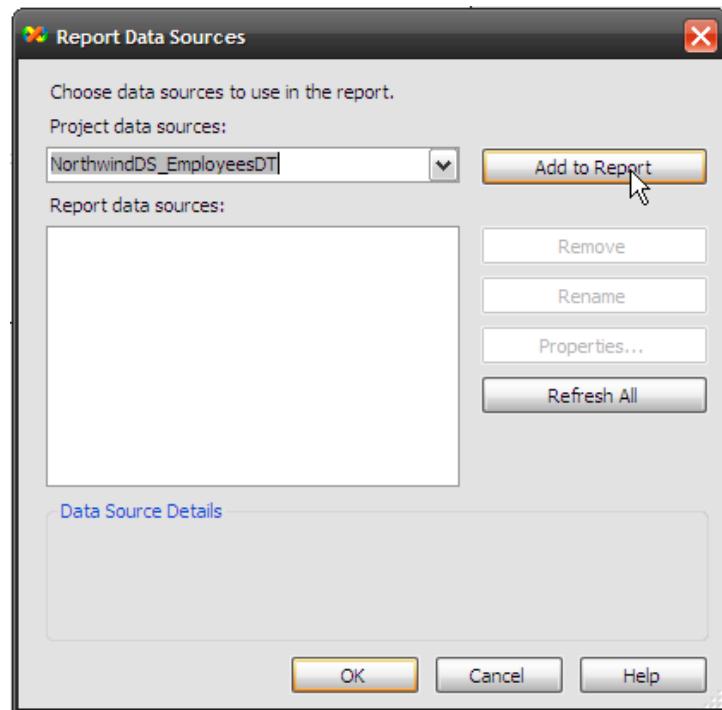
1. Dari project web double click file **ReportTemplateList.rdlc** sehingga menampilkan layout designnya.



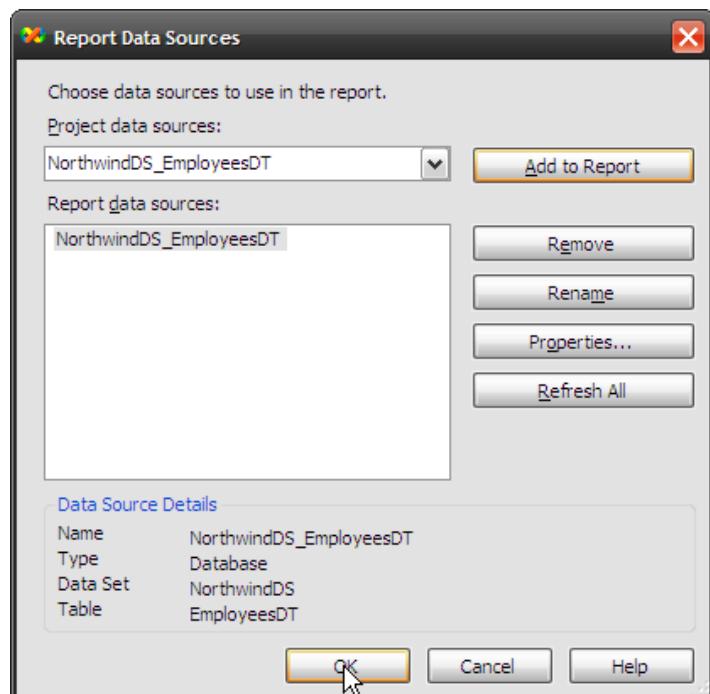
2. Sebelum men-drag control list, terlebih dahulu tambahkan datasource kedalam design report dengan cara → click report design maka akan muncul menu Report → Pilih Data Sources..



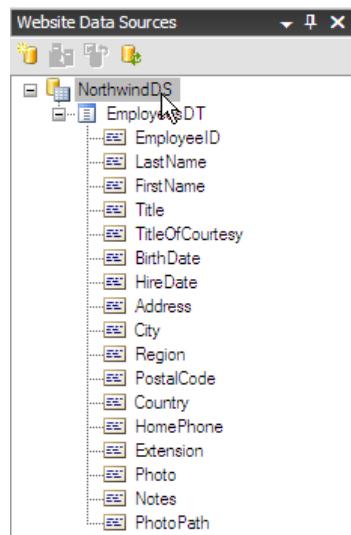
Setelah data source dipilih maka akan muncul dialog seperti dibawah :



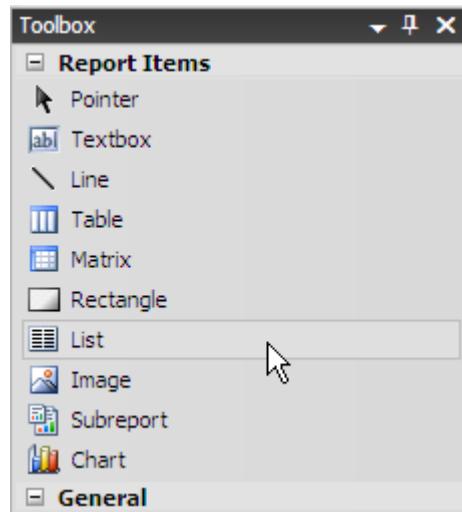
3. Click button Add to Report → click button OK



- Setelah itu akan muncul Website Data Sources, data sorce ini yang akan digunakan sebagai data source untuk report.

