



INDONESIA
.NET DEVELOPER COMMUNITY



Seri Belajar

ASP.NET 4.0

Membangun Sistem Pengelolaan User

M Reza Faisal, Microsoft MVP ASP/ASP.NET

Kata Pengantar

Puji dan syukur diucapkan kepada Allah SWT atas selesainya ebook sederhana yang berjudul Membangun Sistem Pengelolaan User dengan menggunakan ASP.NET 4.0.

Ebook ini adalah ebook kedua yang membahas tentang ASP.NET setelah sebelumnya telah dibuat Ebook dengan judul [Hand on Lab : ASP.NET 3.5](#).

Pada ebook ini akan dipaparkan tahapan-tahapan untuk membuat Sistem Pengelolaan User dari tahap persiapan database, konfigurasi, pembuatan fondasi aplikasi web seperti Data Access Layer sederhana dengan menggunakan LINQ to SQL, pengamanan aplikasi sederhana dan lain-lain.

Sistem Pengelolaan User menggunakan template Web Site Project. Pada setiap tahapan pembangunan aplikasi web ini akan diperkenalkan control-control yang digunakan dan fungsinya. Dengan cara ini diharapkan pembaca akan lebih mengetahui kegunaan control-control ASP.NET. Pada ebook ini juga akan diinformasikan fitur-fitur baru yang dimiliki oleh ASP.NET 4.0.

Selain ASP.NET 4.0, pada ebook ini juga akan diperkenalkan beberapa control AJAX Control Toolkit untuk mempercantik antarmuka dan membuat interaksi user terhadap aplikasi lebih mudah.

Akhir kata, selamat membaca dan semoga ebook ini bermanfaat bagi para web developer pemula untuk membuat aplikasi web. Kritik dan saran akan sangat berarti dan dapat ditujukan via email.

Banjarmasin, Mei 2010

M Reza Faisal

(reza.faisal@gmail.com)

Daftar Isi

<i>Kata Pengantar.....</i>	<i>I</i>
<i>Daftar Isi.....</i>	<i>II</i>
<i>Daftar Gambar.....</i>	<i>V</i>
<i>1 Pendahuluan.....</i>	<i>1-1</i>
Sistem Pengelolaan User	1-1
Mengelola Role.....	1-1
Mengelola User	1-1
Authentication.....	1-1
Mengelola Menu	1-1
Teknologi.....	1-2
ASP.NET 4.0.....	1-2
SQL Server 2008	1-2
<i>2 Persiapan.....</i>	<i>2-3</i>
Persiapan Database	2-3
Membuat Database	2-3
Menyiapkan Database	2-5
Konfigurasi Project Web.....	2-8
Membuat Web Site Project.....	2-8
Membuat Web Application Project	2-10
Konfigurasi	2-11
Menyiapkan Role dan User.....	2-15
Menyiapkan Role	2-15
Menyiapkan User.....	2-16
Menambah Atribut pada User	2-17
Persiapan Database Data	2-19
Apa yang telah dipelajari ?.....	2-20
<i>3 Pembangunan Fondasi</i>	<i>3-21</i>
Pembangunan Area Public.....	3-21
Membuat MasterPage Public.....	3-21
Membuat Theme	3-23
Membuat Antarmuka Public.....	3-24

Apa yang telah dipelajari?	3-35
Pembangunan Area Member	3-36
Membuat Master Page Member	3-36
Membuat Fitur Logout	3-38
Menyiapkan Sitemap	3-38
Mendaftarkan Sitemap	3-40
Membuat Menu	3-41
Hak Akses	3-45
Menambahkan SiteMapPath	3-46
Apa yang telah dipelajari ?	3-47
Struktur Web Site SPU.Web	3-47
Data Access Layer.....	3-47
SPU.Entities	3-48
SPU.DataAccess	3-51
Create, Retrieve, Update dan Delete	3-53
Apa yang telah dipelajari?	3-55
4 Data Master.....	4-56
Design dan Data Tabel Organisasi.....	4-56
Mengelola Organisasi	4-57
Daftar Organisasi	4-68
Detail Organisasi	4-69
Edit Organisasi	4-70
Tambah Organisasi	4-71
Hapus Organisasi.....	4-71
Apa yang telah dipelajari?.....	4-72
5 User Management	5-73
Mengelola User.....	5-73
Kode Lengkap	5-76
Penjelasan Kode	5-89
Daftar User	5-90
Detail User	5-92
Edit User.....	5-94
Tambah User.....	5-96
Hapus User	5-98

Profile User.....	5-99
Apa yang telah dipelajari?.....	5-101
6 Pembenahan.....	6-102
Security.....	6-102
Resource dan Globalization.....	6-105
Global Resource.....	6-106
Local Resource.....	6-109
7 Penutup.....	7-112

Daftar Gambar

Gambar 1. Memulai Membuat Database Baru.	2-4
Gambar 2. Memberi Nama Database.	2-4
Gambar 3. Daftar Database.	2-5
Gambar 4. Isi Database SPU_User.	2-5
Gambar 5. ASP.NET SQL Server Setup Wizard.	2-6
Gambar 6. Configure SQL Server for application service.	2-6
Gambar 7. Memilih Database untuk dimodifikasi.	2-7
Gambar 8. Hasil modifikasi database SPU_User.	2-7
Gambar 9. Blank Solution.	2-9
Gambar 10. Membuat Web Site baru.	2-9
Gambar 11. ASP.NET Empty Web Site.	2-10
Gambar 12. SPU.Web pada Solution Explorer.	2-10
Gambar 13. Web Application Project.	2-11
Gambar 14. ASP.NET Configuration.	2-13
Gambar 15. ASP.NET Web Site Administration Tool.	2-14
Gambar 16. Opsi cara akses.	2-14
Gambar 17. Hasil pemilihan nilai tipe autentifikasi menjadi Forms.	2-15
Gambar 18. Mengelola Role.	2-16
Gambar 19. Membuat Role baru.	2-16
Gambar 20. Existing role.	2-16
Gambar 21. Create user.	2-17
Gambar 22. Form untuk membuat user.	2-17
Gambar 23. Pesan bahwa user berhasil dibuat.	2-17
Gambar 24. Struktur tabel Users dan Membership.	2-18
Gambar 25. Tabel Organisasi.	2-19
Gambar 26. Template Master Page.	3-22
Gambar 27. Struktur folder untuk menyimpan file Master Page.	3-22
Gambar 28. Membuat Theme.	3-23
Gambar 29. Theme pada Solution Explorer.	3-24
Gambar 30. Gambar antarmuka Public.master.	3-25
Gambar 31. Struktur folder penyimpanan file CSS dan gambar.	3-25
Gambar 32. Antarmuka halaman login.	3-27

Gambar 33. Default.aspx, Halaman login.	3-27
Gambar 34. Panel dengan ujung melengkung.	3-28
Gambar 35. Panel.	3-29
Gambar 36. Library AjaxControlToolkit pada folder Bin.	3-30
Gambar 37. Web User Control pada Web Application Project.	3-31
Gambar 38. Lokasi file LoginForm.ascx.	3-32
Gambar 39. Web User Control berhasil ditampilkan.	3-33
Gambar 40. Web User Control untuk form login.	3-34
Gambar 41. Control Login pada Toolbox Visual Studio 2010.	3-34
Gambar 42. Sebelum dan sesudah login.	3-35
Gambar 43. Antarmuka Member.master.	3-36
Gambar 44. Fitur Logout dengan menggunakan LoginStatus.	3-38
Gambar 45. Sitemap Template.	3-39
Gambar 46. Web.sitemap.	3-40
Gambar 47. Control Navigation pada Toolbox.	3-40
Gambar 48. Control SiteMapDataSource pada kelompok Data.	3-41
Gambar 49. Sitemap Provider.	3-42
Gambar 50. Menu dengan style CSS.	3-45
Gambar 51. SiteMapPath.	3-46
Gambar 52. Hirarki pada file Web.sitemap.	3-46
Gambar 53. Struktur file dan folder pada SPU.Web.	3-47
Gambar 54. Menambah project baru pada solution.	3-48
Gambar 55. SPU.Entities.	3-49
Gambar 56. Menambahkan LINQ to SQL Classes, SPUSystem.dbml.	3-49
Gambar 57. Membuat class LINQ to SQL.	3-50
Gambar 58. File konfigurasi app.config dan Settings.settings.	3-50
Gambar 59. Class SPUDataAccess.cs.	3-51
Gambar 60. Jendela Add Reference	3-51
Gambar 61. Daftar reference lengkap.	3-52
Gambar 62. Daftar reference pada project SPU.Web.	3-53
Gambar 63. Data tabel Organisasi.	4-56
Gambar 64. Menampilkan data.	4-57
Gambar 65. Detail Data.	4-57
Gambar 66. Menambah data organisasi.	4-58
Gambar 67. Tambah data.	4-58

Gambar 68. Form edit data.....	4-58
Gambar 69. Memilih item yang akan dihapus.	4-59
Gambar 70. Tombol Hapus.	4-59
Gambar 71. Konfirmasi sebelum menghapus.	4-59
Gambar 72. TreeView dengan atribut ShowLines dan ShowCheckBoxes.	4-68
Gambar 73. Form edit.	4-70
Gambar 74. Daftar user.	5-73
Gambar 75. Form menambah data.	5-74
Gambar 76. Popup Control untuk menampilkan data organisasi.	5-74
Gambar 77. Detail salah satu data user.	5-75
Gambar 78. Detail data admin.	5-75
Gambar 79. Edit data user.	5-75
Gambar 80. Nilai pada atribut RepeatLayout.....	5-76
Gambar 81. Control untuk menampilkan data di dalam GridView.	5-92
Gambar 82. Control CheckBoxList dan TreeView.	5-95
Gambar 83. Form tambah.	5-97
Gambar 84. CheckBox pada daftar user.	5-98
Gambar 85. Profile user.	5-99
Gambar 86. Web.sitemap.....	6-102
Gambar 87. Mengelola Access Rules.	6-103
Gambar 88. Web.config untuk mengatur folder.	6-103
Gambar 89. Form tambah user.	6-105
Gambar 90. Daftar user.	6-105
Gambar 91. Menambah folder App_GlobalResources.	6-106
Gambar 92. Membuat file SPU.resx.	6-106
Gambar 93. Isi file SPU.resx	6-107
Gambar 94. Global resource bahasa Inggris.	6-108
Gambar 95. Detail Organisasi	6-109
Gambar 96. Membuat folder App_LocalResources.	6-110
Gambar 97. Data di dalam OrganisasiList.aspx.resx dan OrganisasiList.aspx.id.resx.	6-110
Gambar 98. Data di dalam OrganisasiList.aspx.en.resx.	6-110
Gambar 99. Struktur folder App_LocalResources.	6-111

Sistem Pengelolaan User

Ketika membangun sistem atau software yang mempunyai karakteristik *multi-user*, maka Sistem Pengelolaan User akan sangat diperlukan. Pada Sistem Pengelolaan User terdapat minimal fitur seperti berikut ini :

- a. Mengelola *Role*.
- b. Mengelola User.
- c. *Authentication*.
- d. Mengelola Menu.

Mengelola Role

Role adalah cara sederhana untuk memberikan hak kepada user untuk mengakses 'area' tertentu. User dapat memiliki lebih satu hak akses atau *Role*, sehingga dimungkinkan user dapat mengakses lebih dari satu 'area' saja.

Pada ebook ini, pengelolaan *Role* tidak mencakup proses CRUD (*Create*, *Retrieve*, *Update* dan *Delete*) secara penuh, tetapi hanya akan diterangkan bagaimana proses pengambilan data (*retrieve*) *Role*. Tujuan dari proses tersebut adalah untuk menggunakan data saat proses pengelolaan user.

Mengelola User

Pada proses pengelolaan user, akan ada proses CRUD data user. Pada proses *create* dan *update* data dimungkinkan pengguna untuk memilih role untuk user tersebut.

Pada proses ini juga dimungkinkan untuk menambah item atau properti tambahan pada data user. Sebagai contoh memberikan properti tambahan seperti :

- a. Alamat.
- b. NIP atau Nomor Induk Pegawai.
- c. Dan lain-lain.

Authentication

User yang telah didaftarkan akan dapat melakukan login pada sistem setelah melewati proses *authentication*.

Mengelola Menu

Fitur ini memungkinkan menu ditampilkan sesuai dengan *Role* yang dimiliki oleh user yang sedang login.

Teknologi

ASP.NET 4.0

Sesuai judul dari ebook ini, maka ASP.NET 4.0 adalah salah satu teknologi yang akan dimanfaatkan untuk membangun Sistem Pengelolaan User sederhana ini.

Pada ebook ini akan dipaparkan hal-hal berikut ini :

- a. Konfigurasi yang harus dilakukan untuk proses pengelolaan *User*, *Role* dan menu.
- b. Penggunaan server control yang berhubungan pengelolaan *User*, *Role* dan menu.

SQL Server 2008

Sistem Pengelolaan User sederhana ini akan dibangun dengan menggunakan SQL Server 2008 sebagai media penyimpanan.

Ada dua hal yang harus dipersiapkan untuk membangun Sistem Pengelolaan User sederhana ini, yaitu :

- a. Persiapan database untuk menyimpan data Role, user dan data lain yang berhubungan dengan pengelolaan user.
- b. Konfigurasi pada web site / web application.

Persiapan Database

Saat menggunakan server control yang berada pada kelompok Login yang berada di Toolbox Visual Studio, maka secara otomatis sebuah database dengan nama `ASPNETDB.MDF` akan dibuat dan disimpan pada folder `App_Data`. Folder ini adalah folder 'resmi' yang biasa terdapat pada template Web Site Project atau Web Application Project. Tetapi hal ini dapat terjadi dengan syarat SQL 2008 Express (bisa menggunakan SQL 2005 Express juga) sudah diinstal.

Di dalam database `ASPNETDB.MDF` akan ditemui tabel, view dan stored procedure yang mendukung pengelolaan *Role* dan *User*.

Untuk sebuah aplikasi web, biasanya penulis menggunakan paling tidak dua buah database yaitu :

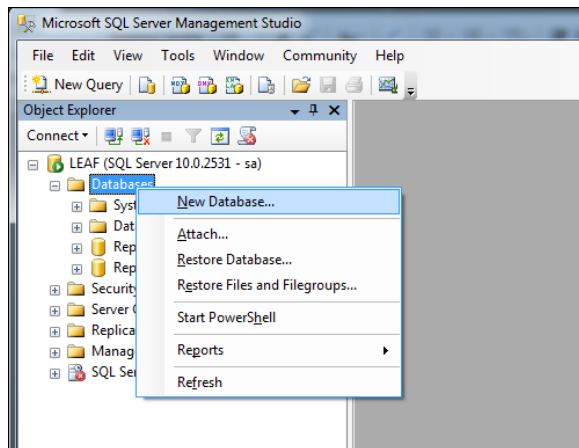
- a. Database untuk menyimpan data dari aplikasi web tersebut.
- b. Database untuk menyimpan data *User* dan *Role* untuk aplikasi web tersebut.

Antara data dari aplikasi web dengan data *User* dapat disimpan dalam satu database tetapi hal itu lebih baik dihindari.

Pada ebook ini akan diterangkan bagaimana memodifikasi database baru agar dapat digunakan untuk menyimpan data *Role* dan *User*.

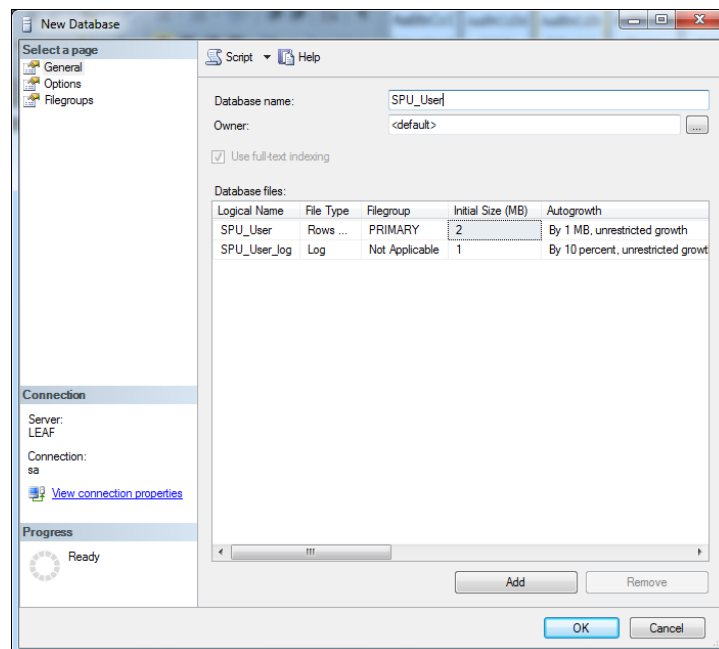
Membuat Database

Berikut ini adalah langkah untuk membuat database pada SQL Server 2008. Setelah berhasil login pada Microsoft SQL Server Management Studio, buat database baru dengan mengklik kanan pada bagian Databases dan pilih New Database, seperti pada gambar di bawah ini.



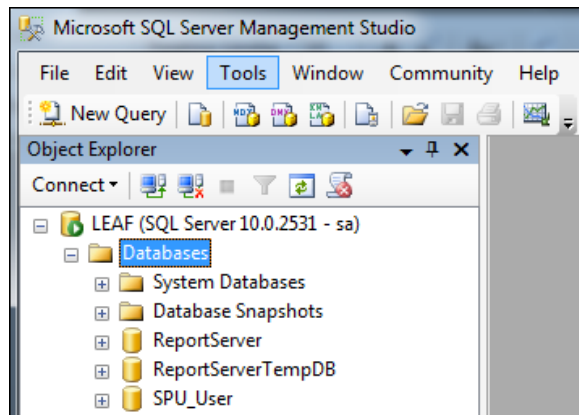
Gambar 1. Memulai Membuat Database Baru.

Setelah memberikan nama database seperti pada gambar di bawah ini.



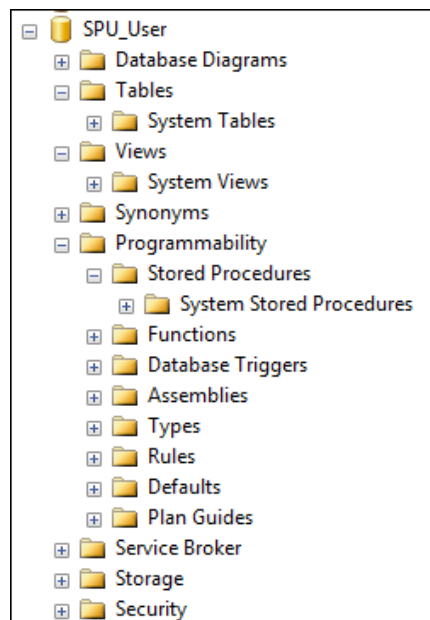
Gambar 2. Memberi Nama Database.

Hasilnya akan dapat dilihat pada daftar database di bawah ini.



Gambar 3. Daftar Database.

Kalau diperhatikan isi dari database SPU_User yang baru saja dibuat dapat dilihat pada gambar di bawah ini.



Gambar 4. Isi Database SPU_User.

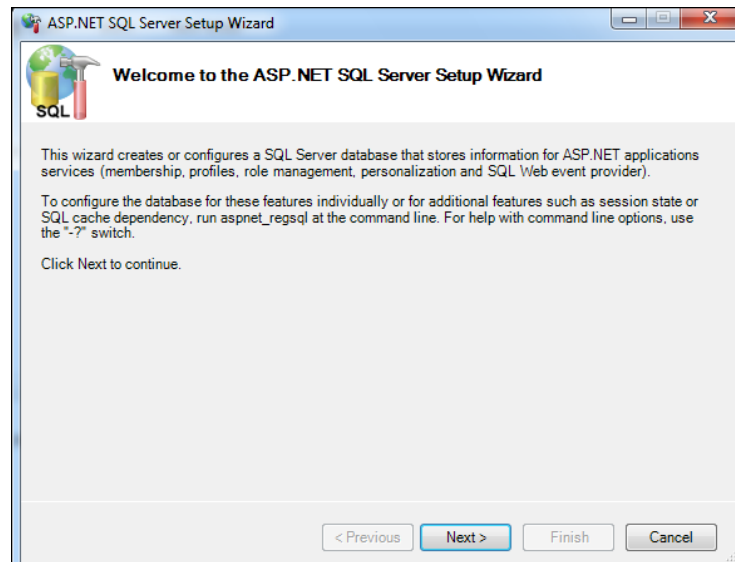
Database SPU_User belum mempunyai tabel, view dan stored procedure.

Menyiapkan Database

Proses persiapan database disini dimaksudkan untuk membuat tabel, view dan stored procedure untuk kebutuhan pengelolaan role dan user. Untuk membuat semua itu dapat menggunakan bantuan aplikasi kecil yang terdapat pada folder

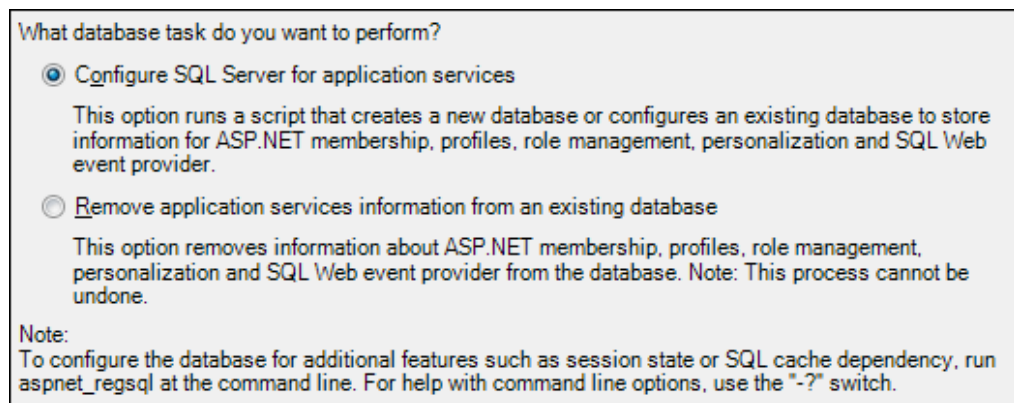
C:\Windows\Microsoft.NET\Framework\v4.0.30319 yaitu `aspnet_regsql.exe`.

Dengan menjalankan klik double pada `aspnet_regsql.exe` maka akan ditampilkan antarmuka berikut ini.



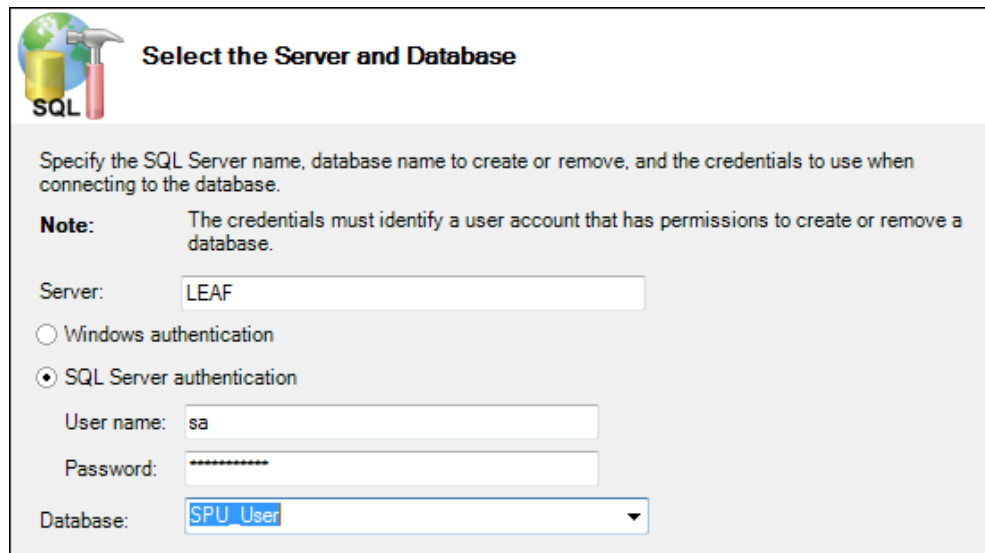
Gambar 5. ASP.NET SQL Server Setup Wizard.

Setelah memilih tombol Next, pilih opsi seperti pada gambar kemudian pilih tombol Next.



Gambar 6. Configure SQL Server for application service.

Langkah selanjutnya adalah memilih database yang akan dimodifikasi, dalam kasus ini adalah database dengan nama `SPU_User`.



Select the Server and Database

Specify the SQL Server name, database name to create or remove, and the credentials to use when connecting to the database.

Note: The credentials must identify a user account that has permissions to create or remove a database.

Server: LEAF

☐ Windows authentication

☒ SQL Server authentication

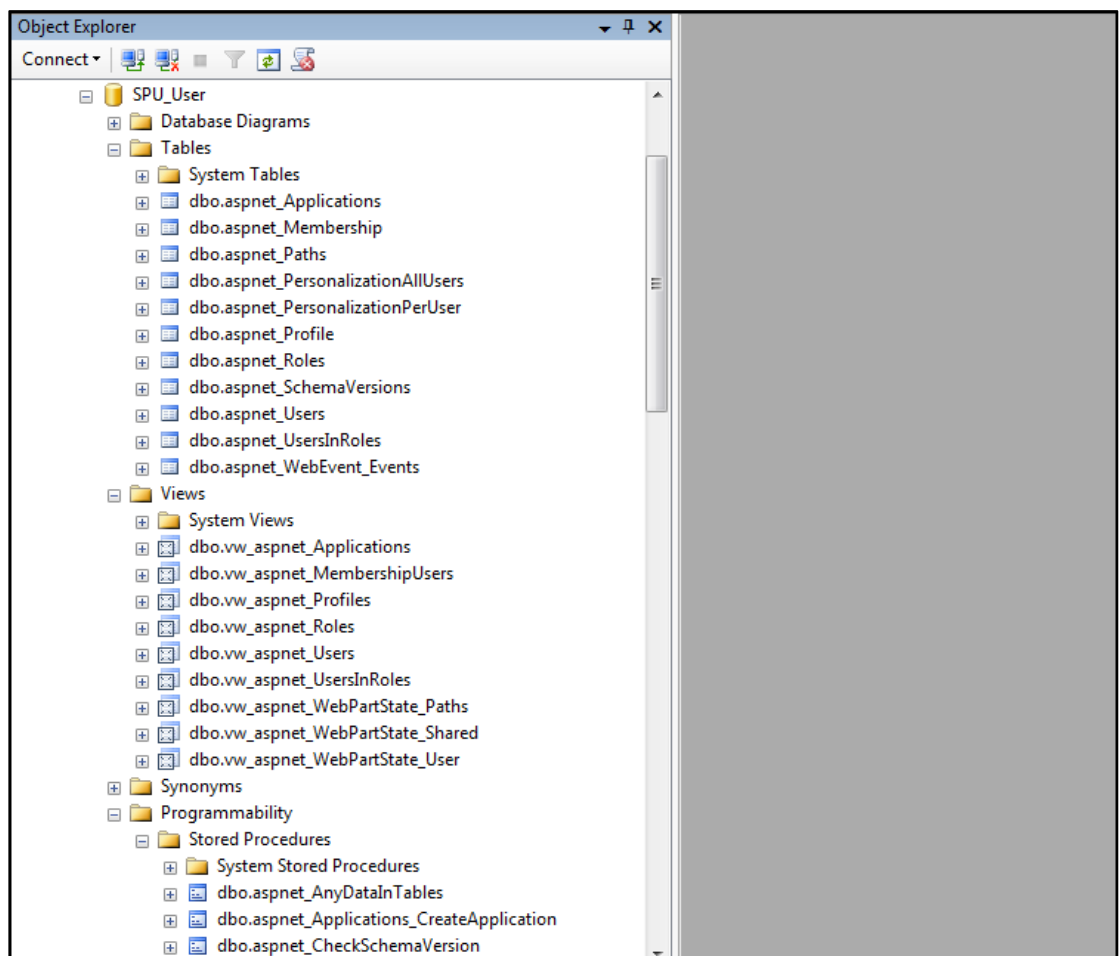
User name: sa

Password: *****

Database: SPU_User

Gambar 7. Memilih Database untuk dimodifikasi.