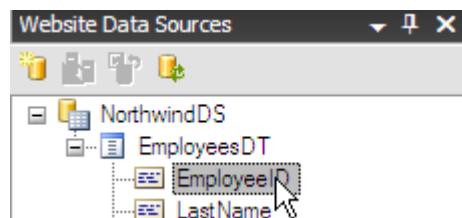


- Kemudian kita dapat memulai menambahkan control list yang terdapat pada toolbox

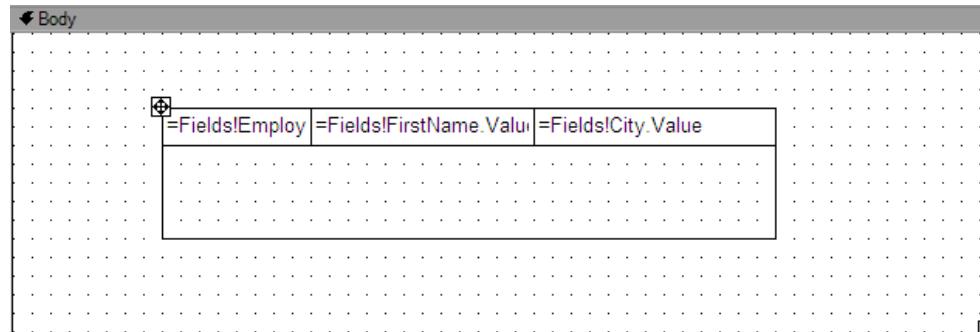


Drag control list kedalam layout designer

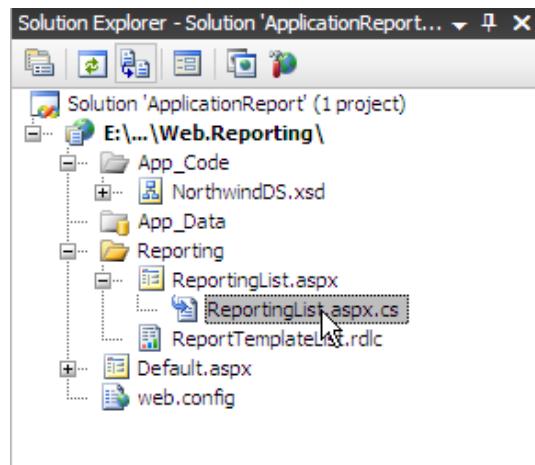
- Drag EmployeesID yang terdapat pada Website Data Sources ke dalam List



Selain EmployeeID, drag juga FirstName, City. Tampilan pada layout akan terlihat seperti berikut :



7. Data yang sudah di drag kedalam List tidak dapat dilihat, apakah data itu bakal muncul atau tidak, proses selanjutnya buka file ReportingList.aspx.cs dan tambahkan code seperti di bawah :



```
ReportingList.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportingList.aspx.cs"
Inherits="Reporting_ReportingList" %>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms",
Version=9.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
    Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
```

```

<form id="form1" runat="server">
<div>
    <rsweb:ReportViewer ID="ReportViewer1" runat="server">
        </rsweb:ReportViewer>
    </div>
</form>
</body>
</html>

```

ReportingList.aspx.cs

```

using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

//add Namespace
using Microsoft.Reporting.WebForms;

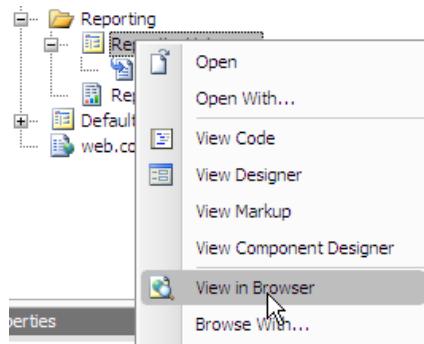
public partial class Reporting_ReportingList : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            // path of rdlc file
            ReportViewer1.LocalReport.ReportPath =
"Reporting\\ReportTemplateList.rdlc";

            ReportViewer1.ProcessingMode = ProcessingMode.Local;
            ReportViewer1.LocalReport.DataSources.Clear();

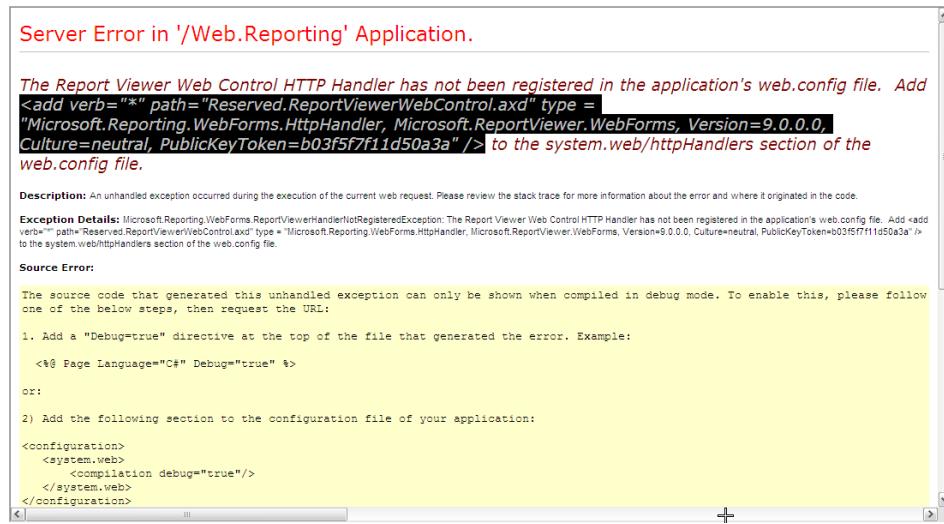
            // adding data source to reportviewer
            ReportViewer1.LocalReport.DataSources.Add(new
ReportDataSource("NorthwindDS_EmployeesDT", new
NorthwindDSTableAdapters.EmployeesDTTableAdapter().GetData()));
            ReportViewer1.LocalReport.Refresh();
        }
    }
}

```

Setelah itu click kanan pada page ReportingList.aspx lalu view in browser



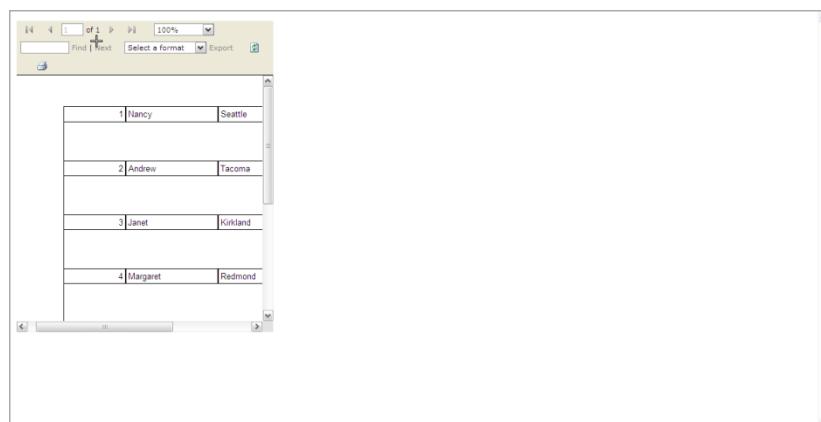
Maka akan muncul tampilan error seperti di bawah :