Setelah proses selesai, maka dapat dilihat perubahan pada database `SPU_User` seperti pada gambar di bawah ini. Terdapat penambahan item tabel, view dan stored procedure pada database ini.



Gambar 8. Hasil modifikasi database SPU_User.

Persiapan sudah selesai dilakukan dan database siap digunakan oleh aplikasi web.

Konfigurasi Project Web

Seperti yang telah diketahui, terdapat dua Project System untuk web developer, yaitu :

- a. Web Site Project.
- b. Web Application Project.

Ada perbedaan diantara keduanya, dan bisa disimpulkan dengan melihat tabel berikut ini.

	Web Site Project	Web Application Project
Kompilasi	Tidak	Iya, perlu dilakukan kompilasi bila melakukan perubahan pada kode.
File Class pada web server	Iya	Tidak, file class sudah dikompilasi menjadi file DLL.
Installer	Tidak	Iya
Mudah menggunakan Unit Testing	Tidak	Iya
Source code ada pada web server	Iya	Tidak, karena sudah dikompilasi.
Remote editing	Iya, dapat melakukan edit secara remote via FTP atau HTTP.	Tidak
ASP.NET MVC	Tidak	Iya
App_Code	Iya	Tidak
Edit kode saat debugging	Tidak	Iya
Web User Control Template (VS 2010)	Tidak	Iya
Code ASPX	CodeFile	CodeBehind
App_WebReferences	Iya	Tidak

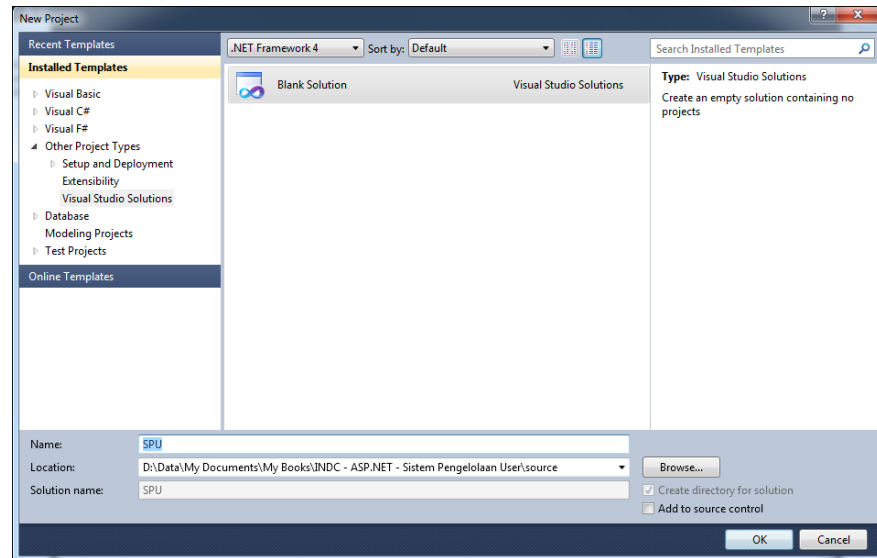
Dari sebagian daftar perbedaan antara Web Application Project dan Web Site Project, maka dapat dijadikan bahan pertimbangan untuk memutuskan tipe project mana yang cocok dengan proyek yang sedang ditangani.

Setelah database disiapkan maka langkah selanjutnya adalah melakukan konfigurasi pada Web Site Project atau Web Application Project. Konfigurasi ini dilakukan agar aplikasi web terkoneksi dengan database tersebut.

Membuat Web Site Project

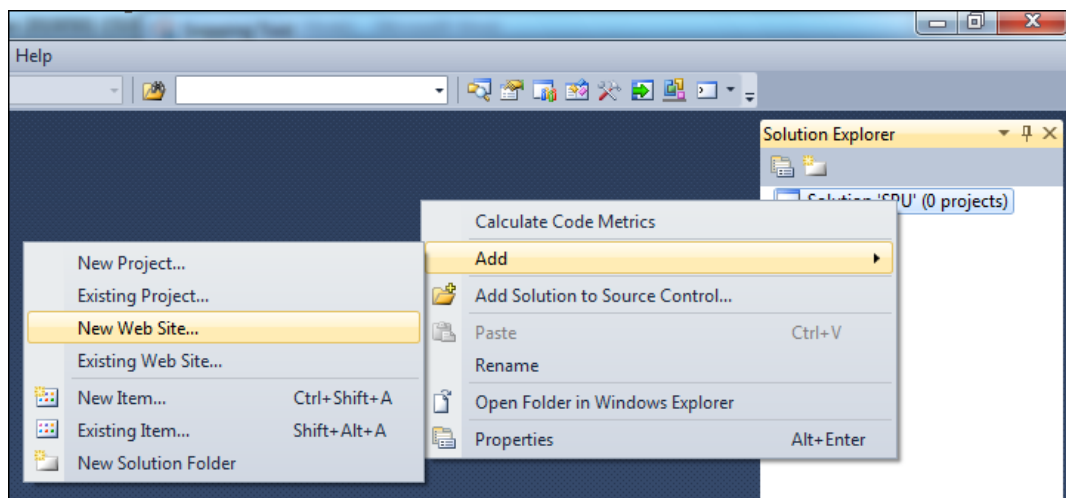
Berikut ini adalah langkah-langkah yang dilakukan untuk membuat Web Site Project dengan Visual Studio 2010.

Langkah pertama adalah membuat Solution kosong dengan cara memilih File > New > Project kemudian pada Installed Templates pilih Other Project Type > Visual Studio Solutions > Blank Solution seperti pada gambar di bawah ini.



Gambar 9. Blank Solution.

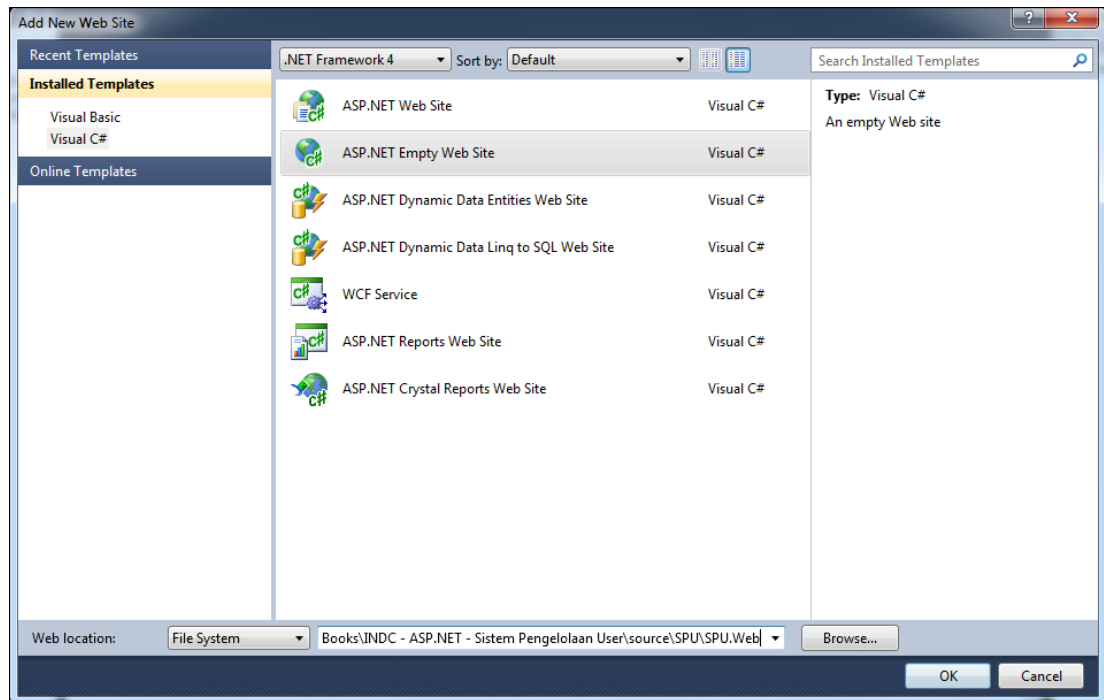
Pada kasus ini dibuat solution dengan nama *SPU*. Selanjutnya klik kanan pada *SPU* yang ada pada Solution Explorer dan buat Web Site baru seperti pada gambar di bawah ini.



Gambar 10. Membuat Web Site baru.

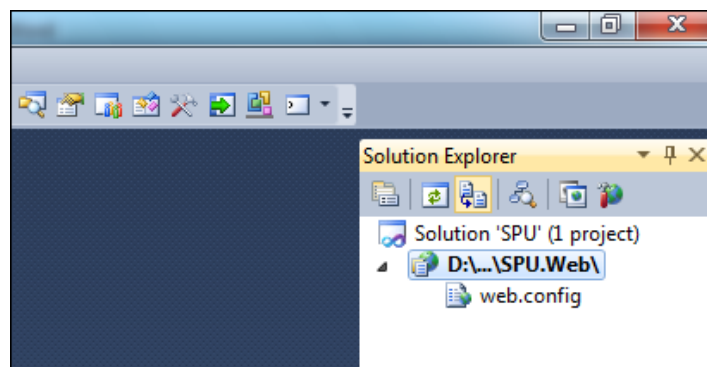
Setelah hal di atas dilakukan maka akan dilihat gambar seperti di bawah ini. Pilih bahasa yang digunakan, pada kasus ini digunakan bahasa C#. Setelah itu pilih ASP.NET Empty Web Site.

Perhatikan lokasi penyimpanan Web Site ini agar sesuai dengan lokasi folder dari solution. Pada kasus ini dibuat Web Site dengan nama *SPU.Web* yang disimpan pada folder *SPU* milik solution yang telah dibuat.



Gambar 11. ASP.NET Empty Web Site.

Hasilnya adalah seperti pada gambar di bawah ini.



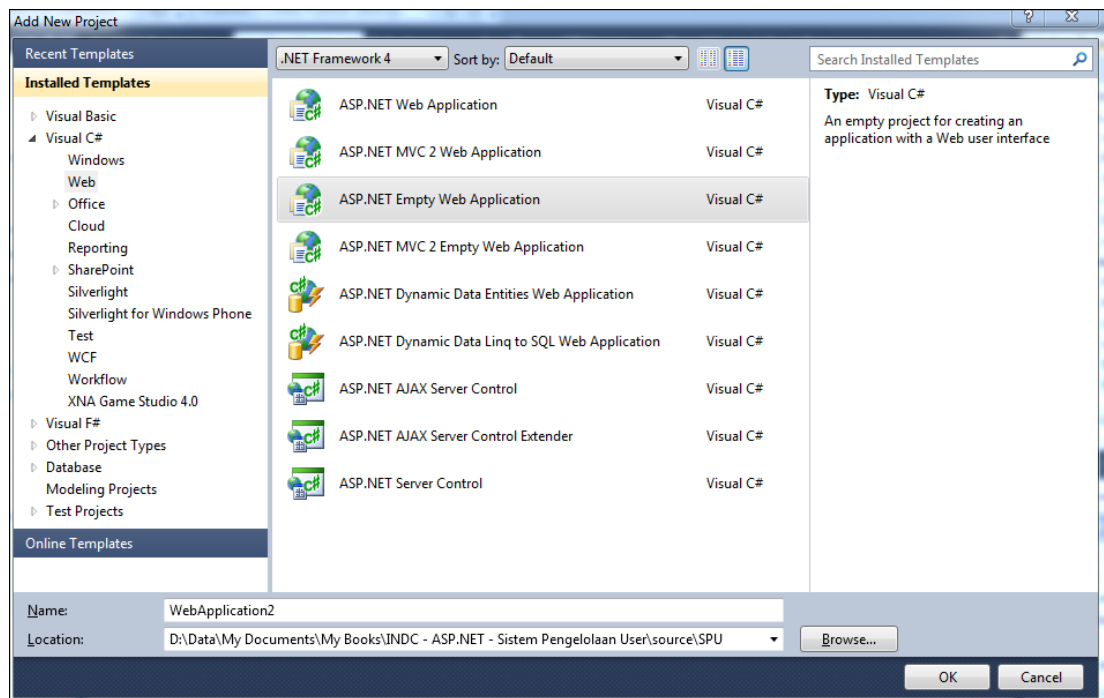
Gambar 12. SPU.Web pada Solution Explorer.

Web Site telah selesai dibuat dan siap digunakan.

Membuat Web Application Project

Bila ingin bekerja pada Web Application Project, maka setelah blank solution dibuat lakukan klik kanan pada solution dan pilih Add > New Project.

Pilih bahasa yang diinginkan, kemudian pilih ASP.NET Empty Web Application selanjutnya berikan nama project yang diinginkan.



Gambar 13. Web Application Project.

Selanjutnya pada ebook ini hanya akan menggunakan Web Site Project sebagai contoh kasus.

Konfigurasi

Pada ASP.NET, `web.config` adalah file konfigurasi utama yang menyimpan konfigurasi seperti cara akses ke database, AJAX, routing, pendaftaran referensi, konfigurasi Membership, Sitemap, konfigurasi Role dan lain-lain.

Ada perbedaan yang mencolok antara `web.config` yang ada pada ASP.NET 4.0 dengan ASP.NET versi sebelumnya. Pada `web.config` versi sebelumnya akan dapat dilihat file ini terisi dengan penuh baris-baris konfigurasi. File `web.config` pada ASP.NET 4.0 lebih terkesan minimalis.

Berikut ini adalah isi dari file `web.config` pada ASP.NET 4.0.

```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
  </system.web>
</configuration>
```

Hal ini mungkin terjadi karena baris-baris konfigurasi telah dipindahkan ke dalam `machine.config`. Pada file `web.config` yang ditampilkan di atas, baris di bawah ini adalah atribut baru yang dapat ditemui pada ASP.NET 4.0

```
<compilation debug="false" targetFramework="4.0" />
```

Atribut ini memungkinkan web developer memilih versi .NET Framework yang akan digunakan oleh aplikasi web yang dibuat.

Pada bagian ini akan diperlihatkan konfigurasi file `web.config` agar dapat menggunakan database `SPU_User`. Langkah pertama adalah menambah baris berikut ini pada file `web.config`.

```
<connectionStrings configSource="connectionString.config" />
```

Sehingga file `web.config` akan seperti berikut ini.

```
<?xml version="1.0"?>
<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->
<configuration>
  <connectionStrings configSource="connectionString.config" />
  <system.web>
    <compilation debug="false" targetFramework="4.0" />
  </system.web>
</configuration>
```

Pada konfigurasi ini sengaja membuat connection string pada file terpisah, yaitu pada file `connectionString.config` dengan isi sebagai berikut.

```
<?xml version="1.0" encoding="utf-8"?>

<connectionStrings>
  <add name="SPU User ConnString"
    connectionString="Persist Security Info=False;Integrated
Security=false;User ID=sa;Password=RAHASIA;Initial
Catalog=SPU_User;server=.;Trusted_Connection=False;Connect Timeout=30"
    providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Selanjutnya adalah menambahkan konfigurasi untuk mengelola Role dan Membership. Membership merupakan hal yang diperlukan untuk mengelola user. Berikut ini adalah isi file `web.config` yang telah dimodifikasi untuk keperluan tersebut.

```
<?xml version="1.0"?>

<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->

<configuration>

  <connectionStrings configSource="connectionString.config" />

  <system.web>
    <compilation debug="false" targetFramework="4.0" />

    <roleManager enabled="true" defaultProvider="AspNetSqlRoleProvider">
      <providers>
        <clear/>
        <add name="AspNetSqlRoleProvider"
          connectionStringName="SPU_User_ConnString"
          applicationName="SPU"
          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="SPU_User_ConnString"
      enablePasswordRetrieval="false"
      enablePasswordReset="true"
      requiresQuestionAndAnswer="false"
      applicationName="SPU"
      requiresUniqueEmail="true"
      passwordFormat="Hashed"
      maxInvalidPasswordAttempts="5"
      minRequiredPasswordLength="5"
      minRequiredNonalphanumericCharacters="0"
      passwordAttemptWindow="10"
      passwordStrengthRegularExpression=""/>
  </providers>
</membership>

</system.web>
</configuration>

```

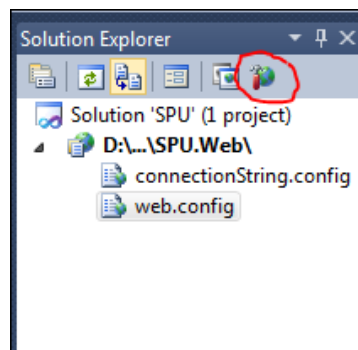
Terdapat dua item yang ditambahkan pada file `web.config` di atas, yaitu :

- a. `roleManager`, item ini berfungsi untuk menangani pengelolaan role.
- b. `membership`, item ini berfungsi untuk menangani pengelolaan user.

Ada dua atribut penting yang mesti diperhatikan pada masing-masing item tersebut, yaitu :

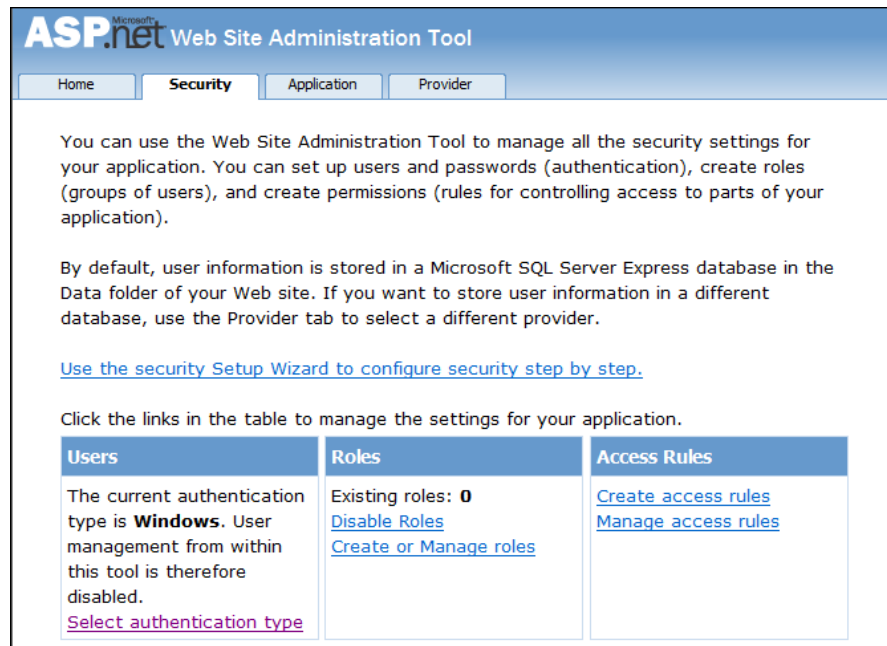
- a. `connectionStringName`, atribut ini berfungsi untuk menentukan nama yang ada pada connection string. Pada file `connectionString.config`, nama connection string yang digunakan adalah `SPU_User_ConnString`. Nama connection string ini digunakan oleh atribut `connectionStringName` yang ada pada `roleManager` dan `membership`.
- b. `applicationName`, atribut ini bernilai nama dari aplikasi web yang sedang dibuat. Developer dapat dengan bebas memberikan nilai dari atribut ini. Yang harus diperhatikan adalah nilai dari atribut ini harus sama pada setiap item yang ada. Maka dapat dilihat nilai atribut `applicationName` pada item `roleManager` dan `membership` adalah sama yaitu `SPU` (atau nama lain yang diinginkan). Sebenarnya satu database dapat digunakan untuk menyimpan role dan user oleh lebih dari satu aplikasi web. Caranya cukup dengan menentukan nilai atribut ini berbeda pada setiap aplikasi web tersebut.

Langkah selanjutnya adalah menjalankan ASP.NET Configuration untuk melihat keberhasilan konfigurasi yang telah dibuat.



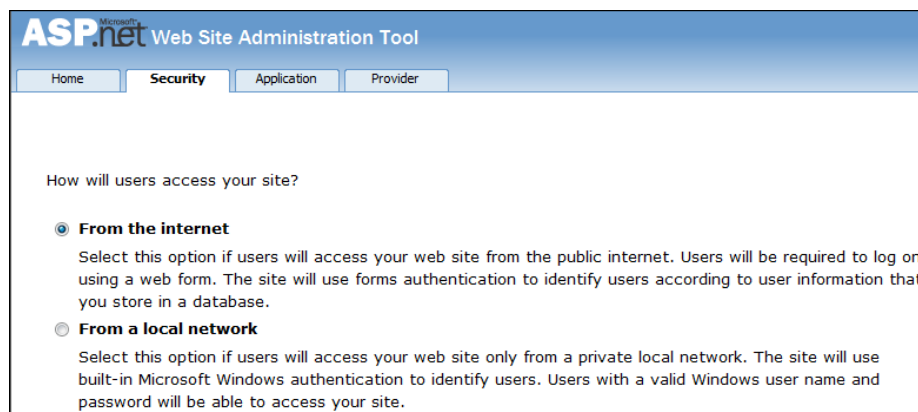
Gambar 14. ASP.NET Configuration.

Dengan mengklik tombol yang dilingkari warna merah seperti pada gambar di atas maka akan ditampilkan halaman web seperti berikut.



Gambar 15. ASP.NET Web Site Administration Tool.

Pada bagian Users, dapat dilihat tipe autentifikasi yang digunakan adalah `Windows` untuk itu harus diubah menjadi `Forms` dengan cara memilih link `Select authentication type`.



Gambar 16. Opsi cara akses.

Pilih opsi `From the Internet`, dengan memilih ini maka nilai tipe autentifikasi akan berubah menjadi `Forms`. Hal ini memungkinkan user harus melakukan proses login dengan form yang disediakan oleh aplikasi web.

Hasilnya akan dapat dilihat seperti pada gambar di bawah ini.



Gambar 17. Hasil pemilihan nilai tipe autentifikasi menjadi Forms.

Cara lain yang lebih mudah untuk mengubah tipe autentifikasi ini adalah dengan cara menambahkan baris berikut ini pada `web.config`.

```
<authentication mode="Forms" />
```

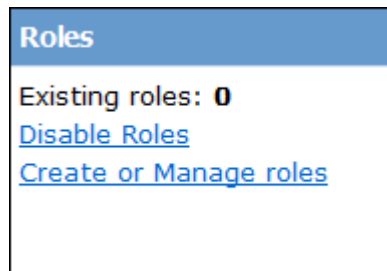
Menyiapkan Role dan User

Pada bagian ini akan dibuat Role dan User sebagai pendukung awal dari Sistem Pengelolaan User sederhana ini.

Role dan user tersebut akan dibuat dengan memanfaatkan ASP.NET Configuration atau ASP.NET Web Site Administration Tool.

Menyiapkan Role

Nama-nama role yang didaftarkan sebaiknya disesuaikan dengan kebutuhan/requirement. Jadi nama-nama role dapat disesuaikan dengan nama Actor yang terdapat pada Use Case. Dengan menjadikan Actor pada Use Case sebagai acuan pembuatan Role maka developer akan dengan mudah menentukan 'area' yang dapat diakses oleh suatu Role.



Gambar 18. Mengelola Role.

Langkah pertama untuk menyiapkan Role adalah dengan memilih link Create or Manage roles seperti pada gambar di atas ini. Selanjutnya akan dapat dilihat antarmuka seperti berikut ini.



Gambar 19. Membuat Role baru.

Pada kasus ini dibuat tiga buah role yaitu : Administrator dan Member.

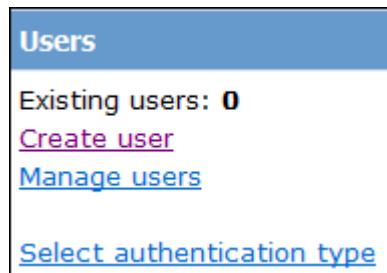


Gambar 20. Existing role.

Menyiapkan User

Selanjutnya adalah membuat satu user yang berfungsi sebagai administrator atau penguasa tertinggi pada Sistem Pengelolaan User sederhana ini.

Untuk membuat user baru, yang dilakukan adalah memilih link Create User pada halaman depan ASP.NET Web Site Administration Tool.



Gambar 21. Create user.

Kemudian isikan data user pada form yang telah disediakan seperti pada gambar di bawah ini.

Gambar 22. Form untuk membuat user.

Berikan seluruh role yang tersedia untuk user ini. Kemudian klik tombol Create User.

Gambar 23. Pesan bahwa user berhasil dibuat.

Menambah Atribut pada User

Di atas dapat dilihat item-item yang dapat diisi pada form untuk membuat user baru, yaitu :

- a. User name.

- b. Password.
- c. Email.

Dan berikut ini adalah struktur tabel Users dan Membership yang digunakan untuk menyimpan data user.

LEAF.SPU_User - ...aspnet_Membership

Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
Password	nvarchar(128)	<input type="checkbox"/>
PasswordFormat	int	<input type="checkbox"/>
PasswordSalt	nvarchar(128)	<input type="checkbox"/>
MobilePIN	nvarchar(16)	<input checked="" type="checkbox"/>
Email	nvarchar(256)	<input checked="" type="checkbox"/>
LoweredEmail	nvarchar(256)	<input checked="" type="checkbox"/>
PasswordQuestion	nvarchar(256)	<input checked="" type="checkbox"/>
PasswordAnswer	nvarchar(128)	<input checked="" type="checkbox"/>
IsApproved	bit	<input type="checkbox"/>
IsLockedOut	bit	<input type="checkbox"/>
CreateDate	datetime	<input type="checkbox"/>
LastLoginDate	datetime	<input type="checkbox"/>
LastPasswordChange...	datetime	<input type="checkbox"/>
LastLockoutDate	datetime	<input type="checkbox"/>
FailedPasswordAttem...	int	<input type="checkbox"/>
FailedPasswordAttem...	datetime	<input type="checkbox"/>
FailedPasswordAnsw...	int	<input type="checkbox"/>
FailedPasswordAnsw...	datetime	<input type="checkbox"/>
Comment	ntext	<input checked="" type="checkbox"/>

LEAF.SPU_User - dbo.aspnet_Users

LEAF.SPU_User - ...asp

Column Name	Data Type	Allow Nulls
ApplicationId	uniqueidentifier	<input type="checkbox"/>
UserId	uniqueidentifier	<input type="checkbox"/>
UserName	nvarchar(256)	<input type="checkbox"/>
LoweredUserName	nvarchar(256)	<input type="checkbox"/>
MobileAlias	nvarchar(16)	<input checked="" type="checkbox"/>
IsAnonymous	bit	<input type="checkbox"/>
LastActivityDate	datetime	<input type="checkbox"/>

Gambar 24. Struktur tabel Users dan Membership.

Ada kalanya perlu dibuat atribut tambahan yang diperlukan pada data user, misalnya seperti :

- a. Nomor Induk Pegawai.
- b. Nomor Induk Mahasiswa.
- c. Tipe User.
- d. Alamat Rumah.
- e. Nama Lengkap.
- f. Organisasi ID.
- g. Dan lain-lain.

Untuk menambahkan atribut-atribut tersebut tidak perlu membuat tabel baru, yang perlu dilakukan cukup mendaftarkan pada file `web.config`.

Pada kasus ini misalnya akan ditambahkan atribut-atribut berikut ini :

- a. NomorInduk, atribut ini untuk menyimpan nomor induk dari user.
- b. NamaLengkap, atribut untuk menyimpan nama lengkap user.
- c. Alamat, atribut untuk menyimpan alamat user.
- d. OrganisasiID, atribut yang menentukan organisasi dari user.

Berikut ini adalah cara untuk menambahkan keempat atribut ini pada `web.config` agar dapat digunakan pada aplikasi web.