Copy-kan tanda yang di drag ke web.config dibawah tag berikut :

```
<httpHandlers>
  <remove verb="*" path="*.asmx"/>
  <add verb="*" path="Reserved.ReportViewerWebControl.axd" type="Microsoft.Reporting.WebForms.HttpHandler, Microsoft.ReportViewer.WebForms, Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a" />
  <add verb="*" path="*.asmx" validate="false" />
  <add verb="*" path="*_AppService.axd" validate="false" />
  <add verb="GET,HEAD" path="ScriptResource.axd" validate="false" />
</httpHandlers>
```

Setelah itu Save All, kemudian page error tadi di refresh.



Data telah berhasil muncul, tinggal kita bermain di properti Microsoft ReportViewer agar tampilan bisa dilihat lebih enak

8. Tambahkan pada properti seperti berikut

```
ReportingList.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportingList.aspx.cs"
Inherits="Reporting_ReportingList" %>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms",
Version=9.0.0.0, Culture=neutral,
PublicKeyToken=b03f5f7f11d50a3a"
    Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb"
%>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <rsweb:ReportViewer ID="ReportViewer1" Height="100px"
Width="100%" ShowFindControls="False"
ShowPageNavigationControls="False" runat="server"
ExportContentDisposition="AlwaysInline">
                </rsweb:ReportViewer>
            </div>
        </form>
    </body>
</html>
```

Maka tampilan akan terlihat seperti dibawah :

The screenshot shows a Microsoft ReportViewer control with a toolbar at the top containing zoom controls, a 'Select a format' dropdown, and 'Export' buttons. Below the toolbar is a scrollable grid displaying 9 rows of data. Each row contains three columns: Employee ID (1-9), Employee Name (Nancy, Andrew, Janet, Margaret, Steven, Michael, Laura, Anne), and City (Seattle, Tacoma, Kirkland, Redmond, London, London, Seattle, London). A vertical scrollbar is visible on the right side of the grid.

1	Nancy	Seattle
2	Andrew	Tacoma
3	Janet	Kirkland
4	Margaret	Redmond
5	Steven	London
6	Michael	London
7	Laura	Seattle
8	Anne	London

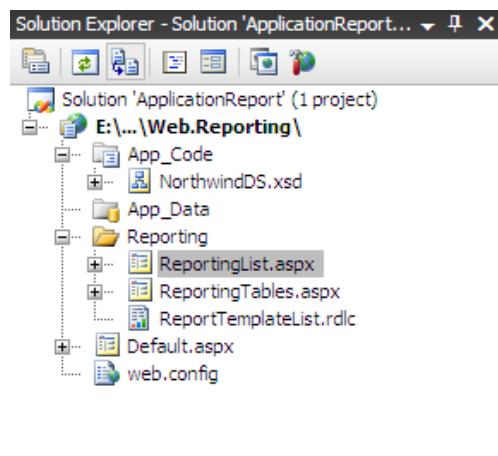
9. Proses menampilkan data menggunakan List sudah selesai.

Tables

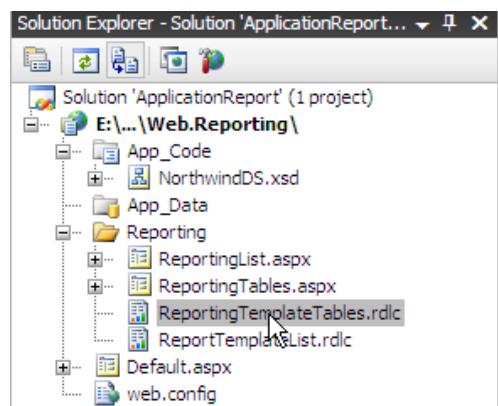
Tables adalah control yang ada di report viewer designer yang juga berfungsi untuk menampilkan data seperti halnya List, namun di table sudah lebih bagus lagi featernya dikarenakan ada header, detail dan footernya.

Seperti apa penggunaan Table, berikut contoh latihannya :

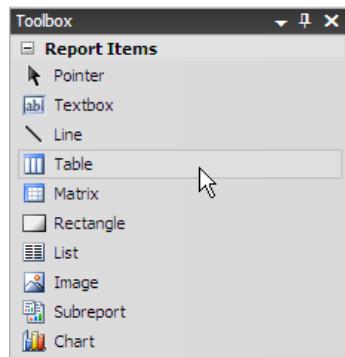
1. Buat page dengan nama ReportingTables.aspx pada Folder Reporting



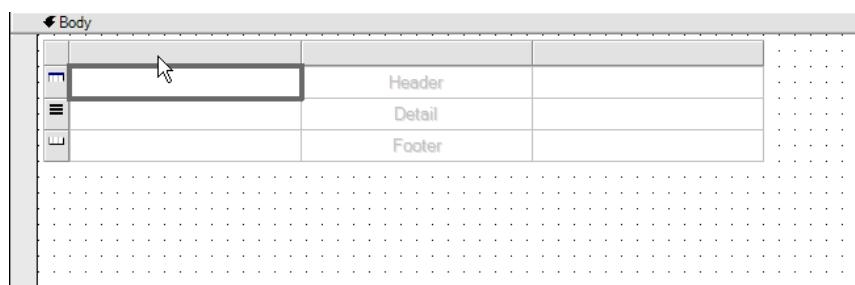
2. Tambahkan file ReportingTemplateTables.rdlc pada folder Reporting



3. Data yang akan ditampilkan menggunakan datatable EmployeesDT, tambahkan datasorce kedalam file ReportingTemplateTables.rdlc
4. Drag control tables yang ada di toolbox kedalam layout designer



5. Tampilan akan terlihat seperti dibawah



Drag data yang akan ditampilkan kedalam Tables dari Website Data Sources

Employee ID	First Name	City
=Fields!EmployeeID.Value	=Fields!FirstName.Value	=Fields!City.Value
Footer		

6. Tambahkan pada File ReportingTables.aspx control MicrosoftReportViewer untuk menampilkan report dari file .rdlc

```

ReportingTables.aspx
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportingTables.aspx.cs"
Inherits="Reporting_ReportingTables" %>

<%@ Register Assembly="Microsoft.ReportViewer.WebForms",
Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <rsweb:ReportViewer ID="ReportViewer1" Height="1000px"
Width="100%" ShowFindControls="False"
ShowPageNavigationControls="False" runat="server"
ExportContentDisposition="AlwaysInline">
            </rsweb:ReportViewer>
        </div>
    </form>
</body>
</html>

```

```
</form>
</body>
</html>
```

ReportingTables.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

//add Namespace
using Microsoft.Reporting.WebForms;

public partial class Reporting_ReportTables : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            // path of rdlc file
            ReportViewer1.LocalReport.ReportPath =
"Reporting\\ReportingTemplateTables.rdlc";

            ReportViewer1.ProcessingMode = ProcessingMode.Local;
            ReportViewer1.LocalReport.DataSources.Clear();