```
<profile>
  <providers>
    <clear/>
    <add name="AspNetSqlProfileProvider"
          connectionStringName="SPU_User_ConnString"
```

```

        applicationName="SPU"
        type="System.Web.Profile.SqlProfileProvider, System.Web,
            Version=2.0.0.0, Culture=neutral,
            PublicKeyToken=b03f5f7f11d50a3a"/>
    </providers>
    <properties>
        <add name="NomorInduk"
            type="System.String"
            serializeAs="String"
            allowAnonymous="false"
            defaultValue=""
            readOnly="false"/>
        <add name="NamaLengkap"
            type="System.String"
            serializeAs="String"
            allowAnonymous="false"
            defaultValue=""
            readOnly="false"/>
        <add name="Alamat"
            type="System.String"
            serializeAs="String"
            allowAnonymous="false"
            defaultValue=""
            readOnly="false"/>
        <add name="OrganisasiID"
            type="System.String"
            serializeAs="String"
            allowAnonymous="false"
            defaultValue=""
            readOnly="false"/>
    </properties>
</profile>

```

Pada konfigurasi di atas, dapat dilihat terdapat penambahan item profile dengan nilai `connectionStringName` dan `applicationName` yang sama dengan item `roleManager` dan `membership`.

Di dalam item profile ini dapat dilihat bagaimana cara mendaftarkan atribut-atribut tambahan pada user sebagai property dari profile.

Persiapan Database Data

Pada konfigurasi di atas terdapat property `OrganisasiID`, untuk kebutuhan ini maka perlu dibuat satu tabel dengan nama `Organisasi` dengan atribut seperti berikut.

LEAF.SPU_Data - dbo.Organisasi			
	Column Name	Data Type	Allow Nulls
🔑	OrganisasiID	nvarchar(5)	<input type="checkbox"/>
	OrganisasiInduk	nvarchar(5)	<input checked="" type="checkbox"/>
	OrganisasiNama	nvarchar(50)	<input checked="" type="checkbox"/>
	Keterangan	nvarchar(113)	<input checked="" type="checkbox"/>

Gambar 25. Tabel Organisasi.

Tabel ini disimpan pada database baru dengan nama `SPU_Data`. Sehingga untuk Sistem Pengelolaan User sederhana ini akan menggunakan dua buah database yaitu `SPU_User` dan `SPU_Data`.

Setelah database `SPU_Data` dan tabel `Organisasi` dibuat maka perlu dilakukan penambahan connection string pada file `connectionString.config` seperti berikut ini.

```
<add name="SPU Data ConnString"
      connectionString="Persist Security Info=False;
                        Integrated Security=false;User ID=sa;
                        Password=RAHASIA;
                        Initial Catalog=SPU_Data;
                        server=.;Trusted Connection=False;
                        Connect Timeout=30"
      providerName="System.Data.SqlClient"/>
```

Sehingga akan dapat dilihat konfigurasi lengkap dari file connectionString.config seperti berikut ini.

```
<?xml version="1.0" encoding="utf-8"?>

<connectionStrings>
  <add name="SPU User ConnString"
        connectionString="Persist Security Info=False;
                          Integrated Security=false;
                          User ID=sa;Password= RAHASIA;
                          Initial Catalog=SPU User;server=.;
                          Trusted Connection=False;Connect Timeout=30"
        providerName="System.Data.SqlClient"/>

  <add name="SPU_Data_ConnString"
        connectionString="Persist Security Info=False;
                          Integrated Security=false;
                          User ID=sa;Password=RAHASIA;
                          Initial Catalog=SPU_Data;server=.;
                          Trusted_Connection=False;Connect Timeout=30"
        providerName="System.Data.SqlClient"/>
</connectionStrings>
```

Proses ini adalah akhir dari persiapan, selanjutnya adalah proses pembangunan software.

Apa yang telah dipelajari ?

Pada Bab 2 ini telah diperkenalkan hal-hal berikut ini :

- a. Cara menyiapkan database untuk keperluan pengelolaan user dan role dengan menggunakan `aspnet_regsql.exe`.
- b. Cara membuat Blank Solution.
- c. Cara membuat Web Site Project.
- d. Cara membuat Web Application Project.
- e. Mengenal fitur baru pada ASP.NET yaitu `web.config` file refactoring dan multi-targeting.
- f. Cara membuat Connection String.
- g. Cara konfigurasi Role.
- h. Cara konfigurasi Membership.
- i. Cara konfigurasi Profile.

Pembangunan Fondasi

Pada bagian ini akan dipaparkan langkah-langkah yang biasa dilakukan penulis dalam membangun 'rumah' bagi Sistem Pengelolaan User ini. Langkah-langkah yang dilakukan ini juga dapat diikuti pada saat membangun aplikasi web yang lainnya.

Istilah 'rumah' disini dimaksudkan sebagai pondasi bagi Sistem Pengelolaan User ini. Seperti halnya rumah sebenarnya, 'rumah' ini akan mempunyai 'pintu masuk' (autentifikasi) dengan fasilitas keamanan agar hanya orang yang mempunyai 'kunci' (user dan password) saja yang dapat masuk ke 'rumah' ini.

Setelah orang yang mempunyai 'kunci' tersebut masuk ke dalam 'rumah' maka orang tersebut akan diberikan fasilitas untuk mengakses menu sesuai dengan hak akses (role) yang dimiliki oleh orang tersebut. Menu ini akan mengantarkan orang tersebut pada area yang sesuai dengan hak akses (Role) yang dimilikinya.

Dari penjelasan tersebut di atas, maka fondasi dari aplikasi web ini akan mempunyai hal-hal seperti berikut ini :

- a. Sistem *autentifikasi*.
- b. Menu yang dapat menyesuaikan dengan role yang dimiliki oleh user yang sedang login.

Berikut ini adalah langkah-langkah yang dapat diikuti untuk membangun fondasi aplikasi web.

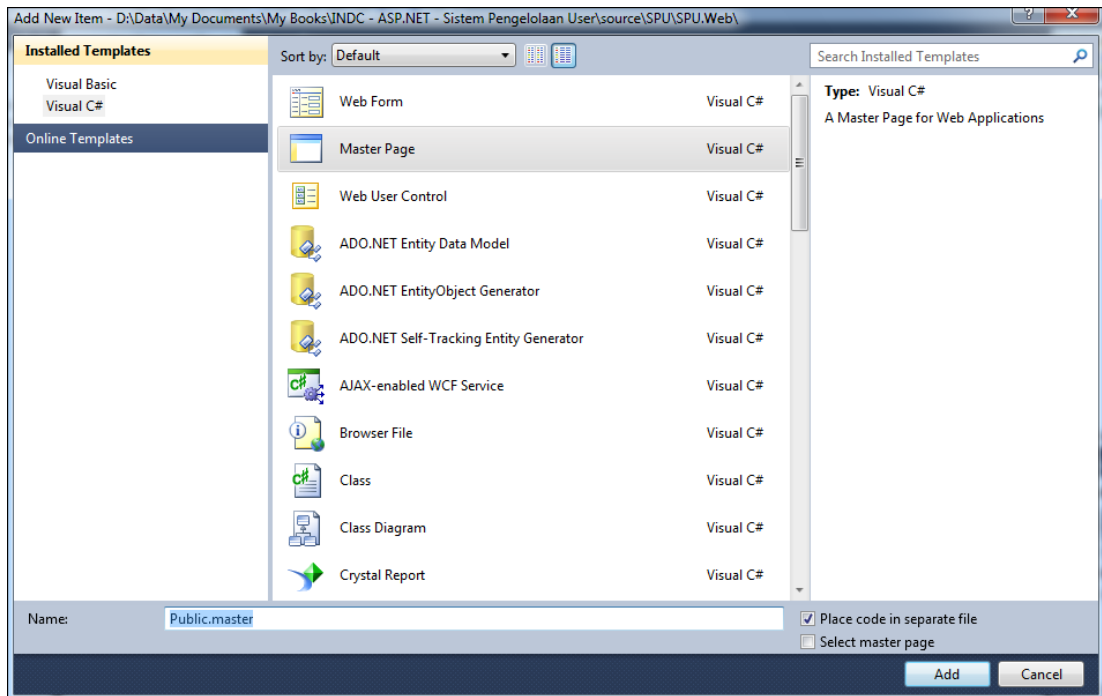
Pembangunan Area Public

Area public adalah area yang dapat dikunjungi oleh user tanpa harus login. Pada area ini dapat akan terdapat form login. Form login ini sudah mempunyai kemampuan autentifikasi sesuai dengan data user yang telah didaftarkan.

Membuat MasterPage Public

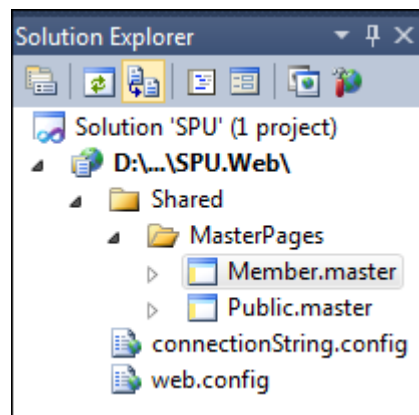
Master Page adalah media yang dapat digunakan untuk membuat layout yang konsisten untuk halaman-halaman pada aplikasi web. Pada satu Master Page selain dapat dibuat layout yang akan berlaku pada semua halaman yang menggunakannya, selain itu dapat pula diberikan perilaku atau behaviour pada Master Page sehingga perilaku ini juga akan berlaku pada halaman-halaman tersebut.

Pada area public ini dibuat satu Master Page dengan nama file `Public.master`.



Gambar 26. Template Master Page.

Kedua Master Page tersebut akan disimpan pada folder `Shared\MasterPages` seperti yang dilihat pada gambar di bawah ini.



Gambar 27. Struktur folder untuk menyimpan file Master Page.

Struktur seperti gambar di atas bukan keharusan, hanya kebiasaan penulis untuk mengelompokkan file agar terlihat rapi.

Selanjutnya dapat dilakukan perubahan pada file `Member.master` seperti berikut.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Sistem Pengelolaan User</title>
  <asp:ContentPlaceholder id="Head Member" runat="server">
  </asp:ContentPlaceholder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:ContentPlaceholder id="ContentPlaceHolder Member" runat="server">

        <asp:ContentPlaceholder>
      </div>
    </form>
```

```
</body>
</html>
```

Pada kode di atas ini perubahan yang terjadi adalah mengubah id pada setiap ContentPlaceHolder menjadi Head_Member dan ContentPlaceHolder_Member. Berikut pula pada file Public.member akan diubah menjadi seperti berikut ini.

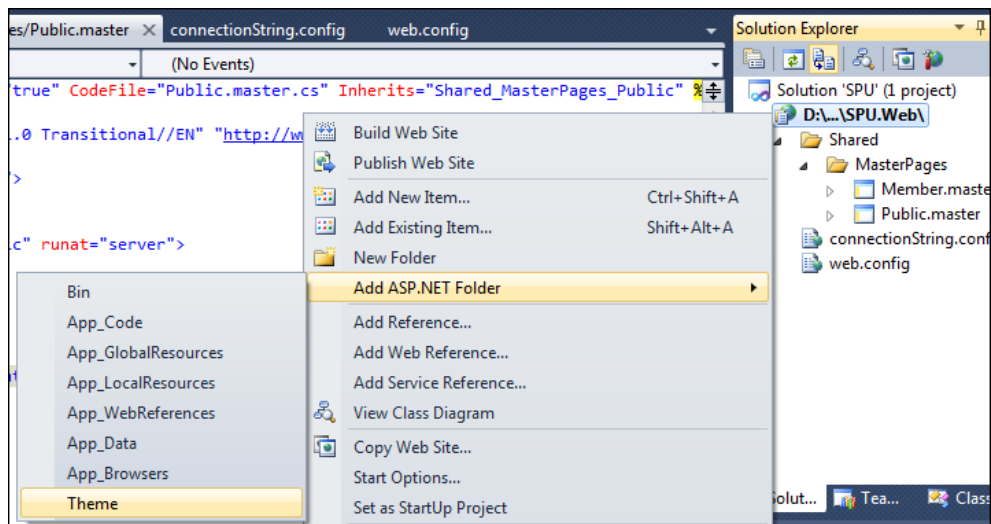
```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Sistem Pengelolaan User</title>
  <asp:ContentPlaceHolder id="Head Public" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:ContentPlaceHolder id="ContentPlaceHolder Public" runat="server">

      </asp:ContentPlaceHolder>
    </div>
  </form>
</body>
</html>
```

Tidak ada aturan yang mengharuskan perubahan id pada setiap ContentPlaceHolder seperti yang dilakukan di atas, hal ini dilakukan karena kebiasaan penulis untuk memberikan id yang dapat memberikan informasi berupa lokasi atau fungsinya.

Membuat Theme

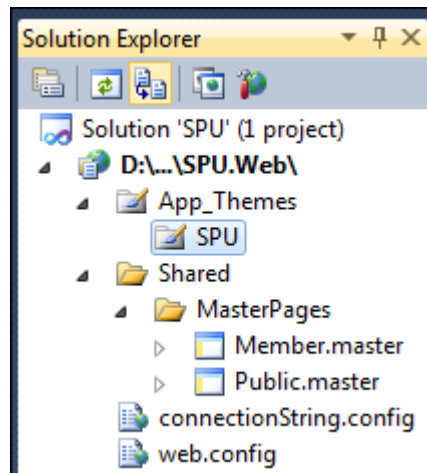
Theme adalah media yang dapat digunakan untuk menyimpan file Skin, Cascading Style Sheet (CSS) dan gambar yang digunakan untuk menghiasi antarmuka halaman web.



Gambar 28. Membuat Theme.

Untuk membuat Theme, klik kanan pada Web Site yang ada pada Solution Explorer kemudian pilih Add ASP.NET Folder > Theme.

Kemudian berikan nama pada Theme tersebut, pada kasus ini nama yang digunakan adalah SPU.



Gambar 29. Theme pada Solution Explorer.

Selanjutnya adalah melakukan konfigurasi agar Theme SPU aktif sehingga style-style yang ditulis pada Skin atau CSS dapat dimanfaatkan oleh halaman web yang ada pada Web Site ini. Jadi file-file CSS yang telah dibuat tidak perlu lagi di daftarkan pada halaman web (Master Page atau Web Form) seperti berikut ini :

```
<link href="../../App Themes/SPU/default.css" rel="stylesheet" type="text/css" />
```

Apabila Theme SPU telah diaktifkan maka seluruh file-file CSS yang tersimpan pada Theme ini akan didaftarkan didaftarkan seperti cara di atas secara otomatis.

Untuk mengaktifkannya cukup dengan menambahkan baris berikut ini pada web.config.

```
<pages theme="SPU"/>
```

Sehingga file web.config akan menjadi seperti berikut ini.

```
<?xml version="1.0"?>

<!--
For more information on how to configure your ASP.NET application, please visit
http://go.microsoft.com/fwlink/?LinkId=169433
-->

<configuration>

  <connectionStrings configSource="connectionString.config" />

  <system.web>
    <pages theme="SPU"/>
    .
    .
  </system.web>

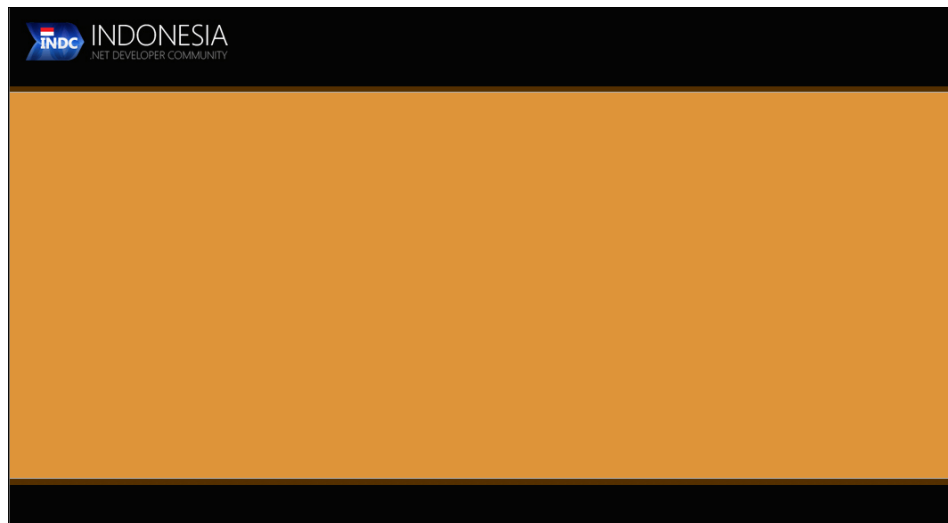
</configuration>
```

Membuat Antarmuka Public

Pada bagian ini akan diperlihatkan antarmuka pada Public.master dan halaman utama untuk login.

Antarmuka Master Page Public

Berikut ini adalah antarmuka pada `Public.master`.

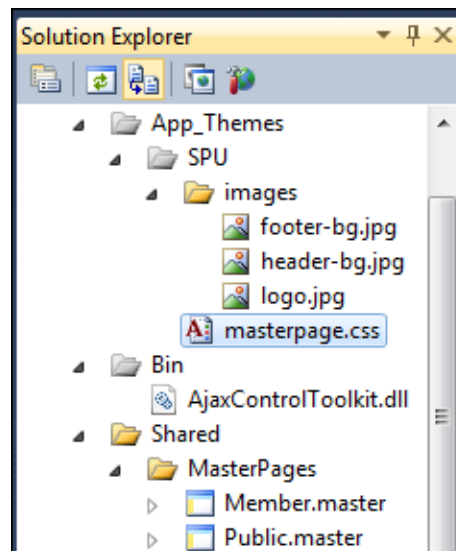


Gambar 30. Gambar antarmuka `Public.master`.

Untuk membuat tersebut menggunakan item-item seperti berikut :

- a. `masterpage.css`.
- b. `footer-bg.jpg`.
- c. `header-bg.jpg`.
- d. `logo.jpg`.

Item-item tersebut disimpan pada folder dengan struktur seperti gambar berikut ini.



Gambar 31. Struktur folder penyimpanan file CSS dan gambar.

Berikut ini isi dari file `Public.master`.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Public.master.cs"
Inherits="Shared MasterPages Public" %>

<!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></title>
    <meta http-equiv="X-UA-Compatible" content="IE=7" />
    <asp:ContentPlaceHolder id="Head_Public" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
        <div id="Header">
            <h1>INDC</h1>
            <div id="Logo"></div>
        </div>

        <div id="MainContent">
            <asp:ContentPlaceHolder id="ContentPlaceHolder_Public" runat="server">

            </asp:ContentPlaceHolder>
        </div>

        <div id="Footer"></div>
    </form>
</body>
</html>

```

Dan berikut ini adalah isi dari file masterpage.css.

```

body
{
    margin: 0;
    padding: 0;
    background-color:#040404;
}

#Header
{
    width: 100%;
    height: 94px;
    background-image: url(images/header-bg.jpg);
    background-repeat: repeat-x;
}

#Header h1
{
    display: none;
}

#Header #Logo
{
    width:236px;
    height:49px;
    background-image:url(images/logo.jpg);
    background-repeat:no-repeat;
    margin-left:13px;
    margin-top:13px;
}

#MainContent
{
    height:432px;
    min-height:432px;
    background-color:#de9439;
    border-top:solid 1px #b3b3b3;
    border-bottom:solid 1px #b3b3b3;
}

#Footer
{
    width: 100%;
    height: 50px;
    background-image: url(images/footer-bg.jpg);
    background-repeat: repeat-x;
}

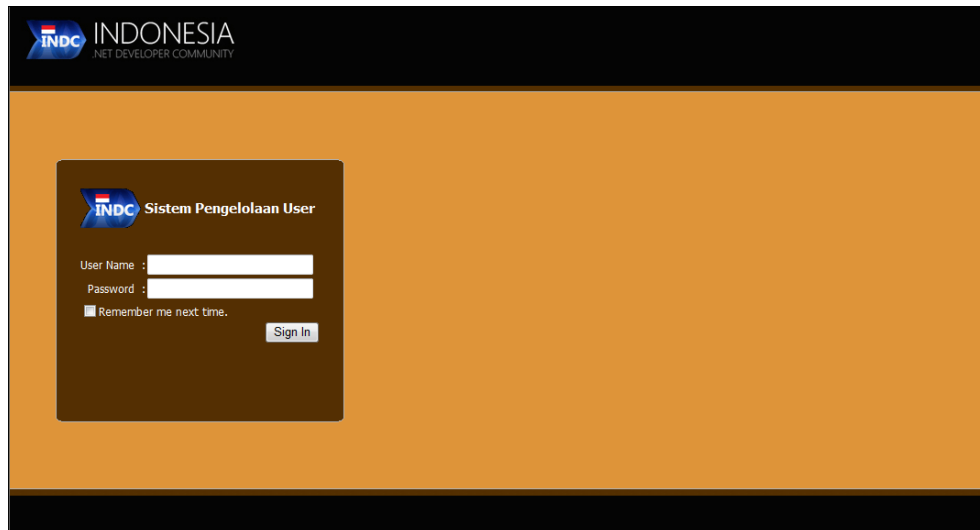
```

Catatan :

Silakan mengunduh source code untuk mendapatkan file-file tersebut.

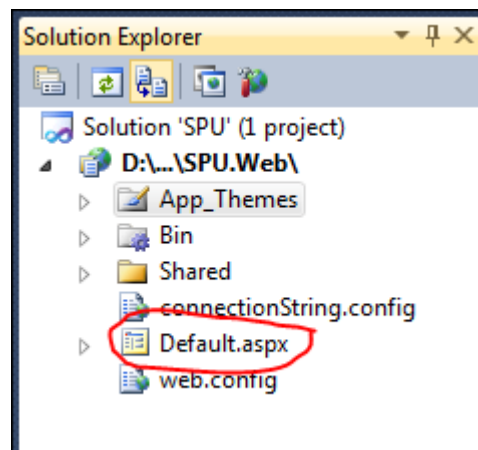
Antarmuka Halaman Login

Dan berikut ini adalah antarmuka untuk halaman login.



Gambar 32. Antarmuka halaman login

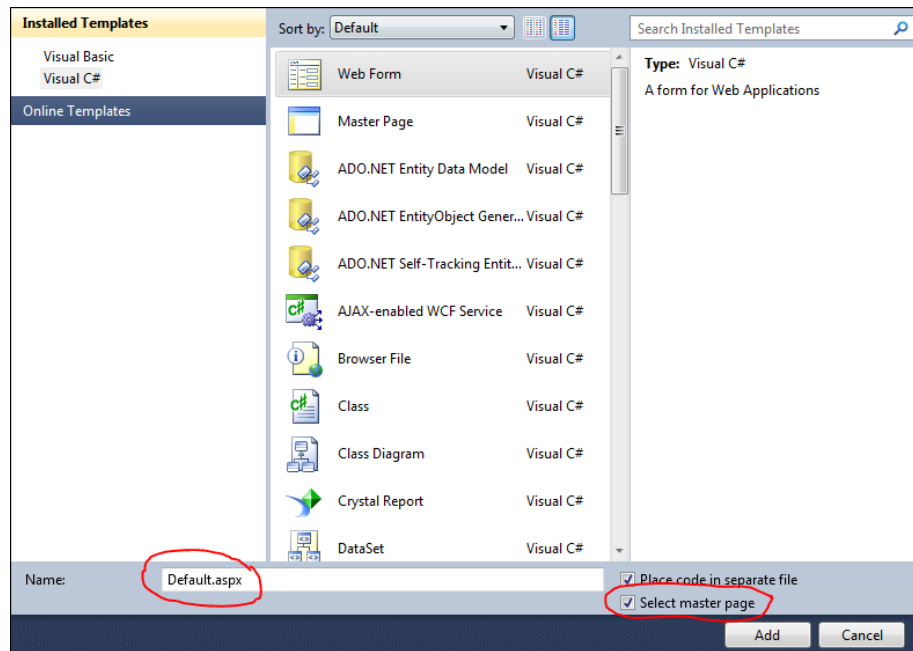
Halaman login ini disimpan pada direktori root dengan nama `Default.aspx`.



Gambar 33. `Default.aspx`, Halaman login.

Untuk membuat halaman login yang memanfaatkan `Public.master` sebagai Master Page dapat dilakukan dengan cara seperti berikut ini.

Pertama klik kanan pada `SPU.Web` dan pilih Add New Item, Pilih Web Form, beri nama filenya dengan nama `Default.aspx` kemudian centang Select master page.



Klik tombol Add, kemudian pilih Public.master pada folder Shared\MasterPages. Maka akan dapat dilihat kode halaman Default.aspx seperti berikut ini.

```
<%@ Page Title="" Language="C#"
    MasterPageFile="~/Shared/MasterPages/Public.master"
    AutoEventWireup="true"
    CodeFile="Default.aspx.cs" Inherits="_Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head Public" Runat="Server">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder_Public"
    Runat="Server">
</asp:Content>
```

Langkah pertama adalah membuat kotak warna coklat tua dengan ujung-ujung melengkung seperti pada gambar di bawah ini.



Gambar 34. Panel dengan ujung melengkung.

Untuk membuat kotak tersebut, yang pertama dilakukan adalah dengan menambahkan Panel seperti kode berikut ini :

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Shared/MasterPages/Public.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits=" Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head_Public" Runat="Server">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder_Public"
Runat="Server">
    <div id="PanelContainer">
        <asp:Panel ID="Panel_Login" CssClass="PanelLogin" runat="server">
            <div id="LoginContainer">
                // form login
            </div>
        </asp:Panel>
    </div>
</asp:Content>
```

Dan berikut kode-kode style yang ditambahkan pada file ui.css.

```
#PanelContainer
{
    margin-top: 73px;
    margin-left: 50px;
}

.PanelLogin
{
    width: 313px;
    height: 275px;
    background-color: #542f02;
    color: #ffffff;
}

#LoginContainer
{
    margin-left: 23px;
}
```

Hasilnya akan menjadi seperti pada gambar di bawah ini.

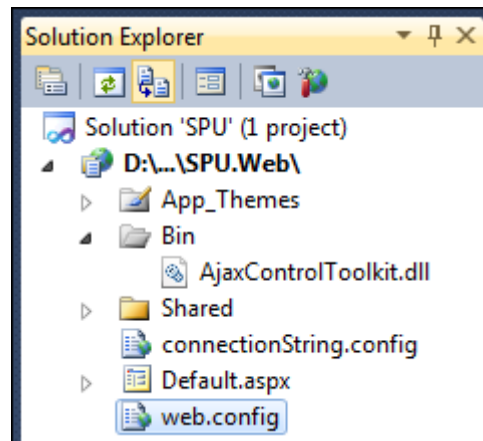


Gambar 35. Panel.

Untuk membuat agar ujung-ujung Panel tersebut menjadi melengkung seperti pada gambar 34 maka dapat memanfaatkan kontrol Rounded Corner yang terdapat pada AJAX Control Toolkit.