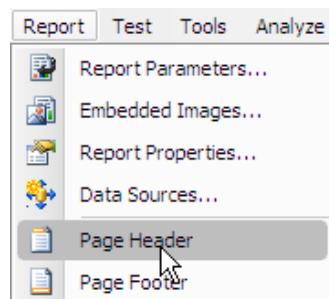
            // adding data source to reportviewer
            ReportViewer1.LocalReport.DataSources.Add(new
ReportDataSource("NorthwindDS_EmployeesDT", new
NorthwindDSTableAdapters.EmployeesDTTableAdapter().GetData()));
            ReportViewer1.LocalReport.Refresh();
        }
    }
}
```

7. Click kanan pada page ReportingTables.aspx maka tampilan aka terlihat seperti di bawah :

The screenshot shows a report viewer window with a table containing the following data:

Employee ID	First Name	City
1	Nancy	Seattle
2	Andrew	Tacoma
3	Janet	Kirkland
4	Margaret	Redmond
5	Steven	London
6	Michael	London
7	Laura	Seattle
8	Anne	London

8. Pada report tambahkan Header untuk judul Report.

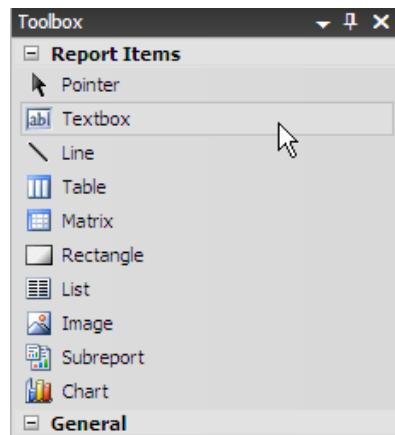


Tampilan akan terlihat seperti dibawah

The screenshot shows the report design surface with the following structure:

- Page Header:** Contains a single row with three columns labeled "Employee ID", "First Name", and "City".
- Body:** Contains a single row with three columns. The first column contains the expression "=Fields!EmployeeID.Value", the second column contains "=Fields!FirstName.Value", and the third column contains "=Fields!City.Value". Below this row is a label "Footer".

9. Tambahkan pada Header control TextBox pada toolbox dengan cara men-drag control tersebut untuk menuliskan Label



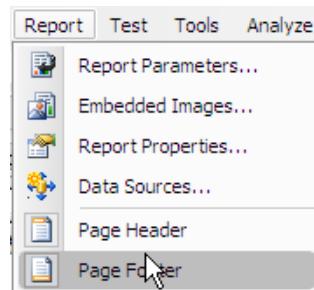
Ketik pada TextBox Data Employees sehingga menjadi :

Employee ID	First Name	City
=Fields!EmployeeID.Value	=Fields!FirstName.Value	=Fields!City.Value
	Footer	

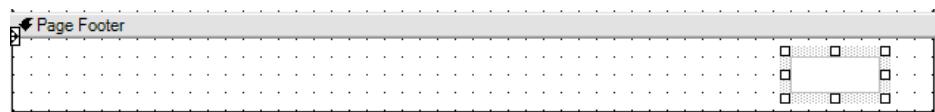
10. Refresh browser maka tampilan akan terlihat ada Header pada report

Employee ID	First Name	City
1	Nancy	Seattle
2	Andrew	Tacoma
3	Janet	Kirkland
4	Margaret	Redmond
5	Steven	London
6	Michael	London
7	Laura	Seattle
8	Anne	London

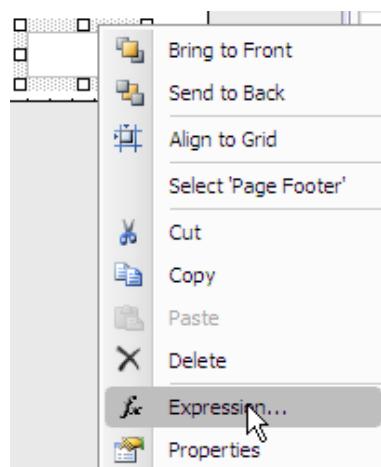
11. Untuk mengetahui jumlah halaman untuk report tambahkan Footer pada menu Report



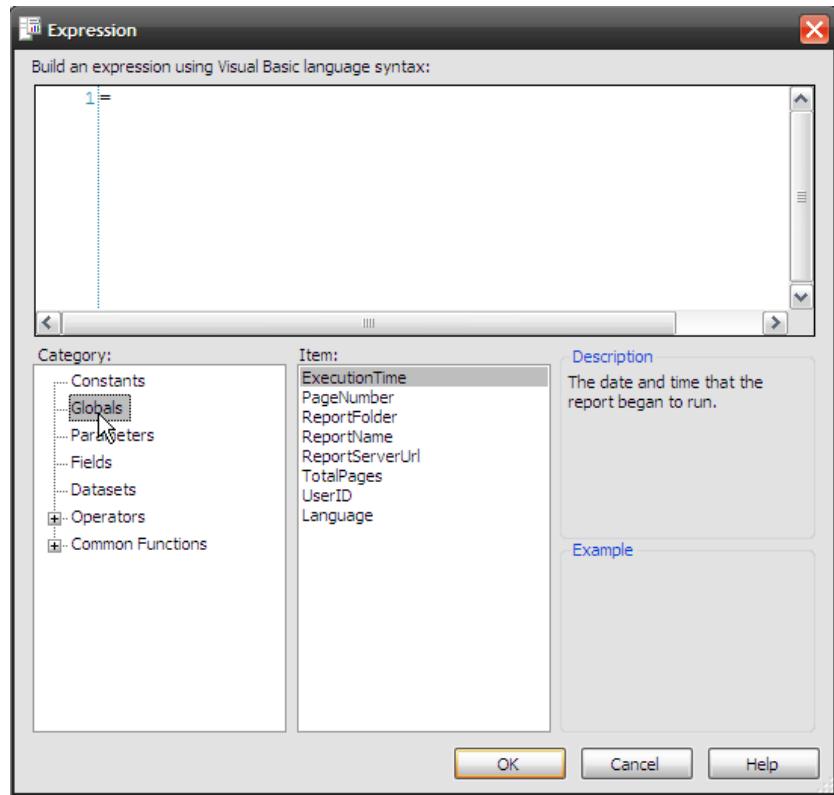
Setelah di pilih, pada Footer report tambahkan TextBox dari tool box



12. Arahkan cursor ke TextBox → click kanan pilih Expression

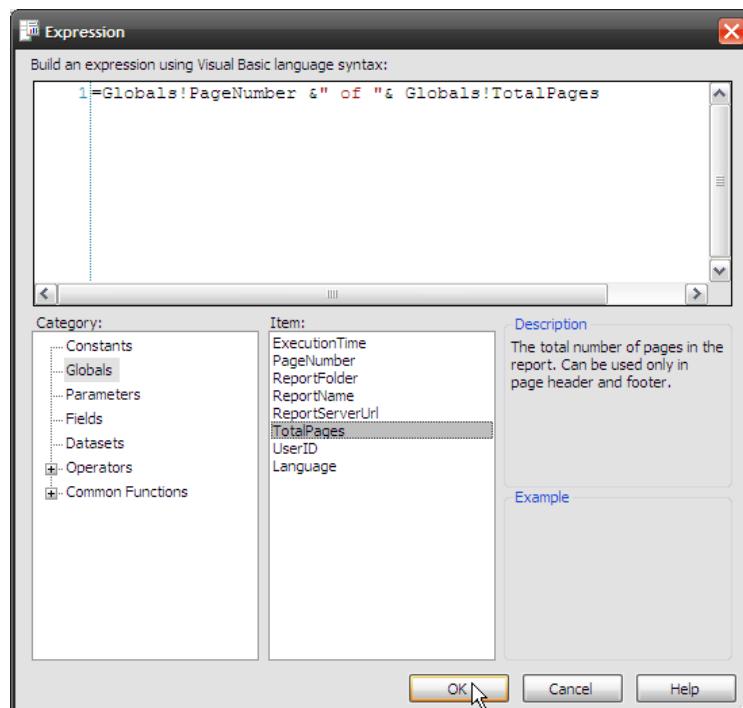


13. Setelah pilih Expression akan muncul diaolog seperti di bawah

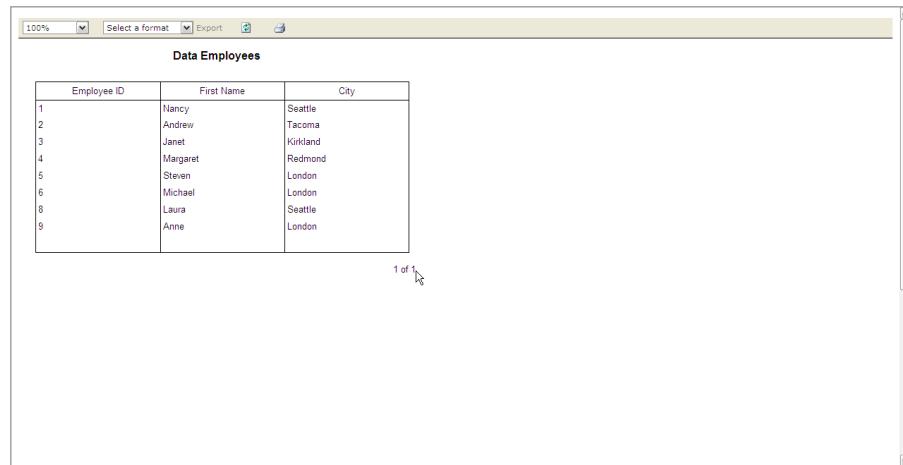


Pilih Category Global, di list Item pilih PageNumber dengan mengarahkan cursor ke area expression, lalu tambahkan juga TotalPages.

14. Gambar dibawah memperlihatkan pengisian Expression



15. click OK kemudian save perubahan design report, sekarang refresh browser anda! Maka akan terlihat tampilan seperti dibawah ini

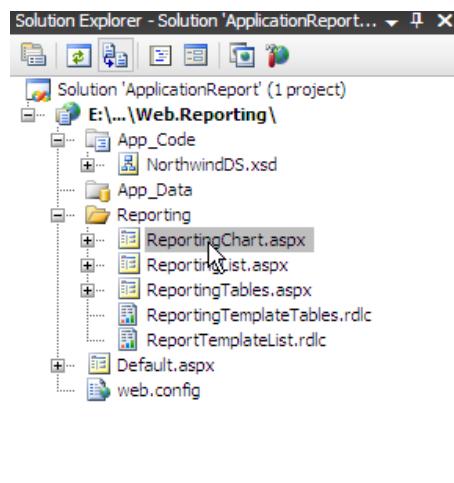


Sekarang sudah terlihat report yang mengandung Header untuk Title dari report dan Footer untuk halaman dari report.

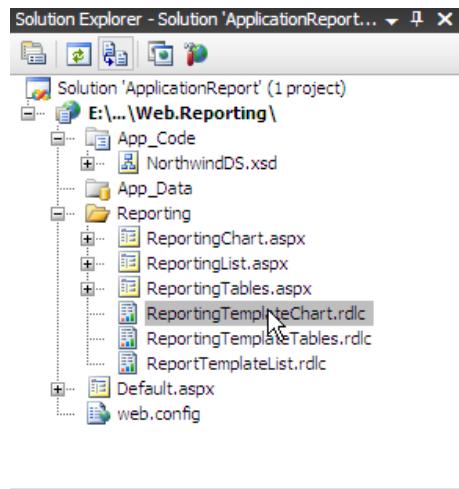
Chart

Chart adalah suatu control yang ada di reportviewer, yang kegunaannya sangat membantu dalam membuat laporan berbaskan grafik. Untuk menggunakan control chart di reportviewer sangat mudah, hanya dengan drag-and-drop lalu isi chart tersebut dengan datasource yang akan ditampilkan. Bagaimana proses pembuatan chart mari kita ikuti bersama langkah-langkahnya :

1. Buat page dengan nama ReportingChart.aspx pada Folder Reporting



2. Tambahkan file ReportingTemplateChart.rdlc pada folder Reporting



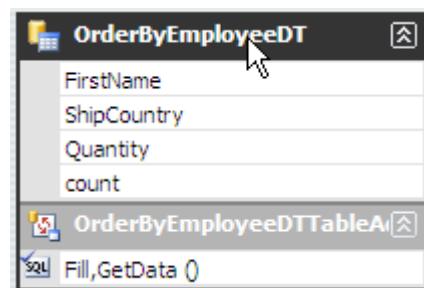
3. Untuk menampilkan Chart kita persiapkan dahulu data source-nya, untuk itu tambahkan table adapter pada dataset.
4. Drag TableAdapter pada layout dataset → click Next → Pilih UseSQLStatement → click Next → click button advanced Option dan un-check generate Update → Masukan Query dibawah :