AJAX Control Toolkit dapat diunduh pada <http://ajaxcontroltoolkit.codeplex.com>, untuk keperluan Sistem Pengelolaan User sederhana ini diperlukan AJAX Control Toolkit untuk .NET Framework 4.0.

Setelah mengunduh AJAX Control Toolkit, salin file AjaxControlToolkit.dll ke folder Bin yang ada di dalam SPU.Web.



Gambar 36. Library AjaxControlToolkit pada folder Bin.

Bila pada SPU.Web belum ada folder Bin, maka bisa ditambahkan sendiri dengan cara klik kanan pada SPU.Web kemudian pilih Add ASP.NET Folder > Bin.

Selanjutnya menambahkan baris di bawah ini pada file Public.master.

```
<asp:ScriptManager ID="ScriptManager Public" runat="server">
</asp:ScriptManager>
```

Kemudian untuk ujung Panel menjadi melengkung maka tambahkan kontrol Rounded Corner pada file Default.aspx sehingga isi file ini menjadi seperti berikut.

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Shared/MasterPages/Public.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits="Default" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head Public" Runat="Server">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder Public"
Runat="Server">
    <div id="PanelContainer">
        <asp:Panel ID="Panel_Login" CssClass="PanelLogin" runat="server">
            <div id="LoginContainer">
                // form login
            </div>
        </asp:Panel>
        <ajaxToolkit:RoundedCornersExtender ID="rce"
            TargetControlID="Panel_Login"
            Radius="13"
            Corners="All"
            BorderColor="#b3b3b3"
            runat="server" />
    </div>
</asp:Content>
```

Pada kode di atas diketahui id dari Panel adalah Panel_Login, nilai tersebut akan digunakan sebagai nilai pada TargetControlID pada kontrol Rounded Corner. Hasilnya dapat dilihat seperti pada gambar 34.

Langkah selanjutnya adalah menambahkan form login di dalam Panel_Login. Pada kasus ini form login akan disimpan pada Web User Control.

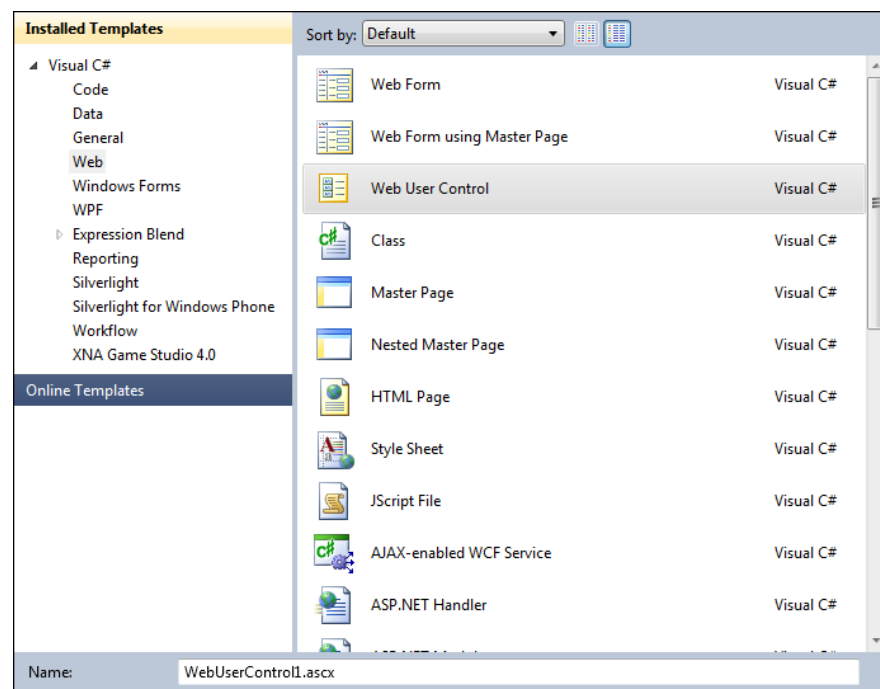
Sebelum membuat Web User Control, telah sering digunakan Control seperti :

- a. Button (asp:Button)
- b. Label (asp:Label)
- c. Login (asp:Login)
- d. Dan lain-lain.

Control-control itu dapat dipergunakan berulang-ulang dalam satu halaman atau halaman web lainnya. Web developer dapat membuat control yang dapat digunakan ulang, control yang dibuat sendiri oleh web developer disebut Web User Control.

Web User Control adalah file dengan extension ASCX dengan karakteristik seperti yang telah disebutkan pada paragraf di atas.

Sayangnya pada tidak dapat menambahkan Web User Control dengan VS2010 dan ketika menggunakan Web Site Project. Templat Web User Control baru dapat ditemui ketika menggunakan Web Application Project, seperti pada gambar di bawah ini.



Gambar 37. Web User Control pada Web Application Project.

Karena pada kasus ini bekerja di atas Web Site Project, maka ada beberapa cara untuk menambahkan Web User Control. Pertama adalah dengan membuat Web User Control pada Web Application Project kemudian menyalin file tersebut pada Web Site Project. Kedua adalah dengan membuat Web Form kemudian mengubahnya menjadi Web User Control.

Opsi kedua akan digunakan untuk membuat Web User Control. Web User Control ini akan disimpan pada folder Shared/Controls. Maka lakukan klik kanan pada folder tersebut kemudian pilih Add New Item kemudian pilih Web Form dan beri nama LoginForm.aspx.

Selanjutnya ganti nama file `LoginForm.aspx` menjadi `LoginForm.ascx`. selanjutnya pada file `LoginForm.ascx`, ubah baris berikut ini :

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="FormLogin.ascx.cs"
Inherits="Shared Controls FormLogin" %>
```

Menjadi :

```
<%@ Control Language="C#" AutoEventWireup="true" CodeFile="FormLogin.ascx.cs"
Inherits="Shared_Controls_FormLogin" %>
```

```
Ini Web User Control
```

Dari kedua baris di atas, dapat dilihat perubahan `Page` menjadi `Control`. Kemudian hapus baris lain sehingga isi file `LoginForm.ascx` ini hanya berisi seperti contoh di atas.

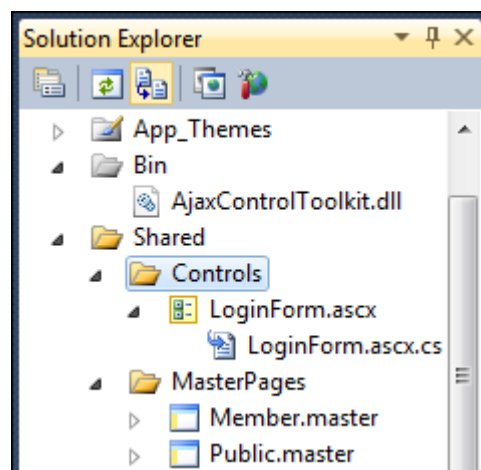
Perubahan selanjutnya akan dilakukan pada file `LoginForm.ascx.cs`, pada file ini akan diubah baris berikut ini :

```
public partial class Shared_Controls_LoginForm : System.Web.UI.Page
```

Menjadi :

```
public partial class Shared Controls LoginForm : System.Web.UI.UserControl
```

Berikut ini adalah lokasi file `FormLogin.ascx` yang dapat dilihat pada Solution Explorer.



Gambar 38. Lokasi file `LoginForm.ascx`.

Selanjutnya akan ditunjukkan cara menampilkan Web User Control ini pada file `Default.aspx` (Web Form). Berikut ini adalah isi file `Default.aspx` yang telah menggunakan file `LoginForm.ascx` sebagai Web User Control.

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Shared/MasterPages/Public.master" AutoEventWireup="true"
CodeFile="Default.aspx.cs" Inherits=" Default" %>

<%@ Register Src="~/Shared/Controls/LoginForm.ascx" TagName="Login"
TagPrefix="CommonControl" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head Public" Runat="Server">
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder_Public"
Runat="Server">
    <div id="PanelContainer">
        <asp:Panel ID="Panel_Login" CssClass="PanelLogin" runat="server">
```



```

        <div id="LoginContainer">
            <CommonControl:Login ID="Login Form" runat="server" />
        </div>
    </asp:Panel>
    <ajaxToolkit:RoundedCornersExtender ID="rce"
        TargetControlID="Panel_Login"
        Radius="13"
        Corners="All"
        BorderColor="#b3b3b3"
        runat="server" />
</div>
</asp:Content>

```

Pada baris dengan latar kuning yang pertama dapat dilihat bagaimana cara untuk mendaftarkan Web User Control agar dapat digunakan pada halaman Web Form. Selanjutnya pada baris dengan latar kuning yang kedua, dapat dilihat cara penulisan Web User Control agar dapat ditampilkan pada halaman Web Form saat 'dijalankan' pada web browser.

Dengan contoh kode di atas maka akan menghasilkan tampilan seperti berikut ini



Gambar 39. Web User Control berhasil ditampilkan.

Sekarang adalah mengubah LoginForm.ascx agar berfungsi sebagai form login seperti pada gambar 32.

Berikut ini adalah isi file LoginForm.ascx yang sudah berisi form login.

```

<%@ Control Language="C#" AutoEventWireup="true" CodeFile="LoginForm.ascx.cs"
Inherits="Shared_Controls_LoginForm" %>
<div id="LoginHeader">
    <table>
        <tr>
            <td>
                
            </td>
            <td id="LoginTitle">Sistem Pengelolaan User</td>
        </tr>
    </table>
</div>
<div id="LoginFormContainer">
    <asp:LoginView ID="LoginView Public" runat="server">
        <LoggedInTemplate>
            Selamat datang
            <strong><asp:LoginName ID="LoginName1" runat="server" /></strong>
            |
            <asp:HyperLink ID="HyperLink_Dashboard"
                NavigateUrl="~/Dashboard/Default.aspx"
                runat="server">Dashboard</asp:HyperLink>
            |
            <asp:LoginStatus ID="LoginStatus1" LoginText="Logout"
                LogoutAction="Redirect" LogoutPageUrl="~/Default.aspx"

```

```

        runat="server" />
</LoggedInTemplate>
<AnonymousTemplate>
    <asp:Login ID="Login Public"
        FailureText="Login failed. Please try again!"
        TitleText=""
        LoginButtonText="Sign In"
        PasswordLabelText="Password&nbsp;   :"
        UserNameLabelText="User&nbsp;   Name&nbsp;   :"
        TextBoxStyle-Width="175px"
        runat="server">
    </asp:Login>
</AnonymousTemplate>
</asp:LoginView>
</div>

```

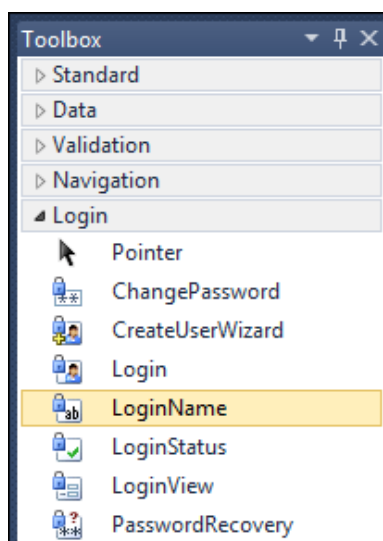
Hasilnya adalah seperti pada gambar di bawah ini.



The image shows a web login form with a dark blue header containing the INDC logo and the title 'Sistem Pengelolaan User'. Below the header, there are two text input fields labeled 'User Name :' and 'Password :'. A checkbox labeled 'Remember me next time.' is positioned below the password field. A blue 'Sign In' button is located at the bottom right of the form area.

Gambar 40. Web User Control untuk form login.

Pada contoh kode di atas, dapat dilihat terdapat control-control yang digunakan untuk membuat form login. Control-control tersebut dapat ditemui pada Toolbox Visual Studio 2010 pada bagian Login.



Gambar 41. Control Login pada Toolbox Visual Studio 2010.

Control-control yang digunakan pada kode di atas adalah :

- a. `LoginView`, control ini berfungsi untuk menampilkan area sebelum dan sesudah login. Apabila user belum login maka area yang ditampilkan adalah area pada bagian `AnonymouseTemplate`. Sedangkan area yang dapat dilihat setelah login adalah area pada bagian `LoggedInTemplate`.
- b. `LoginName`, control ini berfungsi untuk menampilkan nama user yang sedang login.
- c. `LoginStatus`, control ini berfungsi untuk menampilkan status login dan dapat juga berupa tombol untuk login dan logout. Apabila status user belum login maka control ini akan menampilkan link untuk login, sebaliknya akan menampilkan link untuk logout.
- d. `Login`, control ini berfungsi untuk menampilkan form login.

Di bawah ini adalah gambar antarmuka halaman sebelum dan sesudah login.



Gambar 42. Sebelum dan sesudah login.

Bila user belum login maka yang ditampilkan adalah form login, setelah user login maka dapat dilihat nama user yang sedang login dan link untuk logout.

Dari contoh ini dapat dilihat kemudahan yang diberikan oleh ASP.NET untuk membuat form login. Form login ini dapat berjalan lancar karena sebelumnya telah dilakukan konfigurasi untuk :

- a. Koneksi database pada file `web.config`.
- b. Konfigurasi pengaturan role (`roleManager`) pada file `web.config`.
- c. Konfigurasi pengaturan membership pada file `web.config`.

Apa yang telah dipelajari?

Pada sub bagian ini telah dipelajari beberapa fitur dan control. Berikut ini adalah hal-hal yang telah dipelajari :

- Mengetahui fungsi dan cara membuat Master Page.
- Mengetahui fungsi dan cara penerapan Theme.
- Mengetahui fungsi dan cara pembuatan Web User Control.
- Membuat form login dengan menggunakan control-control pada Toolbox Login di Visual Studio 2010.

Pembangunan Area Member

Area member adalah area yang hanya dapat dikunjungi oleh user yang sudah login atau yang sudah melewati proses autentifikasi.

Pada area ini user mempunyai fasilitas dasar seperti :

- Menu untuk memudahkan user mengakses fitur-fitur yang telah terdaftar.
- Fasilitas logout.

Membuat Master Page Member

Master Page untuk member ini akan diberi nama `Member.master`. Antarmuka akan mirip dengan `Public.master` kecuali ada penambahan area untuk menu dan fitur logout.

Antarmuka `Member.master` akan seperti gambar di bawah ini.



Gambar 43. Antarmuka Member.master.

Pada antarmuka dapat dilihat terdapat kotak yang ditandai dengan latarbelakang merah. Kotak merah yang berada disebelah kiri adalah untuk menampung menu sedangkan kotak merah disebelah kanan untuk menampung fitur Logout.

Berikut ini adalah isi dari file `Member.master`.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="Shared_MasterPages_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>Sistem Pengelolaan User</title>
    <meta http-equiv="X-UA-Compatible" content="IE=7" />
```

```

        <asp:ContentPlaceHolder id="Head_Member" runat="server">
        </asp:ContentPlaceHolder>
</head>
<body>
<body>
    <form id="form1" runat="server">
        <asp:ScriptManager ID="ScriptManager_Member" runat="server">
        </asp:ScriptManager>
        <div id="Header">
            <h1>INDC</h1>
            <div id="Logo"></div>
        </div>

        <div id="MenuContainer">
            <div id="MenuHolder"></div>

            <div id="LogoutHolder"></div>
        </div>

        <div id="MainContentMember">
            <asp:ContentPlaceHolder id="ContentPlaceHolder Member" runat="server">

            </asp:ContentPlaceHolder>
        </div>

        <div id="Footer"></div>
    </form>
</body>
</html>

```

Dari berikut ini adalah style yang ditambahkan pada file `masterpage.css`. Style-style ini berfungsi untuk mengatur tata letak dan antarmuka dari Master Page ini.

```

#MainContentMember
{
    _height:400px;
    min-height:400px;
    background-color:#de9439;
    border-top:solid 1px #b3b3b3;
    border-bottom:solid 1px #b3b3b3;
}

#MenuContainer
{
    width:100%;
    height:32px;
    border-top:solid 1px #b3b3b3;
}

#MenuHolder
{
    width:60%;
    height:30px;
    float:left;
}

#LogoutHolder
{
    width:35%;
    height:30px;
    float:right;
    text-align:right;
    padding-top:4px;
    padding-right:5px;
}

```

Membuat Fitur Logout

Selanjutnya adalah menambahkan fitur logout pada area yang telah disediakan. Fitur logout menggunakan control LoginStatus dengan penambahan penggunaan atribut. Sehingga tampilannya akan berbeda dengan output LoginStatus yang pada gambar 42.

Berikut ini adalah kode lengkap dari control LoginStatus yang digunakan.

```
<asp:LoginStatus ID="LoginStatus_Member"
LogoutImageUrl="~/App_Themes/SPU/images/icon-logout.png"
LogoutPageUrl="~/Default.aspx" LogoutAction="Refresh"
runat="server" />
```

Dari kode di atas dapat dilihat penggunaan atribut-atribut yang digunakan, yaitu :

- LogoutImageUrl, atribut ini berisi gambar yang akan dijadikan tombol logout.
- LogoutPageUrl, atribut ini berisi url atau alamat halaman login.
- LogoutAction, atribut ini berisi aksi yang akan dilakukan saat aksi logout dilakukan. Nilai dari atribut ini adalah RedirectToLoginPage, Redirect dan Refresh.

Hasilnya dapat dilihat pada gambar berikut ini.

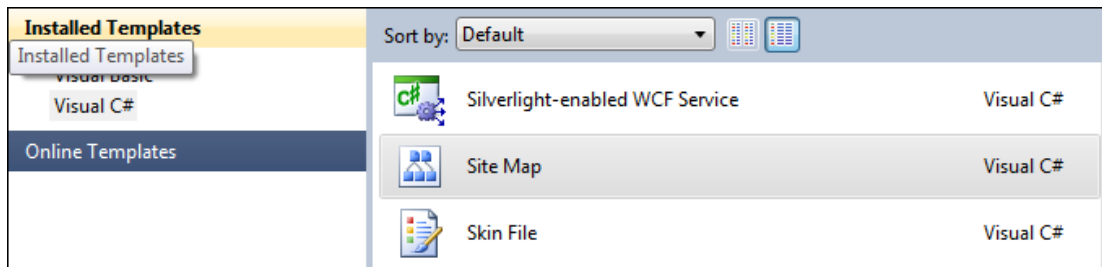


Gambar 44. Fitur Logout dengan menggunakan LoginStatus.

Menyiapkan Sitemap

Sitemap adalah media yang digunakan untuk menyimpan data halaman yang akan ditampilkan pada menu. Sitemap adalah file text yang menggunakan aturan penulisan seperti XML.

Berikut ini adalah cara yang dilakukan untuk membuat file Sitemap. Klik kanan pada Web Site Project dan pilih Add New Item. Pada template pilih Site Map, seperti pada gambar di bawah ini.



Gambar 45. Sitemap Template.

Berikan nama `Web.sitemap` untuk file tersebut, maka akan segera dapat dilihat isi file `Web.sitemap` seperti berikut ini.

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="" description="">
    <siteMapNode url="" title="" description="" />
    <siteMapNode url="" title="" description="" />
  </siteMapNode>
</siteMap>
```

File ini mempunyai format penulisan seperti format file XML dan cara penyimpanan data yang mengikuti aturan tree.

Berikut ini adalah contoh penyimpanan data dengan aturan tree.

	Dashboard
	Data Master
	Organisasi
	User Management
	Daftar User
	Member
	Profile

Dari data tersebut maka dapat dibuat isi dari file `Web.sitemap` seperti berikut ini :

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="Dashboard"
    description="Dashboard" roles="*">

    <siteMapNode url="" title="Data Master" description="Data Master"
      roles="Administrator">
      <siteMapNode url="~/DataMaster/OrganisasiList.aspx"
        title="Organisasi" description="Daftar Organisasi"
        roles="Administrator"/>
    </siteMapNode>

    <siteMapNode url="" title="User Management"
      description="User Management" roles="Administrator">
      <siteMapNode url="~/UserManagement/UserList.aspx" title="User"
        description="Daftar User" roles="Administrator"/>
    </siteMapNode>

    <siteMapNode url="" title="Member" description="Member" role="Member">
      <siteMapNode url="~/Dashboard/Profile.aspx" title="Profile"
        description="Profile" role="Member"/>
    </siteMapNode>
  </siteMapNode>
</siteMap>
```

Atau dengan melihat gambar di bawah ini.

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="Dashboard" description="Dashboard" roles="*">

    <siteMapNode url="" title="Data Master" description="Data Master" roles="Administrator">
      <siteMapNode url="~/DataMaster/OrganisasiList.aspx" title="Organisasi" description="Daftar Organisasi" roles="Administrator"/>
    </siteMapNode>

    <siteMapNode url="" title="User Management" description="User Management" roles="Administrator">
      <siteMapNode url="~/UserManagement/UserList.aspx" title="User" description="Daftar User" roles="Administrator"/>
    </siteMapNode>

    <siteMapNode url="" title="Member" description="Member" roles="Member">
      <siteMapNode url="~/Dashboard/Profile.aspx" title="Profile" description="Profile" roles="Member"/>
    </siteMapNode>

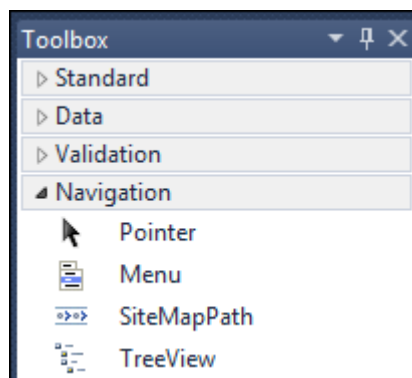
  </siteMapNode>
</siteMap>
```

Gambar 46. Web.sitemap.

Dari data atau gambar 46, dapat dilihat data struktur tree dari data pada Web.sitemap.

Mendaftarkan Sitemap

Sitemap yang telah dibuat dapat didaftarkan di dalam web.config. Dengan mendaftarkannya pada web.config maka akan didapat sebuah provider yang dapat digunakan oleh control-control yang terdapat pada group Navigation pada Toolbox.



Gambar 47. Control Navigation pada Toolbox.

Berikut ini adalah konfigurasi yang dapat ditambahkan pada web.config untuk mendaftarkan Web.sitemap.

```
<siteMap enabled="true">
  <providers>
    <add name="SPU_Sitemap"
        siteMapFile="~/Web.sitemap"
        type="System.Web.XmlSiteMapProvider"
        securityTrimmingEnabled="true"/>
  </providers>
</siteMap>
```

Baris-baris konfigurasi di atas dapat diletakkan di dalam elemen `<system.web>...</system.web>`. Atribut name adalah nama provider. Atribut siteMapFile berisi nama file sitemap yang ingin didaftarkan. Sedangkan atribut securityTrimmingEnabled berisi nilai true atau false. Bila atribut ini bernilai true, maka menu akan menampilkan data dari file sitemap ketika user sudah login.

Perlu diperhatikan, web developer dapat membuat lebih satu file sitemap, dengan nama yang berbeda-beda tentunya. Kemudian file-file tersebut dapat didaftarkan pada file web.config dengan aturan seperti yang dilihat di atas.

Berikut ini adalah contoh cara mendaftarkan beberapa file sitemap.


```

<siteMap enabled="true">
  <providers>
    <add name="SPU_Sitemap"
      siteMapFile="~/Web.sitemap"
      type="System.Web.XmlSiteMapProvider"
      securityTrimmingEnabled="true"/>
    <add name="SPU_Member"
      siteMapFile="~/Member.sitemap"
      type="System.Web.XmlSiteMapProvider"
      securityTrimmingEnabled="true"/>
    <add name="SPU_Admin"
      siteMapFile="~/Admin.sitemap"
      type="System.Web.XmlSiteMapProvider"
      securityTrimmingEnabled="true"/>
  </providers>
</siteMap>

```

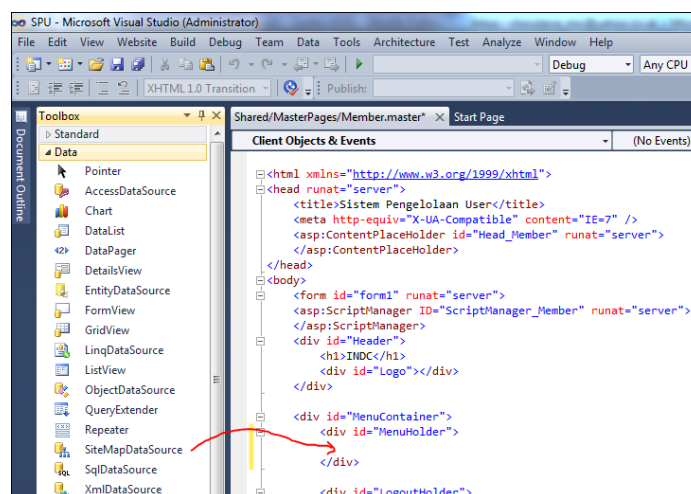
Dari contoh di atas yang membedakan masing-masing konfigurasi adalah nilai dari atribut `name` dan atribut `siteMapFile`.

Membuat Menu

ASP.NET mempunyai contro-control yang berfungsi untuk membuat dan menampilkan menu seperti yang dapat dilihat pada gambar 47.

Setelah persiapan dilakukan seperti yang telah dipaparkan di atas, maka langkah-langkah selanjutnya yang dapat dilakukan adalah seperti penjelasan berikut ini.

Seperti yang dapat dilihat pada gambar 43, maka dapat dilihat lokasi untuk meletakkan menu.



Gambar 48. Control `SiteMapDataSource` pada kelompok Data.

Langkah pertama adalah menambahkan control `SiteMapDataSource`. Control ini berfungsi sebagai datasource yang akan digunakan oleh control Menu. Sesuai namanya, `SiteMapDataSource` adalah control data source yang menggunakan file sitemap sebagai sumber data.

Berikut adalah contoh kode dari control `SiteMapDataSource` setelah ditambahkan pada file `Member.master`.

```

<asp:SiteMapDataSource ID="SiteMapDataSource_Member" runat="server" />

```

Setelah control ditambah, perlu ditambahkan atribut-atribut lain seperti berikut ini.

```
<asp:SiteMapDataSource ID="SiteMapDataSource Member"
    SiteMapProvider="SPU Sitemap"
    ShowStartingNode="false"
    runat="server" />
```

Terdapat atribut baru pada control di atas yaitu :

- a. SiteMapProvider, atribut ini berisi nama provider sitemap yang telah di daftarkan pada file web.config.

```
<siteMap enabled="true">
    <providers>
        <add name="SPU_Sitemap"
            siteMapFile="~/Web.sitemap"
            type="System.Web.XmlSiteMapProvider"
            securityTrimmingEnabled="true"/>
    </providers>
</siteMap>
```

Gambar 49. Sitemap Provider.

- b. ShowStartingNode, atribut ini bernilai false, artinya data yang ditampilkan pada menu tidak perlu dimulai dari awal yang ada pada file Web.sitemap. Artinya yang akan ditampilkan pada menu dimulai dari Data Master, User Management, Member tanpa Dashboard, karena Dashboard adalah titik awal (starting node) pada data.

Setelah itu tambahkan control Menu pada file Member.master. Berikut adalah contoh kodenya.