```

SELECT Employees.FirstName, Orders.ShipCountry, OrderDetails.Quantity,
COUNT(*) AS count, Orders.OrderDate
FROM Orders INNER JOIN
Employees ON Orders.EmployeeID = Employees.EmployeeID INNER
JOIN
OrderDetails ON Orders.OrderID = OrderDetails.OrderID
GROUP BY Employees.FirstName, OrderDetails.Quantity, Orders.ShipCountry,
Orders.OrderDate
  
```

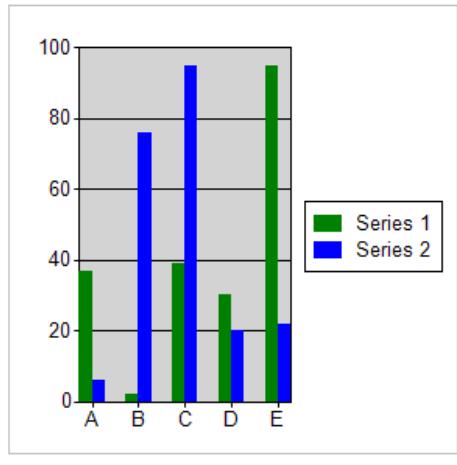
Dengan query diatas akan diketahui jumlah Employee yang melakukan Order.

5. Selanjutnya click Next sampe Finish → Beri nama data table dengan nama OrderByEmployeeDT → save all

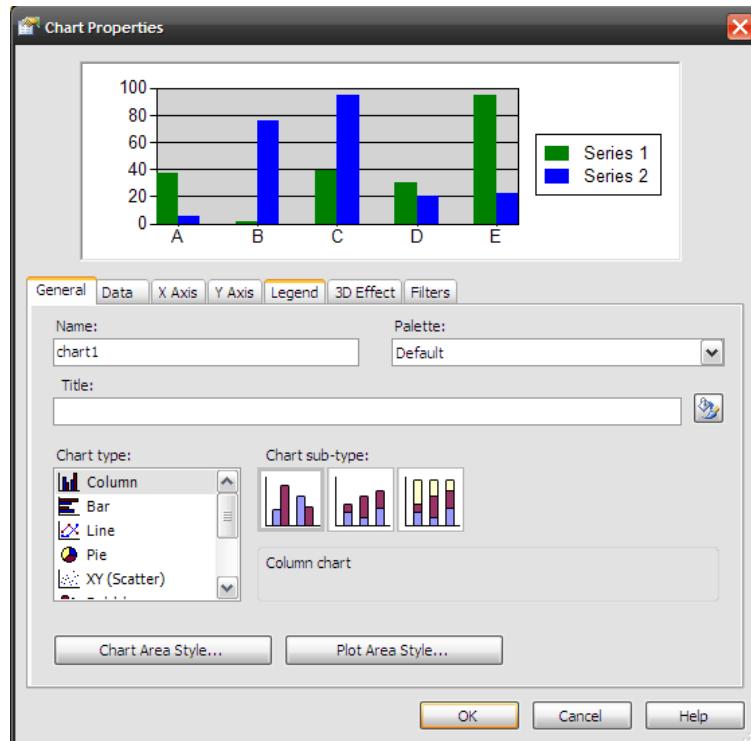


6. Buka design report viewer untuk membuat grafik

7. Tambahkan data source pada file ReportingTemplateChart.rdlc → drag control chart pada layout design

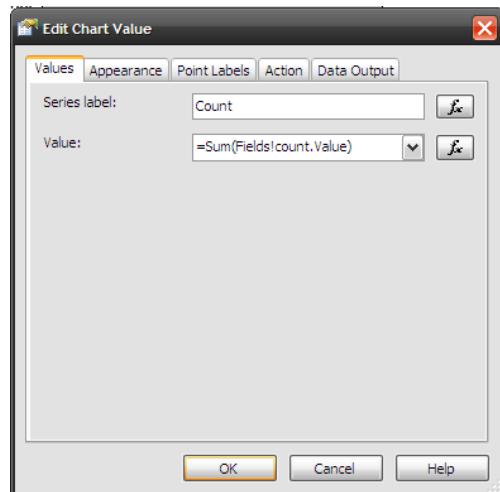


8. Click kanan pada chart maka akan muncul dialog seperti berikut :

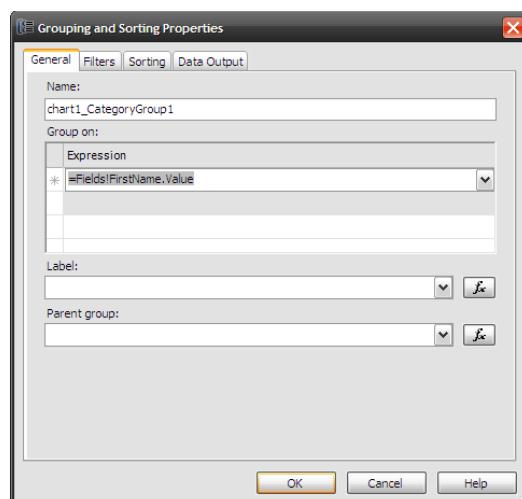


9. Isi Title dengan (Order By Employee)
10. Click tab data → pilih dataset nama → Add value pilih sum(Field!count.value) → click Ok → add category group → pilih pada exspresion Field!FirstName.Value → click OK

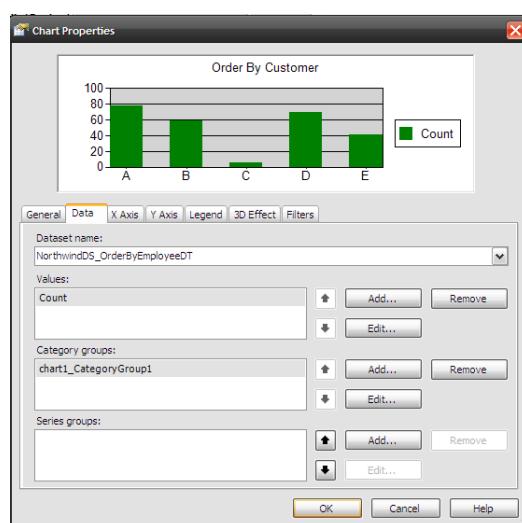
Gambar ketika add value :



Gambar ketika add category :



Gambar Tab data :



11. Tambahkan pada page ReportingChart.aspx control MicrosoftReportViewer lalu tampilkan code seperti berikut :

ReportingChart.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportingChart.aspx.cs" Inherits="Reporting_ReportingChart"
%>
<%@ Register Assembly="Microsoft.ReportViewer.WebForms,
Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <rsweb:ReportViewer ID="ReportViewer1" Height="1000px"
Width="100%" ShowFindControls="False"
ShowPageNavigationControls="False" runat="server"
ExportContentDisposition="AlwaysInline">
                </rsweb:ReportViewer>
            </div>
        </form>
    </body>
</html>
```