```
<asp:Menu ID="Menu Member"
    DataSourceID="SiteMapDataSource Member"
    Orientation="Horizontal"
    runat="server">
</asp:Menu>
```

Pada control di atas dapat dilihat dua atribut tambahan, yaitu :

- a. DataSourceId, atribut ini adalah untuk menentukan id dari datasource yang akan digunakan, dalam hal ini adalah id dari control SiteMapDataSource yaitu SiteMapDataSource_Member.
- b. Orientation, adalah orientasi dari menu.

Sebagai informasi, sebelum ASP.NET 4, control menu di-render sebagai sebuah table sehingga para web developer agak kesulitan untuk menyisipkan style CSS.

Pada ASP.NET 4, control menu di-render dengan menggunakan elemen unordered list dan list. Dengan menggunakan penggunaan semantik *markup* yang menggunakan elemen-elemen tersebut maka diharapkan control menu dapat memenuhi standar penulisan HTML dan CSS.

Setelah langkah-langkah pada bagian ini dilakukan, jangan untuk buru-buru menampilkan file ~/Dashboard/Default.aspx pada browser, karena dapat dipastikan menu tidak akan dapat dilihat. Menu hanya dapat dilihat kalau halaman tersebut di akses oleh user yang sudah login. Hal ini terjadi karena nilai atribut securityTrimmingEnabled pada provider sitemap yang ada pada file Web.config bernilai true, seperti yang dapat dilihat pada gambar 49.

Berikut ini adalah kode lengkap pada file Member.master.

```
<%@ Master Language="C#" AutoEventWireup="true" CodeFile="Member.master.cs"
Inherits="Shared MasterPages 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>Sistem Pengelolaan User</title>
    <meta http-equiv="X-UA-Compatible" content="IE=7" />
    <asp:ContentPlaceHolder id="Head_Member" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <asp:ScriptManager ID="ScriptManager_Member" runat="server">
    </asp:ScriptManager>
    <div id="Header">
        <h1>INDC</h1>
        <div id="Logo"></div>
    </div>

    <div id="MenuContainer">
        <div id="MenuHolder">
            <asp:SiteMapDataSource ID="SiteMapDataSource_Member"
                SiteMapProvider="SPU_Sitemap"
                ShowStartingNode="false"
                runat="server" />

            <asp:Menu ID="Menu_Member"
                DataSourceID="SiteMapDataSource_Member"
                Orientation="Horizontal"
                runat="server">
            </asp:Menu>
        </div>

        <div id="LogoutHolder">
            <asp:LoginStatus ID="LoginStatus_Member"
                LogoutImageUrl="~/App_Themes/SPU/images/icon-logout.png"
                LogoutPageUrl="~/Default.aspx"
                LogoutAction="Refresh"
                runat="server" />
        </div>
    </div>

    <div id="SitemapPathContainer">
        <asp:SiteMapPath ID="SiteMapPath_Member" runat="server">
        </asp:SiteMapPath>
    </div>

    <div id="MainContentMember">
        <asp:Panel ID="Panel_Member" CssClass="PanelMember" runat="server">
            <asp:ContentPlaceHolder id="ContentPlaceHolder_Member"
                runat="server">

                </asp:ContentPlaceHolder>
            </asp:Panel>
            <ajaxToolkit:RoundedCornersExtender ID="rce"
                TargetControlID="Panel_Member"
                Radius="13"
                Corners="All"
                BorderColor="#b3b3b3"
                runat="server" />
        </div>

        <div id="Footer"></div>
    </form>
</body>
</html>
```

Dan berikut ini adalah style-style yang akan digunakan untuk mengubah tampilan standar menu.

```
/* Menu Member */
#Menu Member .level1 li a
{
    color:#ffffff;
    width:130px;
    height:25px;
    padding-top:5px;
    padding-left:5px;
    background-color:#000000;
    border:solid 1px #b3b3b3;
    border-top:solid 0px #b3b3b3;
}

#Menu Member .level1 li a:hover
{
    color:#ffffff;
    width:130px;
    height:25px;
    padding-top:5px;
    padding-left:5px;
    background-color:#542f02;
    border:solid 1px #b3b3b3;
    border-top:solid 0px #b3b3b3;
    font-weight:bold;
}

#Menu_Member .level2 li a
{
    color:#ffffff;
    width:145px;
    height:25px;
    padding-top:5px;
    padding-left:5px;
    background-color:#000000;
    border:solid 1px #b3b3b3;
    border-top:solid 0px #b3b3b3;
}

#Menu_Member .level2 li a:hover
{
    color:#ffffff;
    width:145px;
    height:25px;
    padding-top:5px;
    padding-left:5px;
    background-color:#542f02;
    border:solid 1px #b3b3b3;
    border-top:solid 0px #b3b3b3;
    font-weight:bold;
}

.PanelMember
{
    width:97%;
    _height:313px;
    min-height:313px;
    background-color:#ffffff;
    margin:auto;
    padding:auto;
}

#ContentApplication
{
    padding-left:5px;
    padding-right:5px;
}
```

Dan hasilnya akan seperti pada gambar di bawah ini.



Gambar 50. Menu dengan style CSS.

Hak Akses

Langkah terakhir adalah memberikan pengaturan akses pada file `Member.master`, tujuan pengaturan akses ini adalah agar halaman-halaman web form yang menggunakan file `Member.master` hanya bisa diakses ketika user sudah login.

Untuk melakukan hal ini cukup dengan menambahkan baris berikut pada method `Page_Load` yang ada pada file `Member.master.cs`.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Shared_MasterPages_Member : System.Web.UI.MasterPage
{
    protected void Page_Load(object sender, EventArgs e)
    {
        bool isLogin = HttpContext.Current.User.Identity.IsAuthenticated;

        if (!isLogin)
        {
            Response.Redirect("~/Default.aspx");
        }
    }
}
```

Baris di bawah ini berfungsi untuk mengetahui status login, bila user yang mengakses halaman belum login maka nilai variable `isLogin` akan bernilai `false`, dan sebaliknya akan bernilai `true`.

```
bool isLogin = HttpContext.Current.User.Identity.IsAuthenticated;
```

Bila nilai variable adalah `false`, maka user dianggap belum login dan user akan diantarkan ke halaman `Default.aspx` yang berada di root web.

```
if (!isLogin)
{
    Response.Redirect("~/Default.aspx");
}
```

Menambahkan SiteMapPath

Control `SiteMapPath` adalah salah satu control yang terdapat pada kelompok `Navigation`. Control ini berfungsi untuk menampilkan hirarki sitemap yang ada di bawah control `Menu`, seperti gambar berikut ini.



Gambar 51. SiteMapPath.

Bila melihat url yang ada pada address bar maka dapat diketahui halaman yang ditampilkan oleh web browser adalah `OrganisasiList.aspx`.

```
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="~/Dashboard/Default.aspx" title="Dashboard" description="Dashboard" roles="Member">
    <siteMapNode url="" title="Data Master" description="Data Master" roles="Member">
      <siteMapNode url="~/DataMaster/OrganisasiList.aspx" title="Organisasi" description="Organisasi" roles="Member">
      </siteMapNode>
    </siteMapNode>
    <siteMapNode url="" title="User Management" description="User Management" roles="Member">
      <siteMapNode url="~/UserManagement/UserList.aspx" title="User" description="User" roles="Member">
      </siteMapNode>
    </siteMapNode>
    <siteMapNode url="" title="Member" description="Member" roles="Member">
      <siteMapNode url="~/Dashboard/Profile.aspx" title="Profile" description="Profile" roles="Member">
      </siteMapNode>
    </siteMapNode>
  </siteMapNode>
</siteMap>
```

Gambar 52. Hirarki pada file `Web.sitemap`.

Dengan memperhatikan isi file `Web.sitemap` di atas, maka dapat diketahui file `OrganisasiList.aspx` akan ditemukan dengan hirarki `Dashboard > Data Master > Organisasi`, yang mana masing-masing nilai dari hirarki tersebut adalah nilai dari atribut `title` pada setiap atribut `siteMapNode`.

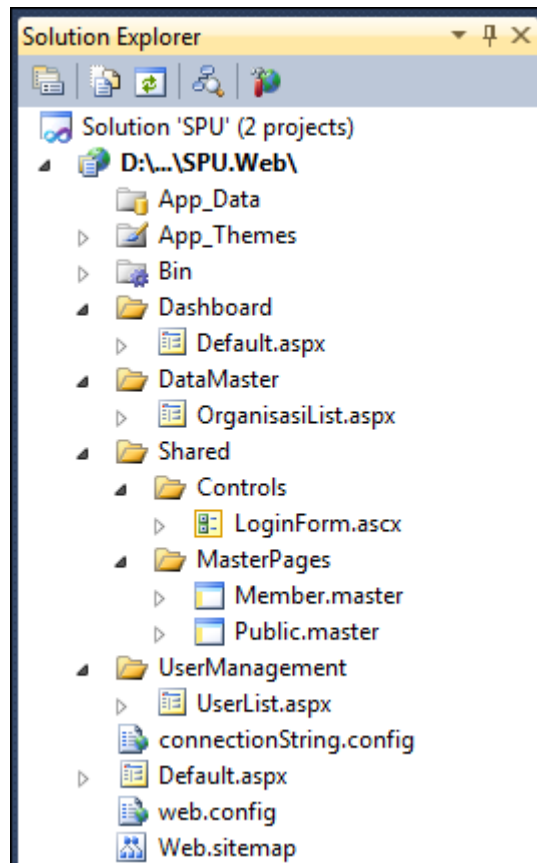
Apa yang telah dipelajari ?

Pada sub bagian ini telah dipelajari beberapa fitur dan control. Berikut ini adalah hal-hal yang telah dipelajari :

- a. Fitur dan control Login.
- b. Membuat dan menggunakan Sitemap.
- c. Control SiteMapDataSource.
- d. Fitur dan control Navigation seperti Menu dan SitemapPath.
- e. Hak akses secara programatically.

Struktur Web Site SPU.Web

Berikut ini adalah screen shot Web Site SPU.Web untuk memperlihatkan struktur file dan folder secara lengkap.



Gambar 53. Struktur file dan folder pada SPU.Web.

Struktur folder `Dashboard`, `DataMaster` dan `UserManagement` beserta file-filenya mengikuti struktur yang telah dibuat pada file `Web.sitemap`.

Data Access Layer

Seperti yang telah dijelaskan pada Bab 2 Persiapan, bahwa terdapat database `SPU_Data` yang akan menyimpan tabel-tabel yang berhubungan dengan aplikasi. Dalam kasus ini adalah tabel `Organisasi`.

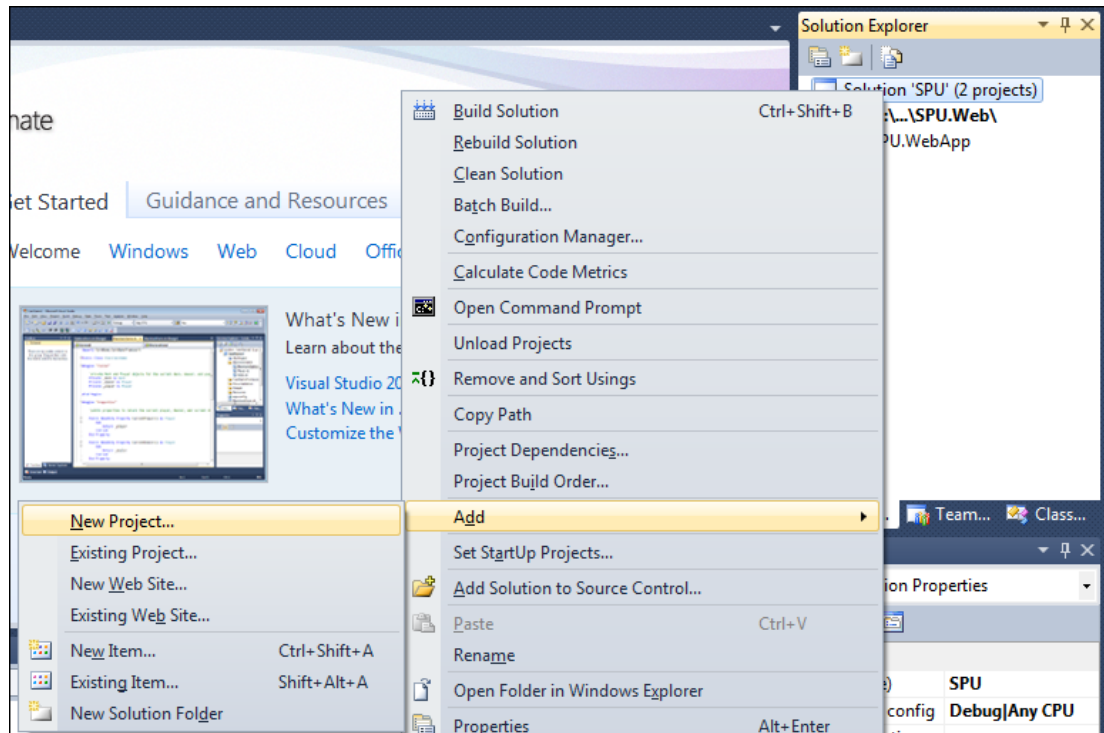
Untuk memudahkan developer bekerja dengan database ini maka perlu dibuat Data Access Layer sederhana dengan menggunakan LINQ to SQL.

Untuk membuat Data Access Layer sederhana ini perlu ditambahkan dua project baru pada SPU solution, kedua project tersebut adalah :

- a. SPU.Entities.
- b. SPU.DataAccess.

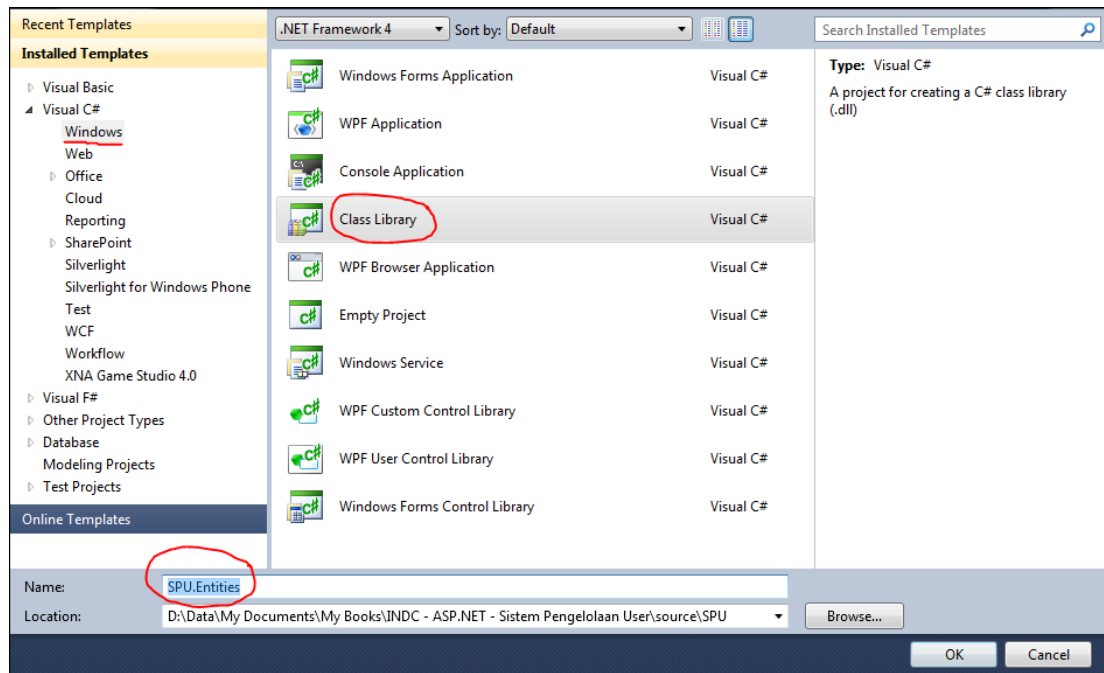
SPU.Entities

Berikut ini adalah langkah membuat project `SPU.Entities`, klik kanan pada `SPU` solution yang ada pada Solution Explorer dan pilih `Add > New Project`.



Gambar 54. Menambah project baru pada solution.

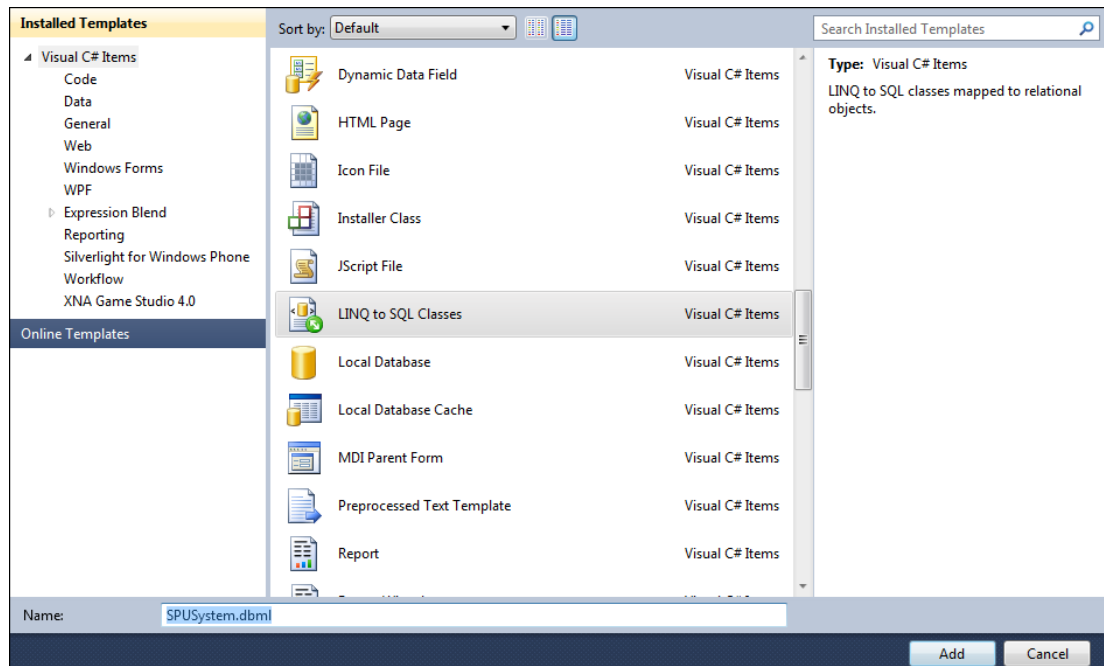
Pada jendela `Add New Project`, pilih `Visual C# > Windows > Class Library` dan beri nama project seperti pada gambar di bawah ini.



Gambar 55. SPU.Entities.

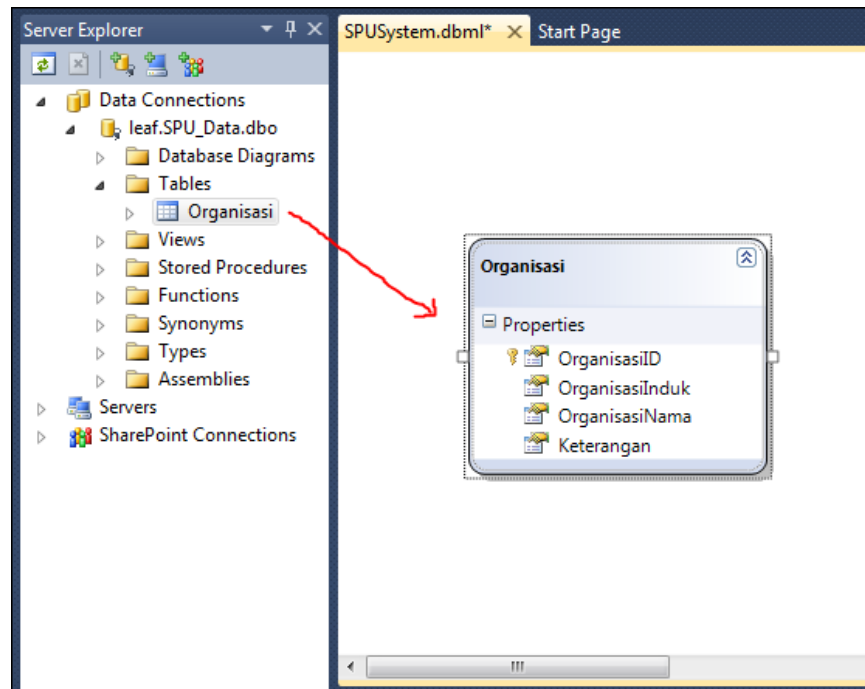
Dan kemudian klik tombol OK.

Hapus file Class1.cs yang ada pada project ini. Langkah selanjutnya adalah menambahkan LINQ to SQL Classes dengan nama SPUSystem.dbml.



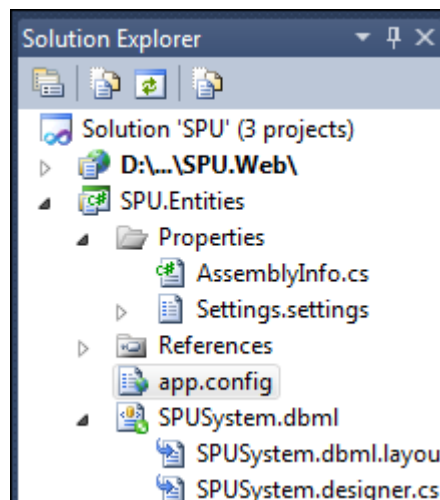
Gambar 56. Menambahkan LINQ to SQL Classes, SPUSystem.dbml.

Selanjutnya adalah membuat class pada `SPUSystem.dbml` dengan cara men-drag tabel-tabel yang ada pada database `SPU_Data` ke dalam `SPUSystem.dbml`.



Gambar 57. Membuat class LINQ to SQL.

Ketika class dibuat dengan cara seperti yang telah diterangkan di atas, maka secara otomatis akan ditambahkan file konfigurasi `app.config` dan `Settings.settings` yang menyimpan connection string atau cara untuk melakukan koneksi ke database `SPU_Data`.



Gambar 58. File konfigurasi `app.config` dan `Settings.settings`.

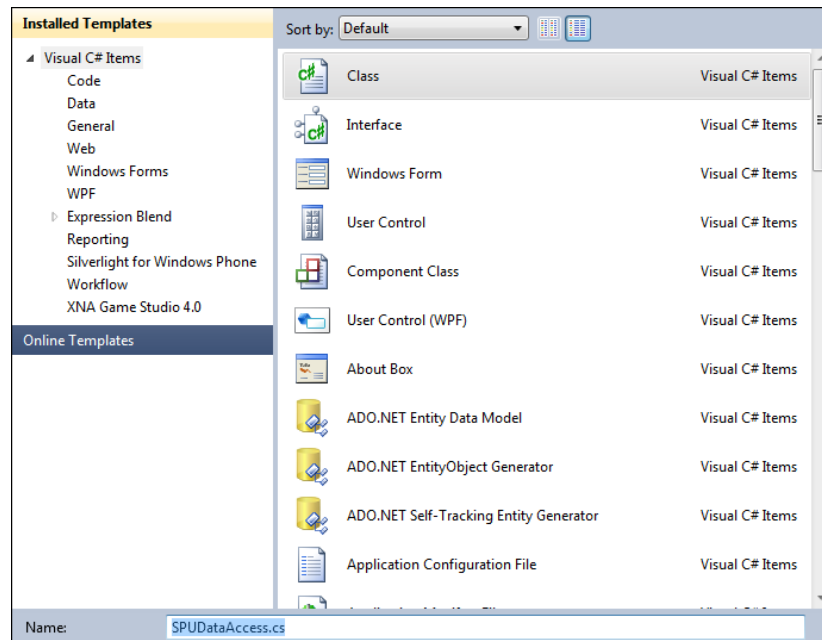
Penyimpanan connection string pada kedua file ini akan menyulitkan saat proses deployment, untuk mempermudah maka perlu dibuat agar cara koneksi dari class ini tetap menggunakan connection string yang ada pada Web Site Project, yaitu pada file `connectionString.config`.

Untuk tujuan tersebut maka dibuat project baru, yaitu `SPU.DataAccess`.

SPU.DataAccess

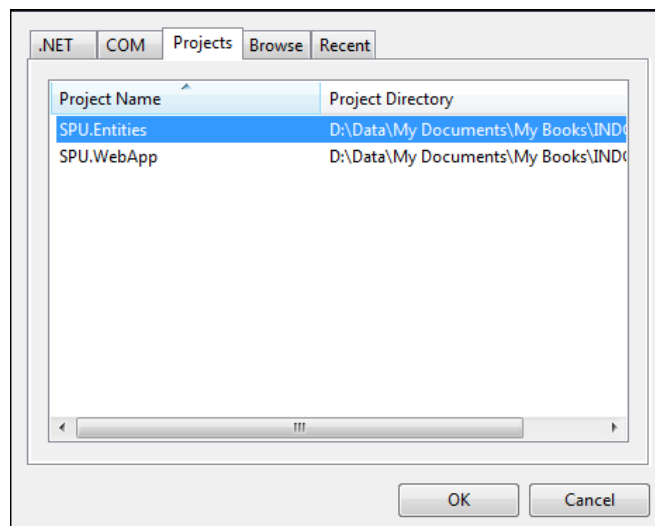
SPU.DataAccess adalah project yang menggunakan template project class library seperti halnya SPU.Entities.

Tambahkan class dengan nama SPUDataAccess.cs pada project ini.



Gambar 59. Class SPUDataAccess.cs.

Selanjutnya adalah menambahkan project SPU.Entities sebagai reference dari project SPU.DataAccess dengan cara klik kanan pada project SPU.DataAccess dan pilih Add Reference. Klik tab Projects dan pilih SPU.Entities.

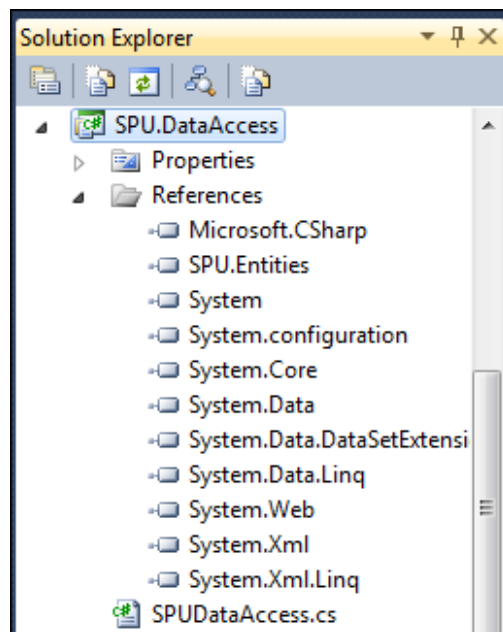


Gambar 60. Jendela Add Reference

Setelah itu tambahkan pula item-item berikut sebagai reference :

- System.Web.
- System.Configuration.
- System.Data.Linq.

Berikut ini adalah daftar reference lengkap pada project `SPU.DataAccess`.



Gambar 61. Daftar reference lengkap.

Dan berikut ini adalah isi file `SPUDataAccess.cs`.

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

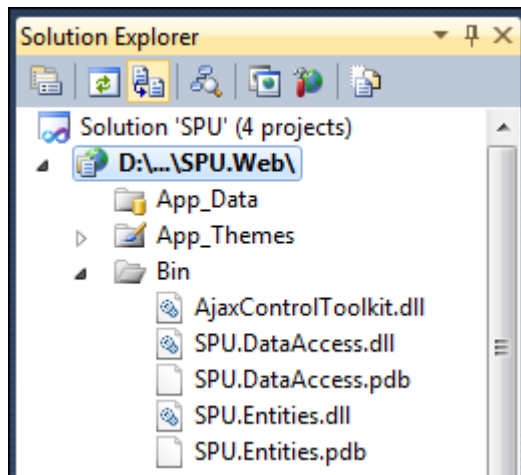
using System.Web.Configuration;

using SPU.Entities;

namespace SPU.DataAccess
{
    public class SPUDataAccess
    {
        public SPUSystemDataContext SPUDataContext = new
        SPUSystemDataContext(WebConfigurationManager.ConnectionStrings["SPU_Data_ConnString"].ToString());
    }
}
```

Dari kode di atas dapat dilihat objek `SPUDataContext` merupakan instansiasi dari class `SPUSystemDataContext` dengan menggunakan konstruktor yang menggunakan parameter dengan nilai connection string. Pada kode di atas dapat diketahui connection string yang digunakan adalah yang disimpan pada file `connectionString.config` pada `SPU.Web`.

Selanjutnya adalah menambahkan project `SPU.DataAccess` sebagai reference pada `SPU.Web` dengan cara klik kanan pada `SPU.Web` dan pilih Add Reference kemudian klik tab Projects dan pilih `SPU.DataAccess`. Hasilnya akan terlihat pada folder Bin seperti pada gambar berikut ini.



Gambar 62. Daftar reference pada project SPU.Web.

Selain itu juga tambahkan `System.Data.Linq` sebagai reference pada `SPU.Web`.

Dan fondasi web site ini telah selesai dibuat dan siap untuk digunakan.

Create, Retrieve, Update dan Delete

Berikut ini adalah contoh penggunaan Data Access Layer yang telah dibuat. Contoh-contoh yang akan diberikan adalah untuk proses :

- Create atau tambah data.
- Retrieve atau mengambil data.
- Update atau modifikasi data.
- Delete atau menghapus data.

Sebelum melakukan aksi-aksi di atas, pastikan namespace berikut ini telah ditambahkan, yaitu :

- `System.Data.`
- `System.Data.Linq.`
- `SPU.DataAccess.`
- `SPU.Entities.`

Create

Berikut adalah contoh untuk menambah data.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

using System.Data;
using System.Data.Linq;

using SPU.DataAccess;
using SPU.Entities;

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

```

SPUDataAccess db = new SPUDataAccess();
Organisasi org = new Organisasi();

org.OrganisasiID = "SI";
org.OrganisasiInduk = "ILKOM";
org.OrganisasiNama = "Sistem Informasi";
org.Keterangan = "Sistem Informasi";

db.SPUDataContext.Organisasis.InsertOnSubmit(org);
db.SPUDataContext.SubmitChanges();
}
}

```

Retrieve

Berikut contoh untuk mengambil data dan menampilkannya pada control `DropDownList` sebagai contoh. Contoh di bawah ini akan mengembalikan akan seluruh data yang disimpan pada tabel `Organisasi`.

```

SPUDataAccess db = new SPUDataAccess();
var data = db.SPUDataContext.Organisasis.Select(p => p);

DropDownList1.DataSource = data;
DropDownList1.DataValueField = "OrganisasiID";
DropDownList1.DataTextField = "OrganisasiNama";
DropDownList1.DataBind();

```

Bila ingin mengambil satu record pada data `Organisasi` dan ingin ditampilkan pada control `Label` maka dapat digunakan dengan cara berikut ini.

```

SPUDataAccess db = new SPUDataAccess();
Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals("ILKOM")).Single<Organisasi>();

Label1.Text = org.OrganisasiID + " " + org.OrganisasiNama;

```

Update

Untuk mengupdate maka dapat dilakukan dengan menggunakan kode berikut ini.

```

SPUDataAccess db = new SPUDataAccess();
Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals("ILKOM")).Single<Organisasi>();

org.OrganisasiNama = "Teknik Informatika";
org.Keterangan = "Teknik Informatika";

db.SPUDataContext.SubmitChanges();

```

Delete

Sedangkan untuk menghapus data maka dapat digunakan kode berikut ini.

```

SPUDataAccess db = new SPUDataAccess();
Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(node.Value)).Single<Organisasi>();

db.SPUDataContext.Organisasis.DeleteOnSubmit(org);
db.SPUDataContext.SubmitChanges();

```

Apa yang telah dipelajari?

Pada sub bagian ini telah dipelajari cara :

- a. Menambahkan project pada solution.
- b. Membuat project menjadi reference project lain.
- c. Menambahkan reference.
- d. Membuat class LINQ to SQL.
- e. Menggunakan Data Access Layer untuk proses CRUD.

Data Master

Data Master, adalah fitur yang semestinya ada di setiap sistem yang dibuat, diminta atau tidak diminta oleh client atau customer.

Pada bagian ini akan dipaparkan cara untuk membangun sub sistem Data Master untuk mengelola data Organisasi.

Design dan Data Tabel Organisasi

Seperti yang telah diperlihatkan pada gambar 25, maka dapat diketahui tabel *Organisasi* mempunyai field-field sebagai berikut :

- OrganisasiID*.
- OrganisasiInduk*, field ini akan berisi nilai induk dari organisasi.
- OrganisasiNama*.
- Keterangan.

Sesuai dengan tabel yang telah dibuat, maka data *Organisasi* akan membentuk hirarki. Jika data dari tabel ini adalah seperti berikut ini.

LEAF.SPU_Data - dbo.Organisasi		LEAF.SPU_Data - dbo.Organisasi		
	OrganisasiID	OrganisasiInduk	OrganisasiNama	Keterangan
	UNLAM	NULL	Universitas Lambung Mangkurat	Universitas Lambung Mangkurat
	FMIPA	UNLAM	Fakultas Matematika & Ilmu Pengetahuan Alam	NULL
	ILKOM	FMIPA	Ilmu Komputer	Ilmu Komputer
	FA	FMIPA	Farmasi	Farmasi
	FI	FMIPA	Fisika	Fisika
	BI	FMIPA	Biologi	Biologi

Gambar 63. Data tabel Organisasi.

Maka akan didapat hirarki seperti berikut.

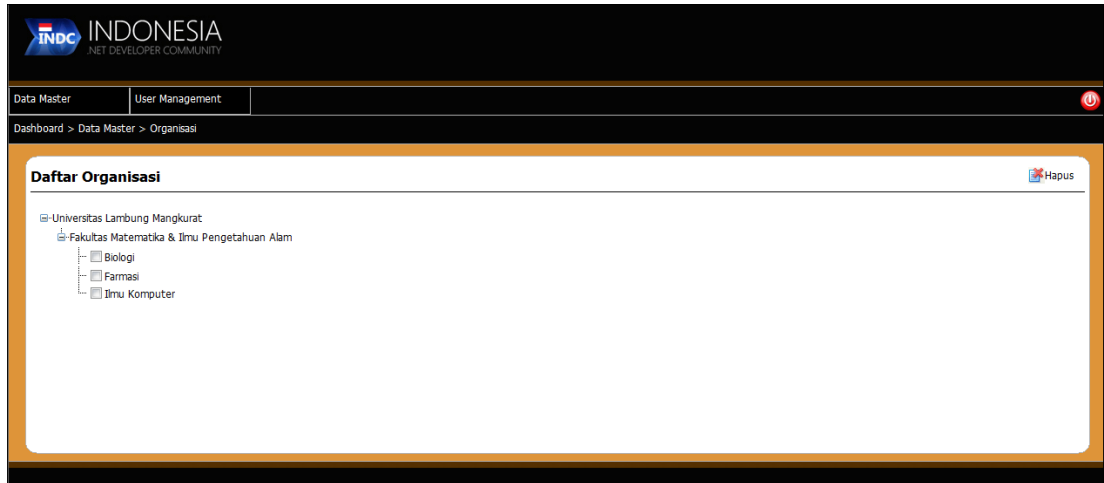
Universitas Lambung Mangkurat		
	Fakultas Matematika dan Ilmu Pengetahuan Alam	
		Ilmu Komputer
		Farmasi
		Fisika
		Biologi

Mengelola Organisasi

Dari data tersebut maka akan dibuat halaman CRUD (Create-Retrieve-Update-Delete) untuk mengelola data Organisasi.

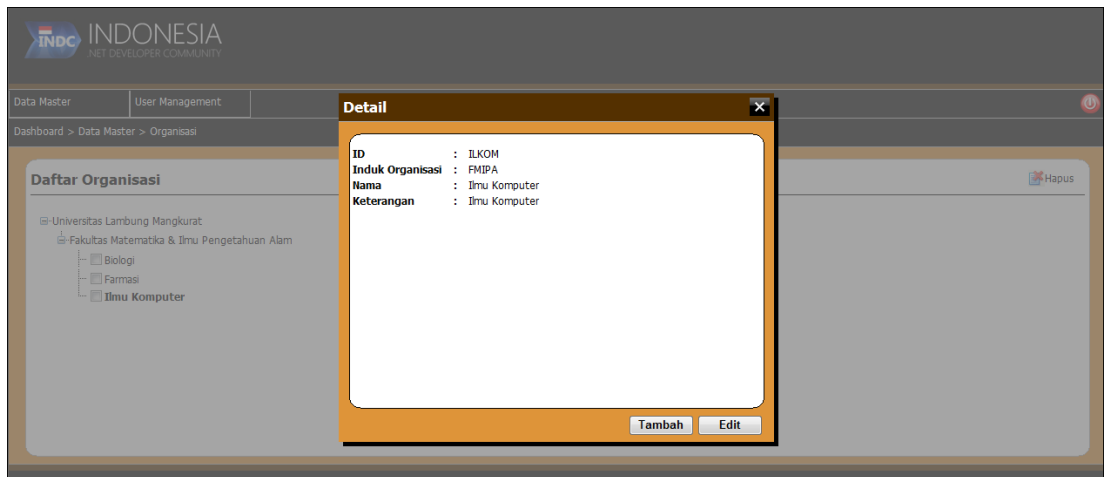
Sebelum memulai membuat halaman untuk mengelola data Organisasi, maka terlebih dahulu akan diperlihatkan bagaimana halaman ini akan bekerja.

Pertama adalah halaman untuk menampilkan data (Retrieve).



Gambar 64. Menampilkan data.

Pada halaman di atas dapat dilihat data ditampilkan secara hirarki, berbentuk tree. Untuk melihat detail data dari item yang ada pada hirarki tree tersebut, cukup dengan mengklik salah satu item maka akan dapat dilihat detail data ditampilkan seperti pada gambar berikut ini.

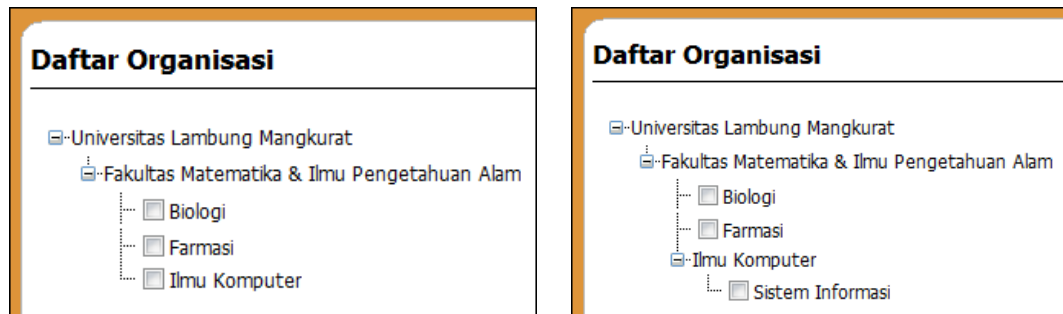


Gambar 65. Detail Data.

Detail data akan ditampilkan pada modal window. Pada modal window ini, user dapat melakukan dua aksi, yaitu :

- Mengedit data yang dipilih/yang ditampilkan pada modal window, caranya cukup mengklik tombol **Edit**.
- Menambah data anak pada item yang dipilih. Artinya data yang ditambahkan akan berada di bawah item yang ditampilkan pada modal windows.

Misalnya data yang sudah ada seperti pada gambar di bawah ini (gambar kiri). Pertama akan dipilih item Ilmu Komputer, setelah detail ditampilkan dan data ditambah maka data yang ditambahkan tersebut akan menempati posisi di bawah item Ilmu Komputer (gambar kanan).



Gambar 66. Menambah data organisasi.

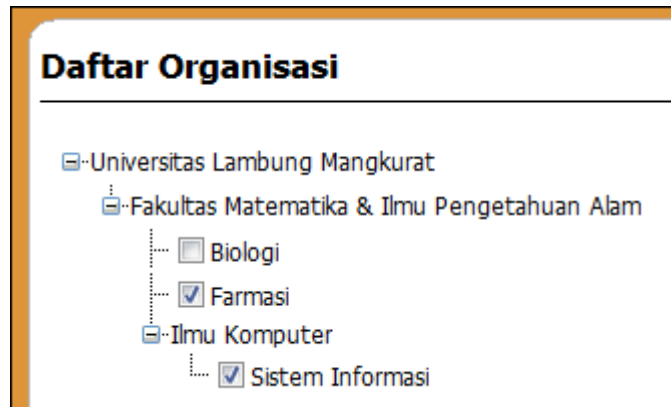
Berikut ini adalah form yang akan ditampilkan saat proses tambah data.

Gambar 67. Tambah data.

Dan berikut adalah antarmuka yang dapat dilihat ketika tombol `Edit` ditekan.

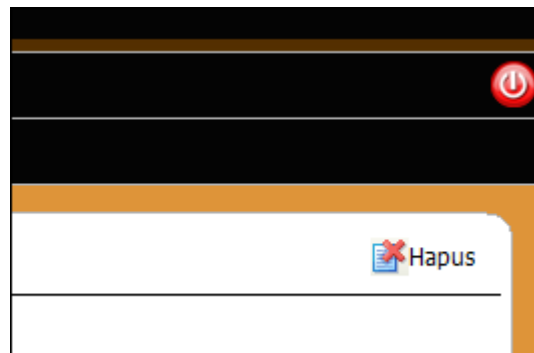
Gambar 68. Form edit data.

Sedangkan untuk menghapus data, terlebih dahulu pilih item yang akan dipilih dengan cara mencentang *Checkbox* pada item yang ingin dihapus, seperti yang dapat dilihat pada gambar berikut.



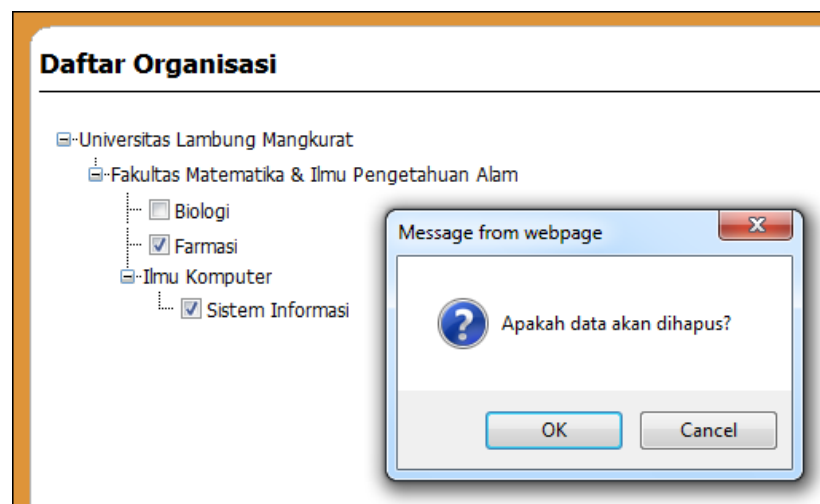
Gambar 69. Memilih item yang akan dihapus.

Kemudian klik tombol *Hapus* yang ada pada sisi kanan.



Gambar 70. Tombol Hapus.

Ketika tombol *Hapus* ditekan maka akan terdapat konfirmasi seperti berikut ini.



Gambar 71. Konfirmasi sebelum menghapus.

Nama halaman yang menampilkan yang telah diperlihatkan di atas adalah `OrganisasiList.aspx` dan `OrganisasiList.aspx.cs` sebagai code file.

Berikut ini adalah isi dari file OrganisasiList.aspx

```
<%@ Page Title="Daftar Organisasi" Language="C#"
MasterPageFile="~/Shared/MasterPages/Member.master" AutoEventWireup="true"
CodeFile="OrganisasiList.aspx.cs" Inherits="DataMaster OrganisasiList" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head_Member" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder Member"
Runat="Server">
    <!-- Main View - Start -->
    <div id="ApplicationContainer">

        <!-- Title & Buttons - Start -->
        <div id="TitleToolContainer">
            <div id="Title">
                <asp:Label ID="Label Title" CssClass="TextTitle" runat="server"
                    Text="Label"></asp:Label>
            </div>
            <asp:UpdatePanel ID="UpdatePanel ActionTool"
                UpdateMode="Conditional" runat="server">
                <ContentTemplate>
                    <div id="ActionTool">
                        <div class="ActionToolItemText">
                            <asp:Label ID="Label_Delete" runat="server"
                                AssociatedControlID="ImageButton Delete"
                                Text="Hapus"></asp:Label>
                        </div>
                        <div class="ActionToolItem">
                            <asp:ImageButton ID="ImageButton_Delete"
                                ImageUrl="~/App Themes/SPU/icons/icon delete.gif"
                                runat="server"
                                onclick="ImageButton Delete Click" />
                            <ajaxToolkit:ConfirmButtonExtender
                                ID="CBE_ConfirmationDelete"
                                ConfirmText="Apakah data akan dihapus?"
                                runat="server"
                                TargetControlID="ImageButton Delete" />
                        </div>
                    </div>
                    <div style="visibility:hidden">
                        <asp:LinkButton ID="LinkButton1" runat="server">
                            LinkButton</asp:LinkButton>
                            <ajaxToolkit:ModalPopupExtender ID="MPE"
                                runat="server"
                                TargetControlID="LinkButton1"
                                PopupControlID="Panel_Modal"
                                BackgroundCssClass="modalBackground"
                                DropShadow="true"
                                PopupDragHandleControlID="Panel Modal Header" />
                        </div>
                    </ContentTemplate>
                </asp:UpdatePanel>
            </div>
            <!-- Title & Buttons - End -->

            <div class="Separator"></div>

            <!-- Main Data - Start -->
            <div id="ContentContainer">
                <asp:UpdatePanel ID="UpdatePanel Main" UpdateMode="Conditional"
                    runat="server">
                    <ContentTemplate>
                        <asp:TreeView ID="TreeView_Main" ShowLines="true"
                            ShowCheckBoxes="Leaf"
                            runat="server"
                            onselectednodechanged="TreeView Main SelectedNodeChanged">
                            <SelectedNodeStyle Font-Bold="true" />
                        </asp:TreeView>
                    </ContentTemplate>
                </asp:UpdatePanel>
            </div>
            <!-- Main Data - End -->
        </div>
    <!-- Main View - End -->
</div>
```

```

<%-- Modal Windows - Start --%>
<asp:Panel ID="Panel Modal" CssClass="PanelModal" runat="server">
    <asp:UpdatePanel ID="UpdatePanel Panel Modal"
        UpdateMode="Conditional" runat="server">
        <ContentTemplate>
            <div class="PanelModalHeader">
                <%-- Header Modal Windows - Start --%>
                <asp:Panel ID="Panel Modal Header" runat="server">
                    <div class="TextTitleModalHeader">
                        <asp:Label ID="Label Modal Title" runat="server"
                            Text="Detail"></asp:Label>
                    </div>
                    <div class="ActionCloseModalHeader">
                        <asp:ImageButton ID="ImageButton Close"
                            ImageUrl="~/App Themes/SPU/icons/icon modal button close.gif"
                            AlternateText="Tutup" runat="server"
                            onclick="ImageButton_Close_Click" />
                    </div>
                </asp:Panel>
                <%-- Header Modal Windows - End --%>
            </div>
            <div class="Separator"></div>
            <div class="PanelModalContentContainer">
                <asp:Panel ID="Panel Modal Content" CssClass="PanelModalContent"
                    runat="server">
                    <%-- Message, Detail & Form - Start --%>
                    <asp:MultiView ID="MultiView Content" runat="server">
                        <asp:View ID="View Detail" runat="server">
                            <table>
                                <tr>
                                    <td>
                                        <asp:Label ID="Label Detail ID"
                                            SkinID="LabelDetail" runat="server"
                                            Text="ID"></asp:Label>
                                    </td>
                                    <td class="SeparatorH"></td>
                                    <td>
                                        <asp:Label ID="Label Value ID"
                                            runat="server"
                                            Text="Label"></asp:Label>
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                        <asp:Label ID="Label_Detail_Induk"
                                            SkinID="LabelDetail" runat="server"
                                            Text="Induk Organisasi"></asp:Label>
                                    </td>
                                    <td class="SeparatorH"></td>
                                    <td>
                                        <asp:Label ID="Label_Value_Induk"
                                            runat="server"
                                            Text="Label"></asp:Label>
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                        <asp:Label ID="Label_Detail_Nama"
                                            SkinID="LabelDetail" runat="server"
                                            Text="Nama"></asp:Label>
                                    </td>
                                    <td class="SeparatorH"></td>
                                    <td>
                                        <asp:Label ID="Label_Value_Nama"
                                            runat="server"
                                            Text="Label"></asp:Label>
                                    </td>
                                </tr>
                                <tr>
                                    <td>
                                        <asp:Label ID="Label_Detail_Keterangan"
                                            SkinID="LabelDetail" runat="server"
                                            Text="Keterangan"></asp:Label>
                                    </td>
                                    <td class="SeparatorH"></td>
                                </tr>
                            </table>
                        </asp:View>
                    </asp:MultiView>
                </asp:Panel>
            </div>
        </ContentTemplate>
    </asp:UpdatePanel>
</asp:Panel>

```

```

        <td>
            <asp:Label ID="Label Value Keterangan"
                runat="server"
                Text="Label"></asp:Label>
        </td>
    </tr>
</table>
</asp:View>
<asp:View ID="View Form" runat="server">
    <table>
        <tr>
            <td colspan="3">
                <asp:ValidationSummary
                    ID="ValidationSummary Form"
                    ValidationGroup="Default"
                    runat="server" />
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label Form ID"
                    SkinID="LabelDetail" runat="server"
                    Text="ID"></asp:Label>
            </td>
            <td class="SeparatorH"></td>
            <td>
                <asp:TextBox ID="TextBox Form ID"
                    SkinID="TextboxForm"
                    ValidationGroup="Default"
                    runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="RFV Form ID"
                    ValidationGroup="Default"
                    ControlToValidate="TextBox Form ID"
                    runat="server"
                    ErrorMessage="ID harus diisi."
                    ToolTip="ID harus diisi.">
                *</asp:RequiredFieldValidator>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label Form Induk"
                    SkinID="LabelDetail" runat="server"
                    Text="Induk Organisasi"></asp:Label>
            </td>
            <td class="SeparatorH"></td>
            <td>
                <asp:TextBox ID="TextBox Form Induk"
                    SkinID="TextboxForm"
                    runat="server"></asp:TextBox>
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label ID="Label Form Nama"
                    SkinID="LabelDetail" runat="server"
                    Text="Nama"></asp:Label>
            </td>
            <td class="SeparatorH"></td>
            <td>
                <asp:TextBox ID="TextBox Form Nama"
                    SkinID="TextboxForm"
                    ValidationGroup="Default"
                    runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator
                    ID="RFV Form Nama"
                    ValidationGroup="Default"
                    ControlToValidate="TextBox Form Nama"
                    runat="server"
                    ErrorMessage="Nama harus diisi."
                    ToolTip="Nama harus diisi.">
                *</asp:RequiredFieldValidator>
            </td>
        </tr>
    </table>
</asp:View>

```

```

        <td>
            <asp:Label ID="Label Form Keterangan"
                SkinID="LabelDetail" runat="server"
                Text="Keterangan"></asp:Label>
        </td>
        <td class="SeparatorH"></td>
        <td>
            <asp:TextBox ID="TextBox Form Keterangan"
                SkinID="TextboxForm"
                runat="server"></asp:TextBox>
        </td>
    </tr>
</table>
</asp:View>
<asp:View ID="View Message" runat="server">
    <div class="Message">
        <asp:Label ID="Label_Message" runat="server"
            Text="Label"></asp:Label>
    </div>
</asp:View>
</asp:MultiView>
<!-- Message, Detail & Form - End -->
</asp:Panel>
<ajaxToolkit:RoundedCornersExtender ID="rce"
    TargetControlID="Panel Modal Content"
    Radius="13"
    Corners="All"
    BorderColor="#040404"
    runat="server" />
</div>
<div class="PanelModalFooter">
    <!-- Modal Windows Action - Start -->
    <div class="PanelModalAction">
        <asp:Button ID="Button_AddChild" SkinID="ButtonModal"
            runat="server" Text="Tambah"
            onclick="Button_AddChild_Click" />
        <asp:Button ID="Button_Edit" SkinID="ButtonModal"
            runat="server" Text="Edit"
            onclick="Button_Edit_Click" />
        <asp:Button ID="Button_Update" SkinID="ButtonModal"
            runat="server" Text="Update" ValidationGroup="Default"
            onclick="Button_Update_Click" />
        <asp:Button ID="Button_Save" SkinID="ButtonModal"
            runat="server" Text="Simpan" ValidationGroup="Default"
            onclick="Button_Save_Click" />
    </div>
    <!-- Modal Windows Action - End -->
</div>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Panel>
<!-- Modal Windows - End -->
</asp:Content>

```

Halaman ini terbagi atas dua bagian (**baris warna kuning**), yaitu

- Main View, bagian ini adalah area akan menampilkan antarmuka seperti pada gambar 64.
- Modal Windows, bagian ini adalah area yang akan menampilkan modal windows untuk menampilkan detail data dan form untuk proses edit dan tambah data, seperti yang dapat dilihat pada gambar 65, gambar 67 dan gambar 68.

Halaman ini juga menggunakan control AJAX Extension, yaitu control UpdatePanel (**baris warna hijau**). Control ini memungkinkan area yang dilingkupinya dapat di-render secara parsial. Sehingga bila terjadi *postback* atau request ke server, hasil respon dari server akan di-render tanpa harus me-render seluruh halaman ini. Pada halaman ini terdapat 3 (tiga) control UpdatePanel, setiap control tersebut memiliki atribut UpdateMode dengan nilai Conditional. UpdatePanel dibuat seperti itu agar setiap control UpdatePanel tidak akan melakukan *render* konten area bila salah satu dari control UpdatePanel melakukan aksi

postback. Dengan menggunakan atribut `UpdateMode=Conditional`, developer akan lebih mudah mengontrol kapan konten area di update sesuai keinginan. Caranya dapat dilakukan dengan menuliskan kode pada file `OrganisasiList.aspx.cs` seperti yang terlihat pada baris-baris berwarna hijau.

Pada halaman ini juga diperkenalkan control `MultiView` dan `View` (baris warna biru muda). Seperti yang dilihat di atas, control `MultiView` adalah wadah yang didalamnya terdapat control `View`. Di dalam control `MultiView` dapat terdiri atas lebih dari satu control `View`. Konten yang terdapat di dalam control `View` tidak akan ditampilkan, kecuali diaktifkan dengan cara seperti baris-baris warna biru yang dapat dilihat pada file `OrganisasiList.aspx.cs` di bawah ini.

Control yang lain adalah control `ModalPopup` (baris warna merah muda) milik `AJAX Control Toolkit`. Control ini memudahkan developer untuk membuat modal windows seperti yang dapat dilihat pada gambar 65, gambar 67 dan gambar 68.