ReportingChart.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

//add Namespace
using Microsoft.Reporting.WebForms;

public partial class Reporting_ReportingChart : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            // path of rdlc file
            ReportViewer1.LocalReport.ReportPath =
"Reporting\\ReportingTemplateChart.rdlc";

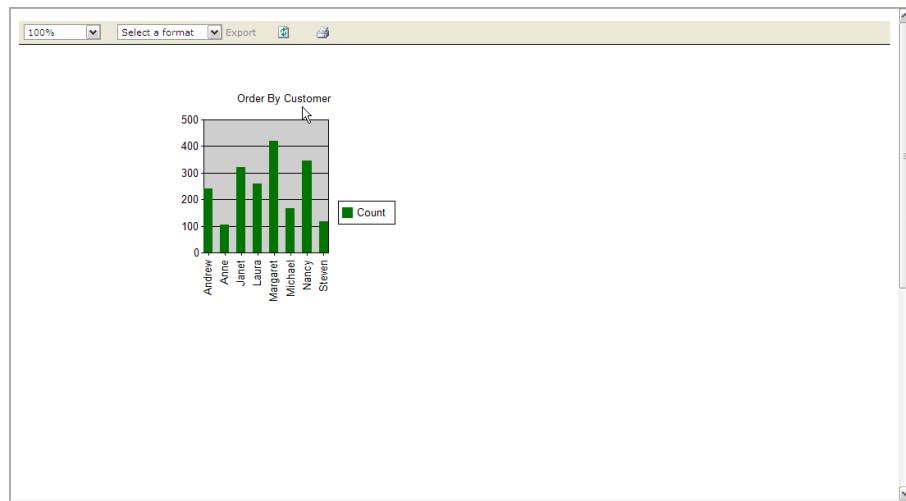
            ReportViewer1.ProcessingMode = ProcessingMode.Local;
            ReportViewer1.LocalReport.DataSources.Clear();
        }
    }
}
```