```
<ajaxToolkit:ModalPopupExtender ID="MPE" runat="server"
    TargetControlID="LinkButton1"
    PopupControlID="Panel_Modal"
    BackgroundCssClass="modalBackground"
    DropShadow="true"
    PopupDragHandleControlID="Panel_Modal_Header" />
```

Pada control `ModalPopup` ini terdapat atribut-atribut yang penting untuk diperhatikan, yaitu:

- `PopupControlID`, atribut ini berisi nilai dari id control `Panel` yang akan menjadi modal windows.
- `PopupDragHandleControlID`, atribut ini berisi dari id control `Panel` yang berada di dalam control `Panel` modal windows. `Panel` ini akan menjadi area yang menjadi 'pegangan' agar modal windows dapat dipindah-pindah.

Kemunculan dari modal windows dapat diatur pada file `OrganisasiList.aspx.cs` dengan cara seperti yang ditulis pada baris-baris berwarna merah muda.

Pada halaman ini juga diperkenalkan cara menggunakan control-control untuk validasi (baris warna ungu). Pada file `OrganisasiList.aspx` terdapat dua control dari group `Validation`, yaitu :

- `ValidationSummary`, control ini berfungsi untuk menampilkan pesan 'kesalahan' karena ada item yang tidak sesuai dengan sarat validasi.
- `RequiredFieldValidator`, control ini berfungsi untuk memeriksa apakah suatu control sudah berisi nilai atau belum. Yang perlu diperhatikan pada control ini adalah atribut `ControlToValidate` yang berisi id dari control yang akan divalidasi.

Setiap control yang terlibat dalam proses validasi harus dikelompokkan dalam suatu group dengan cara memberikan nilai yang sama pada atribut `ValidationGroup`. Seperti pada kode di atas dapat dilihat control-control yang melakukan hal ini adalah :

- `ValidationSummary.`
- `TextBox_Form_ID.`
- `RFV_Form_ID.`
- `TextBox_Form_Nama.`
- `RFV_Form_Nama.`
- `Button_Update.`
- `Button_Save.`

Dengan memberikan atribut `ValidationGroup` pada control-control tersebut maka proses validasi hanya akan berpengaruh pada control-control tersebut saja.

Dan di bawah ini adalah isi dari file `OrganisasiList.aspx.cs`.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.Linq;

using SPU.DataAccess;
using SPU.Entities;

public partial class DataMaster OrganisasiList : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        Label_Title.Text = Page.Title;

        if (!IsPostBack)
        {
            PopulateNodes();
        }
    }

    #region Databind
    public void PopulateNodes()
    {
        TreeView_Main.Nodes.Clear();

        DataTable data = GetTreeViewData();
        DataView child = GetParent(data);
        foreach (DataRowView row in child)
        {
            TreeNode childNode = new TreeNode();
            childNode.Text = row["OrganisasiNama"].ToString();
            childNode.Value = row["OrganisasiID"].ToString();
            TreeView_Main.Nodes.Add(childNode);
            AddChild(data, childNode);
        }

        TreeView_Main.ExpandAll();
    }

    protected DataTable GetTreeViewData()
    {
        SPUDataAccess db = new SPUDataAccess();
        DataTable dt = new DataTable();
        DataRow dr;

        dt.Columns.Add("OrganisasiID");
        dt.Columns.Add("OrganisasiInduk");
        dt.Columns.Add("OrganisasiNama");

        var data = db.SPUDataContext.Organisasis.Select(p => p);
        foreach (Organisasi org in data)
        {
            dr = dt.NewRow();
            dr["OrganisasiID"] = org.OrganisasiID;
            dr["OrganisasiInduk"] = org.OrganisasiInduk;
            dr["OrganisasiNama"] = org.OrganisasiNama;
            dt.Rows.Add(dr);
        }

        return dt;
    }

    DataView GetParent(DataTable parent)
    {

```

```

        DataView view = new DataView(parent);
        view.RowFilter = "OrganisasiInduk='0'";
        return view;
    }

    DataView GetChild(DataTable parent, string parentID)
    {
        DataView view = new DataView(parent);
        view.RowFilter = "OrganisasiInduk='" + parentID + "'";
        return view;
    }

    void AddChild(DataTable parent, TreeNode node)
    {
        DataView child = GetChild(parent, node.Value);
        foreach (DataRowView row in child)
        {
            TreeNode childNode = new TreeNode();
            childNode.Text = row["OrganisasiNama"].ToString();
            childNode.Value = row["OrganisasiID"].ToString();
            node.ChildNodes.Add(childNode);
            AddChild(parent, childNode);
        }
    }
    #endregion Databind

    #region Action
    protected void TreeView_Main_SelectedNodeChanged(object sender, EventArgs e)
    {
        Button_Edit.Visible = true;
        Button_Update.Visible = false;
        Button_Save.Visible = false;
        Button_AddChild.Visible = true;
        MultiView_Content.SetActiveView(View_Detail);

        string id = TreeView_Main.SelectedValue;

        SPUDataAccess db = new SPUDataAccess();
        Organisasi org = db.SPUDataContext.Organisasi.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();

        Label_Value_ID.Text = org.OrganisasiID;
        Label_Value_Induk.Text = org.OrganisasiInduk;
        Label_Value_Nama.Text = org.OrganisasiNama;
        Label_Value_Keterangan.Text = org.Keterangan;

        UpdatePanel_Panel_Modal.Update();
        this.MPE.Show();
    }

    protected void ImageButton_Delete_Click(object sender, ImageClickEventArgs e)
    {
        try
        {
            if (TreeView_Main.CheckedNodes.Count > 0)
            {
                foreach (TreeNode node in TreeView_Main.CheckedNodes)
                {
                    SPUDataAccess db = new SPUDataAccess();
                    Organisasi org = db.SPUDataContext.Organisasi.Where(p =>
p.OrganisasiID.Equals(node.Value)).Single<Organisasi>();
                    db.SPUDataContext.Organisasi.DeleteOnSubmit(org);
                    db.SPUDataContext.SubmitChanges();
                }
            }
        }
        catch (Exception ex)
        {
        }

        PopulateNodes();
        UpdatePanel_Main.Update();
    }

    protected void ImageButton_Close_Click(object sender, ImageClickEventArgs e)

```

```

{
    this.MPE.Hide();
    PopulateNodes();
    UpdatePanel_Main.Update();
}

protected void Button_Edit_Click(object sender, EventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = true;
    Button_Save.Visible = false;
    Button_AddChild.Visible = false;
    TextBox_Form_ID.Enabled = false;
    TextBox_Form_Induk.Enabled = false;
    MultiView_Content.SetActiveView(View_Form);

    string id = TreeView_Main.SelectedValue;
    SPUDataAccess db = new SPUDataAccess();
    Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();

    TextBox_Form_ID.Text = org.OrganisasiID;
    TextBox_Form_Induk.Text = org.OrganisasiInduk;
    TextBox_Form_Nama.Text = org.OrganisasiNama;
    TextBox_Form_Keterangan.Text = org.Keterangan;
}

protected void Button_Update_Click(object sender, EventArgs e)
{
    Button_Edit.Visible = true;
    Button_Update.Visible = false;
    Button_Save.Visible = false;
    Button_AddChild.Visible = true;
    MultiView_Content.SetActiveView(View_Detail);

    string id = TreeView_Main.SelectedValue;
    SPUDataAccess db = new SPUDataAccess();
    Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();

    org.OrganisasiNama = TextBox_Form_Nama.Text;
    org.Keterangan = TextBox_Form_Keterangan.Text;

    db.SPUDataContext.SubmitChanges();

    org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();
    Label_Value_ID.Text = org.OrganisasiID;
    Label_Value_Induk.Text = org.OrganisasiInduk;
    Label_Value_Nama.Text = org.OrganisasiNama;
    Label_Value_Keterangan.Text = org.Keterangan;
}

protected void Button_Save_Click(object sender, EventArgs e)
{
    SPUDataAccess db = new SPUDataAccess();
    Organisasi org = new Organisasi();

    org.OrganisasiID = TextBox_Form_ID.Text;
    org.OrganisasiInduk = TreeView_Main.SelectedValue;
    org.OrganisasiNama = TextBox_Form_Nama.Text;
    org.Keterangan = TextBox_Form_Keterangan.Text;

    db.SPUDataContext.Organisasis.InsertOnSubmit(org);
    db.SPUDataContext.SubmitChanges();

    this.MPE.Hide();
    PopulateNodes();
    UpdatePanel_Panel_Modal.Update();
    UpdatePanel_Main.Update();
}

protected void Button_AddChild_Click(object sender, EventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = false;

```

```

        Button_Save.Visible = true;
        Button_AddChild.Visible = false;
        TextBox_Form_ID.Enabled = true;
        TextBox_Form_Induk.Enabled = true;
        MultiView_Content.SetActiveView(View_Form);

        TextBox_Form_ID.Text = String.Empty;
        TextBox_Form_Induk.Text = TreeView_Main.SelectedNode.Text;
        TextBox_Form_Nama.Text = String.Empty;
        TextBox_Form_Keterangan.Text = String.Empty;
    }
    #endregion Action
}

```

Berikut ini adalah penjelasan dari barisan kode di atas.

Daftar Organisasi

Untuk menampilkan data digunakan control `TreeView` yang dapat ditemui pada group `Navigation`.

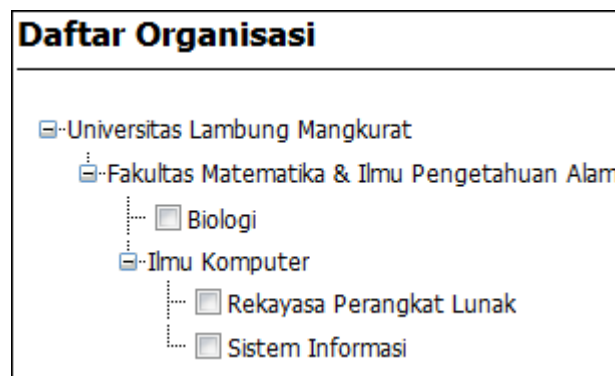
Berikut ini adalah baris kode dari control `TreeView` yang ada pada file `OrganisasiList.aspx`.

```

<asp:TreeView ID="TreeView_Main"
    ShowLines="true"
    ShowCheckBoxes="Leaf"
    runat="server"
    onselectednodechanged="TreeView_Main_SelectedNodeChanged">
    <SelectedNodeStyle Font-Bold="true" />
</asp:TreeView>

```

Pada penulisan control `TreeView` di atas terdapat atribut `ShowLines` untuk menampilkan garis pada setiap item-item pada `TreeView` sedangkan atribut `ShowCheckBoxes` untuk menampilkan control `Checkbox` pada item terendah.



Gambar 72. `TreeView` dengan atribut `ShowLines` dan `ShowCheckBoxes`.

Pada control `TreeView` di atas juga dapat dilihat event `onselectednodechanged`, event ini akan menangkap kejadian ketika salah satu item pada `TreeView` dipilih. Event handler dari `onselectednodechanged` adalah `TreeView_Main_SelectedNodeChanged` yang dapat dilihat pada file `OrganisasiList.aspx.cs`.

Setelah control `TreeView` ditambahkan dengan cara di atas, maka langkah selanjutnya adalah 'menyuntikkan' data ke control `TreeView` agar data ditampilkan seperti pada gambar 72. Untuk membuat hal tersebut maka perlu ditambahkan method-method berikut ini pada file `OrganisasiList.aspx`, yaitu :

- a. PopulateNodes.
- b. GetTreeViewData.
- c. GetParent.
- d. GetChild.
- e. AddChild.

Selanjutnya cukup memanggil method `PopulateNodes` agar data dapat ditampilkan pada control `TreeView`. Apabila ingin menampilkan data ketika halaman `OrganisasiList.aspx` dibuka maka cukup menambahkan kode berikut pada method `Page_Load`.

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        PopulateNodes();
    }
}
```

Detail Organisasi

Detail organisasi adalah fitur untuk menampilkan detail data dari item yang dipilih pada `TreeView` seperti yang dilihat pada gambar 65.

Saat item pada control `TreeView` diklik dengan mouse maka akan ditampilkan modal windows yang didalamnya terdapat detail data dari item yang dipilih.

```
<asp:TreeView ID="TreeView Main" ShowLines="true" ShowCheckBoxes="Leaf"
    runat="server"
    onselectednodechanged="TreeView Main SelectedNodeChanged">
    <SelectedNodeStyle Font-Bold="true" />
</asp:TreeView>
```

Event `onselectednodechanged` adalah event yang menangani perilaku di atas. Dan berikut ini adalah isi dari method `TreeView_Main_SelectedNodeChanged`.

```
protected void TreeView_Main_SelectedNodeChanged(object sender, EventArgs e)
{
    Button_Edit.Visible = true;
    Button_Update.Visible = false;
    Button_Save.Visible = false;
    Button_AddChild.Visible = true;
    MultiView_Content.SetActiveView(View_Detail);

    string id = TreeView_Main.SelectedValue;

    SPUDataAccess db = new SPUDataAccess();
    Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();

    Label_Value_ID.Text = org.OrganisasiID;
    Label_Value_Induk.Text = org.OrganisasiInduk;
    Label_Value_Nama.Text = org.OrganisasiNama;
    Label_Value_Keterangan.Text = org.Keterangan;

    UpdatePanel_Panel Modal.Update();
    this.MPE.Show();
}
```

Edit Organisasi

Setelah detail ditampilkan maka user dapat mengedit data tersebut dengan mengklik tombol Edit.

```
<asp:Button ID="Button Edit"
            SkinID="ButtonModal"
            runat="server"
            Text="Edit"
            onclick="Button_Edit_Click" />
```

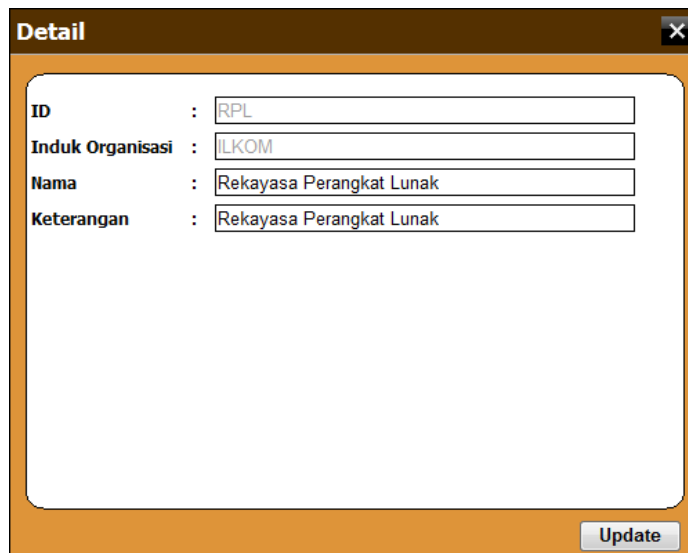
Setelah tombol Edit diklik maka akan dijalankan method `Button_Edit_Click`. Berikut adalah isi dari method ini.

```
protected void Button_Edit_Click(object sender, EventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = true;
    Button_Save.Visible = false;
    Button_AddChild.Visible = false;
    TextBox_Form_ID.Enabled = false;
    TextBox_Form_Induk.Enabled = false;
    MultiView_Content.SetActiveView(View_Form);

    string id = TreeView_Main.Selected.Value;
    SPUDataAccess db = new SPUDataAccess();
    Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();

    TextBox_Form_ID.Text = org.OrganisasiID;
    TextBox_Form_Induk.Text = org.OrganisasiInduk;
    TextBox_Form_Nama.Text = org.OrganisasiNama;
    TextBox_Form_Keterangan.Text = org.Keterangan;
}
```

Dan berikut adalah antarmuka setelah tombol Edit diklik.

The image shows a web application window titled "Detail" with a close button (X) in the top right corner. The window contains a form with four labeled text input fields: "ID" with the value "RPL", "Induk Organisasi" with the value "ILKOM", "Nama" with the value "Rekayasa Perangkat Lunak", and "Keterangan" with the value "Rekayasa Perangkat Lunak". At the bottom right of the form, there is a button labeled "Update".

Gambar 73. Form edit.

Pada form edit akan ditampilkan tombol Update dan menyembunyikan tombol-tombol lainnya.

```
<asp:Button ID="Button Update"
            SkinID="ButtonModal"
            runat="server"
            Text="Update"
            onclick="Button_Update_Click" />
```

Dan setelah tombol Update diklik maka method Button_Update_Click akan dieksekusi.

```
protected void Button_Update_Click(object sender, EventArgs e)
{
    Button Edit.Visible = true;
    Button Update.Visible = false;
    Button_Save.Visible = false;
    Button_AddChild.Visible = true;
    MultiView_Content.SetActiveView(View_Detail);

    string id = TreeView Main.SelectedValue;
    SPUDataAccess db = new SPUDataAccess();
    Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();

    org.OrganisasiNama = TextBox Form Nama.Text;
    org.Keterangan = TextBox Form Keterangan.Text;

    db.SPUDataContext.SubmitChanges();

    org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(id)).Single<Organisasi>();
    Label_Value_ID.Text = org.OrganisasiID;
    Label_Value_Induk.Text = org.OrganisasiInduk;
    Label_Value_Nama.Text = org.OrganisasiNama;
    Label_Value_Keterangan.Text = org.Keterangan;
}
```

Tambah Organisasi

Pada antarmuka detail juga terdapat fasilitas untuk menambah organisasi dengan memilih tombol Tambah.

```
<asp:Button ID="Button_AddChild"
SkinID="ButtonModal"
runat="server"
Text="Tambah"
onclick="Button_AddChild_Click" />
```

Setelah tombol ini diklik maka akan ditampilkan form untuk menambah data. Berikut ini adalah isi dari method Button_AddChild_Click.

```
protected void Button_AddChild_Click(object sender, EventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = false;
    Button_Save.Visible = true;
    Button_AddChild.Visible = false;
    TextBox Form ID.Enabled = true;
    TextBox Form Induk.Enabled = true;
    MultiView_Content.SetActiveView(View_Form);

    TextBox Form ID.Text = String.Empty;
    TextBox Form Induk.Text = TreeView Main.SelectedValue;
    TextBox Form Nama.Text = String.Empty;
    TextBox Form Keterangan.Text = String.Empty;
}
```

Hapus Organisasi

Seperti yang diperlihatkan pada gambar 69 dan gambar 70 dapat menggambarkan bagaimana cara item-item pada dihapus.

Berikut ini tombol aksi untuk menghapus data.

```
<asp:ImageButton ID="ImageButton_Delete"
    ImageUrl="~/App_Themes/SPU/icons/icon_delete.gif" runat="server"
    onclick="ImageButton_Delete_Click" />

<ajaxToolkit:ConfirmButtonExtender ID="CBE_ConfirmationDelete"
    ConfirmText="Apakah data akan dihapus?" runat="server"
    TargetControlID="ImageButton_Delete" />
```

Control `ImageButton_Delete` adalah tombol yang akan mengeksekusi method `ImageButton_Delete_Click` untuk menghapus data. Tetapi sebelum aksi dilakukan terlebih dahulu akan ditampilkan pesan konfirmasi, dengan menggunakan control `ConfirmButton` milik `AJAX Control Toolkit` maka konfirmasi akan dilakukan.

Berikut ini adalah isi dari method `ImageButton_Delete_Click`.

```
protected void ImageButton_Delete_Click(object sender, ImageClickEventArgs e)
{
    try
    {
        if (TreeView_Main.CheckedNodes.Count > 0)
        {
            foreach (TreeNode node in TreeView_Main.CheckedNodes)
            {
                SPUDataAccess db = new SPUDataAccess();
                Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(node.Value)).Single<Organisasi>();
                db.SPUDataContext.Organisasis.DeleteOnSubmit(org);
                db.SPUDataContext.SubmitChanges();
            }
        }
    }
    catch (Exception ex)
    {
    }

    PopulateNodes();
    UpdatePanel_Main.Update();
}
```

Apa yang telah dipelajari?

Pada bagian ini telah dipelajari beberapa fitur dan control yang dimiliki oleh ASP.NET 4, yaitu :

- Control `TreeView` dan cara mengisi data secara hirarki pada control ini.
- Membuat modal windows dengan menggunakan `ModalPopup`, `AJAX Control Toolkit`.
- Mengenal validasi dengan control pada group `Validation`.
- Control `MultiView` dan `View`.
- Control `Button`, `ImageButton` dan `LinkButton`.
- Control `Label` dan `TextBox`.
- Control `UpdatePanel`.
- Membuat pesan konfirmasi dengan control `ConfirmButton`, `AJAX Control Toolkit`.

User Management

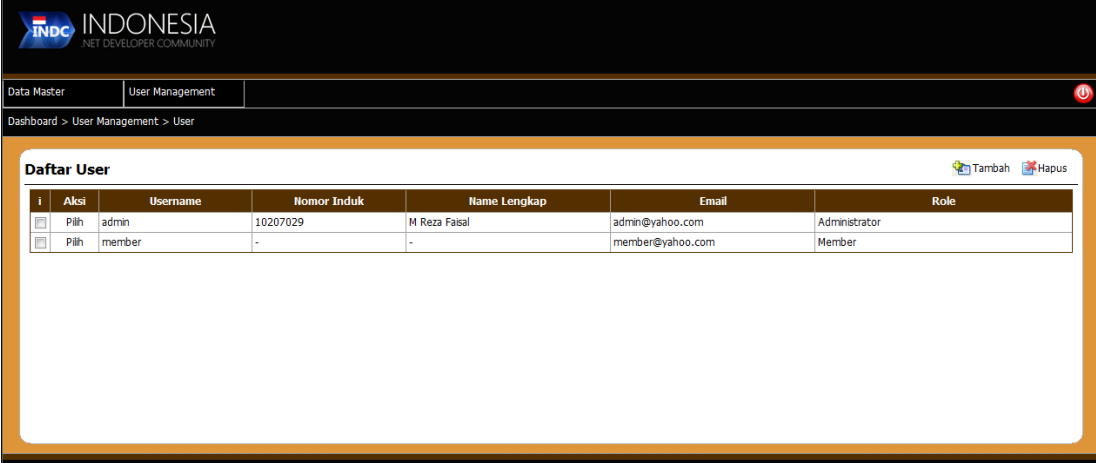
User management adalah modul untuk mengelola user, modul ini terdiri fitur untuk menambah, menghapus, update dan menampilkan data user.

Proses CRUD pada model ini tidak menggunakan Data Access Layer yang telah dibuat, tetapi menggunakan class `Membership` dan `ProfileCommon` yang memang telah disediakan pada ASP.NET untuk mengelola user.

Mengelola User

Sebelum membuat modul ini, terlebih dahulu akan diperlihatkan antarmuka dan cara bekerja dari modul ini.

Berikut ini adalah antarmuka halaman `UserList.aspx`. Pertama kali, halaman ini akan menampilkan daftar user seperti gambar berikut ini.



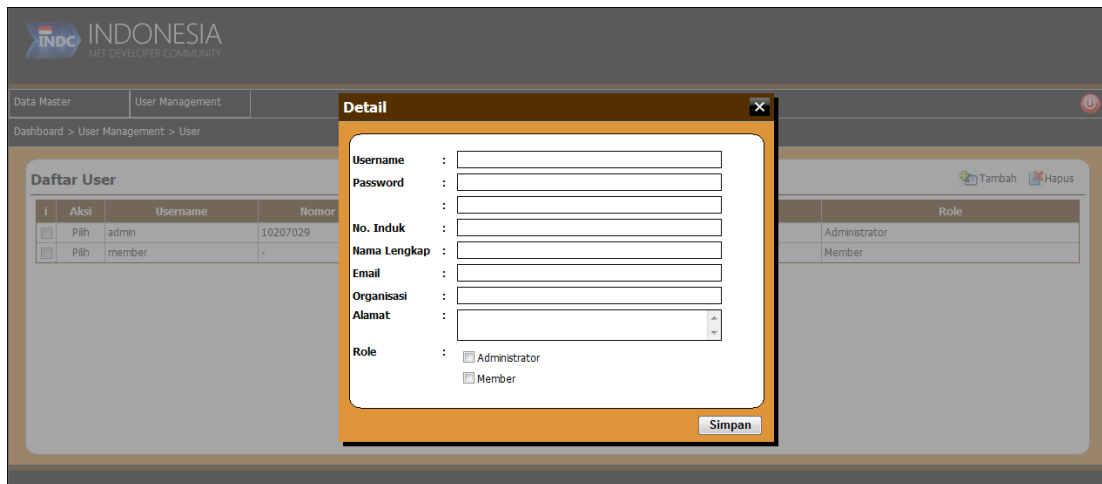
The screenshot shows a web application interface for user management. At the top, there is a header with the INDC Indonesia logo and navigation tabs for 'Data Master' and 'User Management'. Below the tabs, a breadcrumb trail reads 'Dashboard > User Management > User'. The main content area is titled 'Daftar User' and contains a table with two rows of user data. To the right of the table are two buttons: 'Tambah' (Add) and 'Hapus' (Delete). The table has columns for 'i' (index), 'Aksi' (Action), 'Username', 'Nomor Induk' (ID Number), 'Name Lengkap' (Full Name), 'Email', and 'Role'.

i	Aksi	Username	Nomor Induk	Name Lengkap	Email	Role
<input type="checkbox"/>	Pilih	admin	10207029	M Reza Faisal	admin@yahoo.com	Administrator
<input type="checkbox"/>	Pilih	member	-	-	member@yahoo.com	Member

Gambar 74. Daftar user.

Untuk menambah data maka user dapat menekan tombol `Tambah` yang ada pada sisi kanan halaman ini. Setelah tombol `Tambah` dipilih maka akan ditampilkan modal window yang merupakan form untuk menambah data.

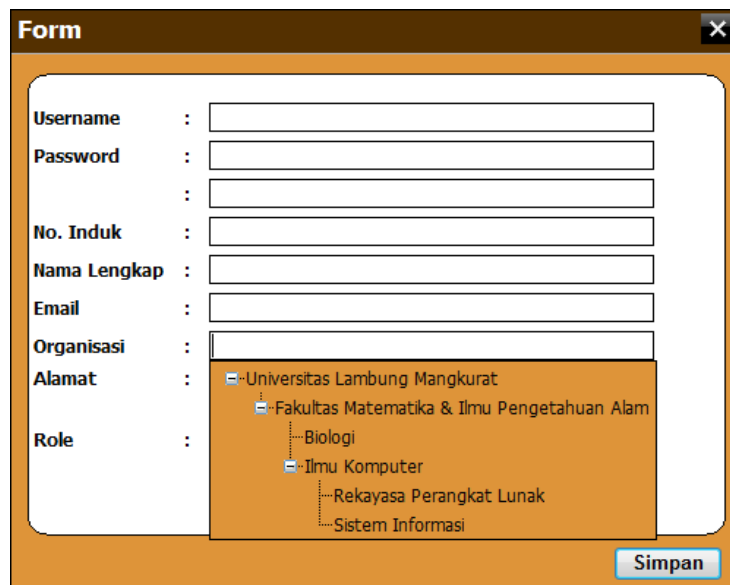
Berikut ini adalah antarmuka yang dapat dilihat ketika modal window ditampilkan.



Gambar 75. Form menambah data.

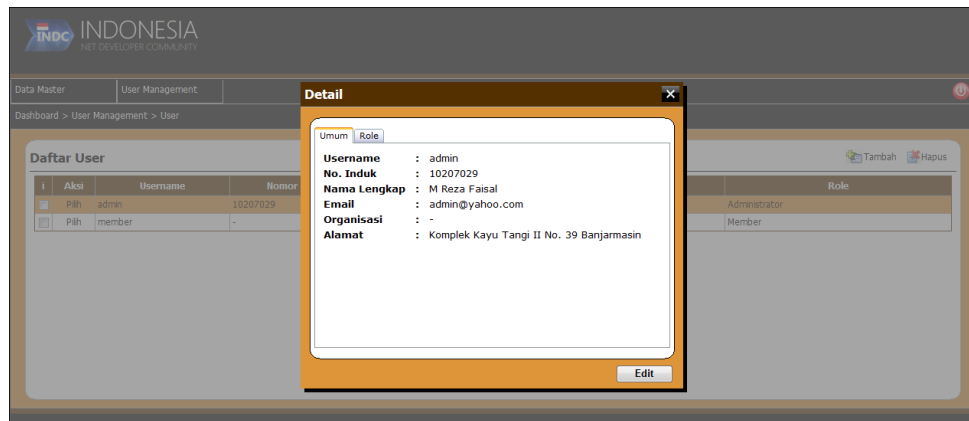
Form ini tidak hanya berisi textbox yang dapat diisi oleh user saat mengisi data, tetapi juga terdapat control `CheckBoxList` yang memudahkan user untuk memilih role yang dapat diberikan pada user yang akan dibuat.

Form ini juga menggunakan control `PopupControl` milik AJAX Control Toolkit sehingga user dapat memilih data organisasi seperti gambar berikut ini.



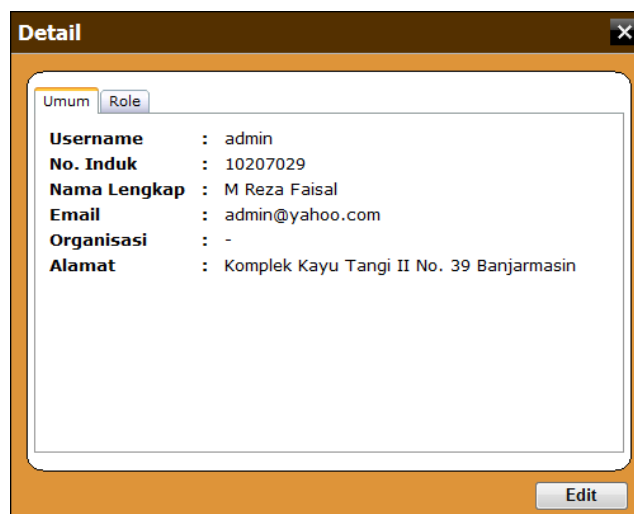
Gambar 76. Popup Control untuk menampilkan data organisasi.

Untuk melihat salah satu detail data user yang telah terdaftar maka user cukup memilih salah satu baris yang ada pada daftar user seperti yang terlihat pada gambar 75. Berikut ini adalah antarmuka ketika data detail ditampilkan.



Gambar 77. Detail salah satu data user.

Bila diperhatikan lebih dekat maka dapat dilihat modal windows seperti berikut ini.



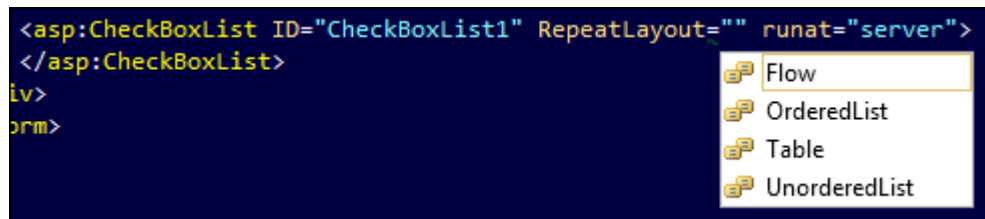
Gambar 78. Detail data admin.

Pada modal window ini terdapat penggunaan control Tab milih AJAX Control Toolkit. Apa bila menekan tombol Edit maka akan ditampilkan form seperti berikut ini.

Gambar 79. Edit data user.

Pada proses edit ini password boleh diisi atau tidak, bila Textbox password tidak diisi maka tidak akan terjadi proses pengubahan password ketika tombol Update ditekan. Password akan diupdate apabila Textbox password diisi, user harus mengisi password sebanyak dua kali untuk memastikan kebenaran password yang diisikan.

Pada gambar di atas dapat dilihat control CheckBoxList yang digunakan untuk menampilkan data Role. Sebagai informasi, pada ASP.NET 4.0 terdapat penambahan 'kemampuan' pada control CheckBoxList maupun RadioButtonList, yaitu penambahan nilai pada yang dapat digunakan oleh atribut RepeatLayout. Pada versi ASP.NET sebelumnya hanya bisa digunakan nilai Flow dan Table, maka pada ASP.NET 4.0 terdapat dua nilai baru seperti yang dapat dilihat pada gambar di bawah ini.



Gambar 80. Nilai pada atribut RepeatLayout.

Kode Lengkap

Untuk mengelola user akan digunakan satu halaman yang disimpan pada file UserList.aspx dan UserList.aspx.cs sebagai code file.

Berikut ini adalah isi dari file UserList.aspx.