```

        // adding data source to reportviewer
        ReportViewer1.LocalReport.DataSources.Add(new
ReportDataSource("NorthwindDS_OrderByEmployeeDT", new
NorthwindDSTableAdapters.OrderByEmployeeDTTableAdapter().GetData()));
        ReportViewer1.LocalReport.Refresh();
    }
}

```

12. Pada page ReportingChart.aspx click kanan pilih view in browser maka akan terlihat tampilan seperti dibawah ini :

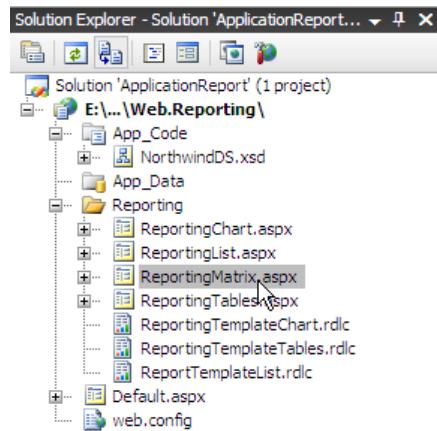


13. Bentuk chart bisa dirubah sesuai kebutuhan dengan cara click kanan pada properti chart di report viewer designernya

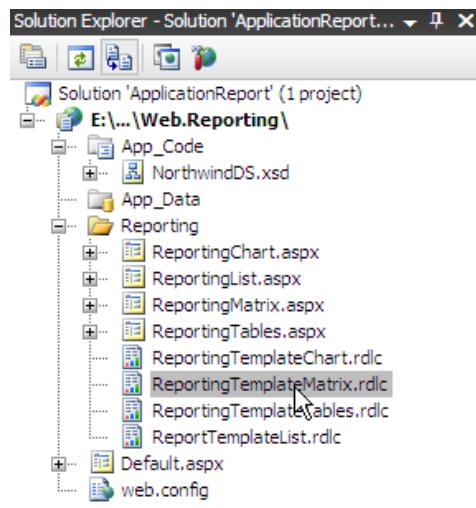
Matrix (Cross Table)

Matrix adalah control yang digunakan untuk kebutuh report cross table, berikut tahapan atau cara mnegunakan control matrix pada report viewer.

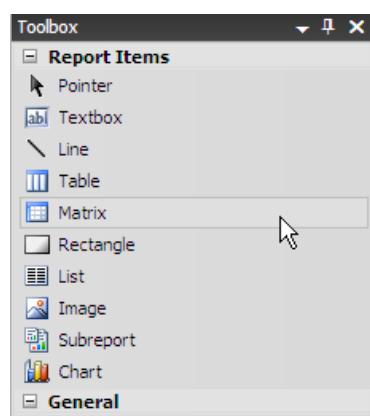
1. Buat page dengan nama ReportingMatrix.aspx pada folder Reporting



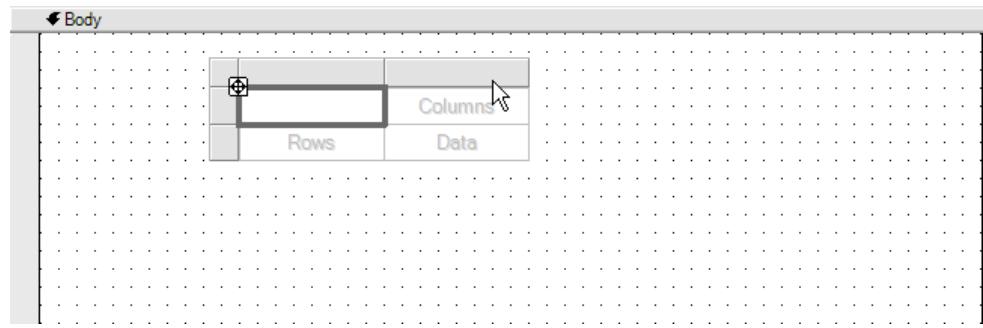
2. Tambahkan file ReportingTemplateMatrix.rdlc pada folder Reporting



3. Data yang akan ditampilkan menggunakan datatable OrderByEmployeeDT, tambahkan datasource kedalam file ReportingTemplateMatrix.rdlc
4. Drag control Matrix yang ada di toolbox kedalam layout designer



5. Tampilan akan terlihat seperti dibawah



Drag data yang akan ditampilkan kedalam Tables dari Website Data Sources

=Fields!OrderDate.Value	
=Fields!FirstName.Value	=Sum(Fields!Count.Value)

6. Tambahkan pada File ReportingMatrix.aspx control MicrosoftReportViewer untuk menampilkan report dari file .rdlc

ReportingMatrix.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="ReportingMatrix.aspx.cs"
Inherits="Reporting_ReportingMatrix" %>
<%@ Register Assembly="Microsoft.ReportViewer.WebForms,
Version=9.0.0.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a"
Namespace="Microsoft.Reporting.WebForms" TagPrefix="rsweb" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Untitled Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <rsweb:ReportViewer ID="ReportViewer1" Height="1000px"
Width="100%" ShowFindControls="False"
ShowPageNavigationControls="False" runat="server"
ExportContentDisposition="AlwaysInline">
            </rsweb:ReportViewer>
        </div>
    </form>
</body>
</html>
```

ReportingMatrix.aspx.cs

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
```

```

using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;

//add Namespace
using Microsoft.Reporting.WebForms;

public partial class Reporting_ReportMatrix : 
System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            // path of rdlc file
            ReportViewer1.LocalReport.ReportPath =
"Reporting\\ReportingTemplateMatrix.rdlc";

            ReportViewer1.ProcessingMode =
ProcessingMode.Local;
            ReportViewer1.LocalReport.DataSources.Clear();

            // adding data source to reportviewer
            ReportViewer1.LocalReport.DataSources.Add(new
ReportDataSource("NorthwindDS_OrderByEmployeeDT", new
NorthwindDSTableAdapters.OrderByEmployeeDTTableAdapter().GetDa
ta()));
            ReportViewer1.LocalReport.Refresh();
        }
    }
}

```

7. Click kanan pada page ReportingMatrix.aspx maka tampilan aka terlihat seperti di bawah :

Employee \ Order Date	3/3/1997 12:00:00 AM	4/7/1998 12:00:00 AM	1/22/1997 12:00:00 AM	9/22/1997 12:00:00 AM	4/9/1998 12:00:00 AM	12/26/1997 12:00:00 AM	9/18/1997 12:00:00 AM	12/17/1997 12:00:00 AM	11/12/1
	2	2	1	3	6	2	3	3	3
Andrew									
Anne									
Janet									
Laura									
Margaret									1
Michael									
Nancy				2	3	3			
Steven									

8. Rubah format date dari (3/3/1997 12:00:00 AM) menjadi (3/3/1997) caranya masukan code berikut pada column di matrix

Employee \ Order	=Month(Fields!OrderD
=Fields!FirstName.\	=Sum(Fields!count

```
=Month(Fields!OrderDate.Value) & "/" &  
Day(Fields!OrderDate.Value) & "/" & Year(Fields!OrderDate.Value)
```