```
<%@ Page Title="Daftar User" Language="C#"
MasterPageFile="~/Shared/MasterPages/Member.master" AutoEventWireup="true"
CodeFile="UserList.aspx.cs" Inherits="UserManagement_UserList" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head_Member" Runat="Server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder_Member"
Runat="Server">
    <!-- Main View - Start -->
    <div id="ApplicationContainer">

        <!-- Title & Buttons - Start -->
        <div id="TitleToolContainer">
            <div id="Title">
                <asp:Label ID="Label_Title" CssClass="TextTitle" runat="server"
                    Text="Label"></asp:Label>
            </div>
            <asp:UpdatePanel ID="UpdatePanel ActionTool" UpdateMode="Conditional"
                runat="server">
                <ContentTemplate>
                    <div id="ActionTool">
                        <div class="ActionToolItemText">
                            <asp:Label ID="Label_Delete" runat="server"
                                AssociatedControlID="ImageButton_Delete"
                                Text="Hapus"></asp:Label>
                        </div>
                        <div class="ActionToolItem">
                            <asp:ImageButton ID="ImageButton_Delete"
                                ImageUrl="~/App Themes/SPU/icons/icon delete.gif"
                                runat="server"
                                onclick="ImageButton_Delete_Click" />
                            <ajaxToolkit:ConfirmButtonExtender
                                ID="CBE_ConfirmationDelete"
                                ConfirmText="Apakah data akan dihapus?"
                                runat="server"
                                TargetControlID="ImageButton_Delete" />
                        </div>
                    </div>
                </ContentTemplate>
            </asp:UpdatePanel>
        </div>
    </div>
</asp:Content>
```

```

        </div>
        <div class="ActionToolItemText">
            <asp:Label ID="Label_Tambah" runat="server"
                Text="Tambah"></asp:Label>
        </div>
        <div class="ActionToolItem">
            <asp:ImageButton ID="ImageButton_Tambah"
                ImageUrl="~/App Themes/SPU/icons/icon add.gif"
                runat="server"
                onclick="ImageButton_Tambah_Click" />
        </div>
    </div>
    <div style="visibility:hidden">
        <asp:LinkButton ID="LinkButton1"
            runat="server">LinkButton</asp:LinkButton>
        <ajaxToolkit:ModalPopupExtender ID="MPE" runat="server"
            TargetControlID="LinkButton1"
            PopupControlID="Panel_Modal"
            BackgroundCssClass="modalBackground"
            DropShadow="true"
            PopupDragHandleControlID="Panel_Modal_Header" />
    </div>
</ContentTemplate>
</asp:UpdatePanel>
</div>
<!-- Title & Buttons - End -->

<div class="Separator"></div>

<!-- Main Data - Start -->
<div id="ContentContainer">
    <asp:UpdatePanel ID="UpdatePanel_Main" UpdateMode="Conditional"
        runat="server">
        <ContentTemplate>
            <asp:GridView ID="GridView_Main" ClientIDMode="Static"
                AutoGenerateColumns="false" DataKeyNames="UserName"
                AllowPaging="true" AllowSorting="true" Width="100%"
                BorderWidth="1"
                OnSelectedIndexChanged="GridView_Main_SelectedIndexChanged"
                OnSorted="GridView_Main_Sorted"
                OnPageIndexChanged="GridView_Main_PageIndexChanged"
                OnPageIndexChanging="GridView_Main_PageIndexChanging"
                runat="server">
                <EmptyDataTemplate>
                    <div class="EmptyDataMessage">
                        <asp:Label ID="Label_GridView_Empty"
                            runat="server"
                            Text="Data tidak ditemukan."></asp:Label>
                    </div>
                </EmptyDataTemplate>
                <HeaderStyle CssClass="GridviewHeader" />
                <SelectedRowStyle CssClass="GridviewSelect" />
                <AlternatingRowStyle CssClass="GridviewAlternate" />
                <Columns>
                    <asp:TemplateField>
                        <HeaderTemplate>
                            <div style="text-align:center">i</div>
                        </HeaderTemplate>
                        <ItemTemplate>
                            <asp:CheckBox ID="CheckBox_Item"
                                runat="server" />
                        </ItemTemplate>
                        <ItemStyle HorizontalAlign="Center"
                            VerticalAlign="Top" />
                    </asp:TemplateField>
                    <asp:TemplateField HeaderText="Aksi" ItemStyle-
                        CssClass="GridviewItem"
                        ItemStyle-VerticalAlign="Top" ItemStyle-
                        HorizontalAlign="Center" HeaderStyle-Width="5%"
                        ItemStyle-Wrap="false">
                        <ItemTemplate>
                            <asp:LinkButton ID="LinkButton_Select"
                                CommandName="Select"
                                runat="server">Pilih</asp:LinkButton>
                        </ItemTemplate>
                    </asp:TemplateField>
                </Columns>
            </ContentTemplate>
        </asp:UpdatePanel>
    </div>

```

```

        <asp:BoundField DataField="UserName"
            HeaderText="Username"
            ItemStyle-CssClass="GridviewItem"
            HeaderStyle-Width="15%"
            ItemStyle-VerticalAlign="Top" />
        <asp:TemplateField HeaderText="Nomor Induk" ItemStyle-
            VerticalAlign="Top"
            ItemStyle-CssClass="GridviewItem"
            HeaderStyle-Width="15%">
            <ItemTemplate>
                <%#
                    GetNomorInduk((String)DataBinder.
                        Eval(Container.DataItem, "UserName"))
                %>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Name Lengkap"
            ItemStyle-VerticalAlign="Top"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-
            Width="20%">
            <ItemTemplate>
                <%#
                    GetFullName((String)DataBinder.
                        Eval(Container.DataItem, "UserName"))
                %>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="Email" HeaderText="Email"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-
            Width="20%" ItemStyle-VerticalAlign="Top" />
        <asp:TemplateField HeaderText="Role" ItemStyle-
            VerticalAlign="Top"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-
            Width="25%">
            <ItemTemplate>
                <%#
                    GetRoles((String)DataBinder.
                        Eval(Container.DataItem, "UserName"))
                %>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
    <PagerSettings Mode="NumericFirstLast"
        FirstPageImageUrl="~/App_Themes/SPU/icons/icon_nav_first_on.gif"
        FirstPageText="Pertama"
        LastPageImageUrl="~/App_Themes/SPU/icons/icon_nav_last_on.gif"
        LastPageText="Akhir"
        NextPageImageUrl="~/App_Themes/SPU/icons/icon_nav_next_on.gif"
        PreviousPageImageUrl="~/App_Themes/SPU/icons/icon_nav_prev_on.gif" />
    <PagerStyle HorizontalAlign="Center" VerticalAlign="Middle"
        ForeColor="#543000" />
    </asp:GridView>
</ContentTemplate>
</asp:UpdatePanel>
</div>
<%-- Main Data - End --%>
</div>
<%-- Main View - End --%>

<%-- Modal Windows - Start --%>
<asp:Panel ID="Panel Modal" CssClass="PanelModal" runat="server">
    <asp:UpdatePanel ID="UpdatePanel_Panel_Modal" UpdateMode="Conditional"
        runat="server">
        <ContentTemplate>
            <div class="PanelModalHeader">
                <%-- Header Modal Windows - Start --%>
                <asp:Panel ID="Panel Modal Header" runat="server">
                    <div class="TextTitleModalHeader">
                        <asp:Label ID="Label_Modal_Title" runat="server"
                            Text="Detail"></asp:Label>
                    </div>
                    <div class="ActionCloseModalHeader">
                        <asp:ImageButton ID="ImageButton_Close"
                            ImageUrl="~/App_Themes/SPU/icons/icon_modal_button_close.gif"
                            AlternateText="Tutup" runat="server"
                            onclick="ImageButton_Close_Click" /></div>
                </asp:Panel>
            </div>
        </ContentTemplate>
    </asp:UpdatePanel>
</asp:Panel>

```

```

        </asp:Panel>
        <!-- Header Modal Windows - End -->
    </div>
    <div class="Separator"></div>
    <div class="PanelModalContentContainer">
        <asp:Panel ID="Panel_Modal_Content" CssClass="PanelModalContent"
            runat="server">
            <!-- Message, Detail & Form - Start -->
            <asp:MultiView ID="MultiView Content" runat="server">
                <asp:View ID="View Detail" runat="server">
                    <div style="padding-left:5px;padding-right:5px">
                        <ajaxToolkit:TabContainer ID="TabContainer1"
                            runat="server" Height="250px">
                            <ajaxToolkit:TabPanel runat="server"
                                HeaderText="Umum">
                                <ContentTemplate>
                                    <table>
                                        <tr>
                                            <td>
                                                <asp:Label
                                                    ID="Label Detail Username"
                                                    SkinID="LabelDetail"
                                                    runat="server"
                                                    Text="Username"></asp:Label>
                                            </td>
                                            <td class="SeparatorH">:</td>
                                            <td>
                                                <asp:Label
                                                    ID="Label_Value_Username"
                                                    runat="server"
                                                    Text="Label"></asp:Label>
                                            </td>
                                        </tr>
                                        <tr>
                                            <td>
                                                <asp:Label
                                                    ID="Label Detail NoInduk"
                                                    SkinID="LabelDetail"
                                                    runat="server"
                                                    Text="No. Induk"></asp:Label>
                                            </td>
                                            <td class="SeparatorH">:</td>
                                            <td>
                                                <asp:Label
                                                    ID="Label_Value_NoInduk"
                                                    runat="server"
                                                    Text="Label"></asp:Label>
                                            </td>
                                        </tr>
                                        <tr>
                                            <td>
                                                <asp:Label
                                                    ID="Label_Detail_NamaLengkap"
                                                    SkinID="LabelDetail"
                                                    runat="server"
                                                    Text="Nama Lengkap">
                                                </asp:Label>
                                            </td>
                                            <td class="SeparatorH">:</td>
                                            <td>
                                                <asp:Label
                                                    ID="Label_Value_NamaLengkap"
                                                    runat="server" Text="Label">
                                                </asp:Label>
                                            </td>
                                        </tr>
                                        <tr>
                                            <td>
                                                <asp:Label
                                                    ID="Label_Detail_Email"
                                                    SkinID="LabelDetail"
                                                    runat="server" Text="Email">
                                                </asp:Label>
                                            </td>
                                            <td class="SeparatorH">:</td>
                                            <td>

```

```

        <asp:Label
            ID="Label Value Email"
            runat="server" Text="Label">
        </asp:Label>
    </td>
</tr>
<tr>
    <td valign="top">
        <asp:Label
            ID="Label Detail Org"
            SkinID="LabelDetail"
            runat="server"
            Text="Organisasi">
        </asp:Label>
    </td>
    <td valign="top"
        class="SeparatorH">:</td>
    <td>
        <asp:Label ID="Label_Value_Org"
            runat="server" Text="Label">
        </asp:Label>
    </td>
</tr>
<tr>
    <td>
        <asp:Label
            ID="Label Detail Alamat"
            SkinID="LabelDetail"
            runat="server"
            Text="Alamat">
        </asp:Label>
    </td>
    <td class="SeparatorH">:</td>
    <td>
        <asp:Label
            ID="Label_Value_Alamat"
            runat="server" Text="Label">
        </asp:Label>
    </td>
</tr>
</table>
</ContentTemplate>
</ajaxToolkit:TabPanel>
<ajaxToolkit:TabPanel ID="TabPanel1"
    runat="server" HeaderText="Role">
    <ContentTemplate>
        <asp:Label ID="Label_Value_Roles"
            runat="server" Text="Label">
        </asp:Label>
    </ContentTemplate>
</ajaxToolkit:TabPanel>
</ajaxToolkit:TabContainer>
</div>
</asp:View>
<asp:View ID="View Form" runat="server">
    <table>
        <tr>
            <td colspan="3">
                <asp:ValidationSummary
                    ID="ValidationSummary Form"
                    ValidationGroup="Default" runat="server" />
            </td>
        </tr>
        <tr>
            <td>
                <asp:Label SkinID="LabelDetail" runat="server"
                    Text="Username"></asp:Label>
            </td>
            <td class="SeparatorH">:</td>
            <td>
                <asp:TextBox ID="TextBox Form Username"
                    ValidationGroup="Default"
                    SkinID="TextboxForm" runat="server">
                </asp:TextBox>
            </td>
        </tr>
    </table>
    <asp:RequiredFieldValidator

```



```

        ID="RFV_Form_Username"
        ValidationGroup="Default"
        ControlToValidate="TextBox Form Username"
        runat="server"
        ErrorMessage="Username harus diisi."
        ToolTip="Username harus diisi.">*
    </asp:RequiredFieldValidator>
</td>
</tr>
<tr>
    <td>
        <asp:Label SkinID="LabelDetail" runat="server"
            Text="Password"></asp:Label>
    </td>
    <td class="SeparatorH"></td>
    <td>
        <asp:TextBox ID="TextBox_Form_Password"
            ValidationGroup="Default"
            TextMode="Password" SkinID="TextboxForm"
            runat="server">
        </asp:TextBox>

        <asp:RequiredFieldValidator
            ID="RFV_Form_Password"
            ValidationGroup="Default"
            ControlToValidate="TextBox Form Password"
            runat="server"
            ErrorMessage="Password harus diisi."
            ToolTip="Password harus diisi.">*
        </asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td class="SeparatorH"></td>
    <td>
        <asp:TextBox ID="TextBox Form PasswordC"
            ValidationGroup="Default"
            TextMode="Password" SkinID="TextboxForm"
            runat="server">
        </asp:TextBox>

        <asp:RequiredFieldValidator
            ID="RFV_Form_PasswordC"
            ValidationGroup="Default"
            ControlToValidate="TextBox_Form_PasswordC"
            runat="server"
            ErrorMessage="Password harus diisi."
            ToolTip="Password harus diisi.">*
        </asp:RequiredFieldValidator>

        <asp:CompareValidator ID="CV_Form_Password"
            ValidationGroup="Default"
            ControlToValidate="TextBox Form PasswordC"
            ControlToCompare="TextBox Form Password"
            Operator="Equal" runat="server"
            ErrorMessage="Password harus sama."
            ToolTip="Password harus sama.">*
        </asp:CompareValidator>
    </td>
</tr>
<tr>
    <td>
        <asp:Label SkinID="LabelDetail" runat="server"
            Text="No. Induk"></asp:Label>
    </td>
    <td class="SeparatorH"></td>
    <td>
        <asp:TextBox ID="TextBox_Form_NoInduk"
            SkinID="TextboxForm"
            runat="server"></asp:TextBox>
    </td>
</tr>
<tr>
    <td>
        <asp:Label SkinID="LabelDetail" runat="server"

```

```

Text="Nama Lengkap"></asp:Label>
</td>
<td class="SeparatorH"></td>
<td>
    <asp:TextBox ID="TextBox_Form_NamaLengkap"
        SkinID="TextboxForm"
        runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td>
    <asp:Label SkinID="LabelDetail" runat="server"
        Text="Email"></asp:Label>
</td>
<td class="SeparatorH"></td>
<td>
    <asp:TextBox ID="TextBox_Form_Email"
        SkinID="TextboxForm"
        runat="server"></asp:TextBox>
    <asp:RequiredFieldValidator
        ID="RFV_Form_Email"
        ValidationGroup="Default"
        ControlToValidate="TextBox_Form_Email"
        runat="server"
        ErrorMessage="Email harus diisi."
        ToolTip="Email harus diisi.">*
    </asp:RequiredFieldValidator>
</td>
</tr>
<tr>
<td>
    <asp:Label SkinID="LabelDetail" runat="server"
        Text="Organisasi"></asp:Label>
</td>
<td class="SeparatorH"></td>
<td>
    <asp:TextBox ID="TextBox_Form_Org"
        SkinID="TextboxForm"
        runat="server"></asp:TextBox>
    <ajaxToolkit:PopupControlExtender ID="PopEx"
        runat="server"
        TargetControlID="TextBox_Form_Org"
        PopupControlID="Panel_Org"
        Position="Bottom" />
    <asp:Panel ID="Panel_Org"
        BorderColor="#000000" BackColor="#de9439"
        runat="server">
        <div id="PopupPanel">
<asp:TreeView ID="TreeView_Main"
ShowLines="true"
OnSelectedNodeChanged="TreeView_Main_SelectedNodeChanged"
runat="server">
    <SelectedNodeStyle Font-Bold="true" />
</asp:TreeView>
        </div>
    </asp:Panel>
</td>
</tr>
<tr>
<td valign="top">
    <asp:Label SkinID="LabelDetail" runat="server"
        Text="Alamat"></asp:Label>
</td>
<td class="SeparatorH" valign="top"></td>
<td>
    <asp:TextBox ID="TextBox_Form_Alatat"
        SkinID="TextboxForm" TextMode="MultiLine"
        runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td valign="top">
    <asp:Label SkinID="LabelDetail" runat="server"
        Text="Role"></asp:Label>
</td>
<td valign="top" class="SeparatorH"></td>

```

```

                <td>
                    <asp:CheckBoxList ID="CheckBoxList Form Role"
                        runat="server">
                    </asp:CheckBoxList>
                </td>
            </tr>
        </table>
    </asp:View>
    <asp:View ID="View Message" runat="server">
        <div class="Message">
            <asp:Label ID="Label_Message" runat="server"
                Text="Label"></asp:Label>
        </div>
    </asp:View>
</asp:MultiView>
<!-- Message, Detail & Form - End -->
</asp:Panel>
<ajaxToolkit:RoundedCornersExtender ID="rce"
    TargetControlID="Panel_Modal_Content"
    Radius="13"
    Corners="All"
    BorderColor="#040404"
    runat="server" />
</div>
<div class="PanelModalFooter">
    <!-- Modal Windows Action - Start -->
    <div class="PanelModalAction">
        <asp:Button ID="Button_Edit" SkinID="ButtonModal" runat="server"
            Text="Edit" onclick="Button_Edit_Click" />
        <asp:Button ID="Button_Update" SkinID="ButtonModal"
            ValidationGroup="Default" runat="server" Text="Update"
            onclick="Button_Update_Click" />
        <asp:Button ID="Button_Save" SkinID="ButtonModal"
            ValidationGroup="Default" runat="server" Text="Simpan"
            onclick="Button_Save_Click" />
    </div>
    <!-- Modal Windows Action - End -->
</div>
</ContentTemplate>
</asp:UpdatePanel>
</asp:Panel>
<!-- Modal Windows - End -->
</asp:Content>

```

Dan berikut ini adalah isi dari file UserList.aspx.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;

using System.Web.Configuration;
using System.Data;
using System.Data.Linq;

using SPU.DataAccess;
using SPU.Entities;

public partial class UserManagement UserList : System.Web.UI.Page
{
    int itemPerPage =
    Convert.ToInt32(WebConfigurationManager.AppSettings["itemPerPage"].ToString());

    protected void Page_Load(object sender, EventArgs e)
    {
        Label_Title.Text = Page.Title;

        if (!IsPostBack)
        {
            GridView Main.DataBind(0);
        }
    }
}

```

```

}

#region TreeView
public void PopulateNodes()
{
    TreeView_Main.Nodes.Clear();

    DataTable data = GetTreeViewData();
    DataView child = GetParent(data);
    foreach (DataRowView row in child)
    {
        TreeNode childNode = new TreeNode();
        childNode.Text = row["OrganisasiNama"].ToString();
        childNode.Value = row["OrganisasiID"].ToString();
        TreeView_Main.Nodes.Add(childNode);
        AddChild(data, childNode);
    }

    TreeView_Main.ExpandAll();
}

protected DataTable GetTreeViewData()
{
    SPUDataAccess db = new SPUDataAccess();
    DataTable dt = new DataTable();
    DataRow dr;

    dt.Columns.Add("OrganisasiID");
    dt.Columns.Add("OrganisasiInduk");
    dt.Columns.Add("OrganisasiNama");

    var data = db.SPUDataContext.Organisasis.Select(p => p);
    foreach (Organisasi org in data)
    {
        dr = dt.NewRow();
        dr["OrganisasiID"] = org.OrganisasiID;
        dr["OrganisasiInduk"] = org.OrganisasiInduk;
        dr["OrganisasiNama"] = org.OrganisasiNama;
        dt.Rows.Add(dr);
    }

    return dt;
}

DataView GetParent(DataTable parent)
{
    DataView view = new DataView(parent);
    view.RowFilter = "OrganisasiInduk='0'";
    return view;
}

DataView GetChild(DataTable parent, string parentID)
{
    DataView view = new DataView(parent);
    view.RowFilter = "OrganisasiInduk='" + parentID + "'";
    return view;
}

void AddChild(DataTable parent, TreeNode node)
{
    DataView child = GetChild(parent, node.Value);
    foreach (DataRowView row in child)
    {
        TreeNode childNode = new TreeNode();
        childNode.Text = row["OrganisasiNama"].ToString();
        childNode.Value = row["OrganisasiID"].ToString();
        node.ChildNodes.Add(childNode);
        AddChild(parent, childNode);
    }
}

protected void TreeView_Main_SelectedNodeChanged(object sender, EventArgs e)
{
    this.PopEx.Commit(TreeView_Main.SelectedNode);
}

```

```

#endregion TreeView

#region GridView
protected void GridView Main DataBind(int NewPageIndex)
{
    MembershipUserCollection allUser =
        Membership.GetAllUsers();

    GridView Main.PageSize = itemPerPage;
    GridView Main.PageIndex = NewPageIndex;
    GridView_Main.DataSource = allUser;
    GridView_Main.DataBind();
}

protected void GridView Main SelectedIndexChanged(object sender, EventArgs e)
{
    Button_Edit.Visible = true;
    Button_Update.Visible = false;
    Button_Save.Visible = false;
    Label Modal Title.Text = "Detail";
    MultiView Content.SetActiveView(View Detail);

    string username = Convert.ToString(GridView_Main.SelectedValue);

    MembershipUser membershipUser = Membership.GetUser(username);
    if (membershipUser != null)
    {
        Label_Value_Username.Text = membershipUser.UserName;
        Label_Value_Email.Text = membershipUser.Email;
        Label_Value_NoInduk.Text = GetNomorInduk(username);
        Label_Value>Nama_Lengkap.Text = GetFullName(username);
        Label_Value_Alamat.Text = GetAlamat(username);
        Label_Value_Org.Text = GetOrganisasi(username);
        Label_Value_Roles.Text = GetRoles(username);
    }

    UpdatePanel Panel Modal.Update();
    this.MPE.Show();
}

protected void GridView_Main_Sorted(object sender, EventArgs e)
{
    GridView Main DataBind(GridView_Main.PageIndex);
    GridView_Main.SelectedIndex = -1;
}

protected void GridView_Main_PageIndexChanged(object sender, EventArgs e)
{
    GridView Main DataBind(GridView_Main.PageIndex);
    GridView_Main.SelectedIndex = -1;
}

protected void GridView_Main_PageIndexChanging(object sender,
GridViewPageEventArgs e)
{
    GridView_Main_DataBind(e.NewPageIndex);
}
#endregion GridView

#region CheckBoxList
protected void CheckBoxList_Form_Roles_DataBind()
{
    string[] roles = Roles.GetAllRoles();

    CheckBoxList Form Role.DataSource = roles;
    CheckBoxList Form Role.DataBind();
    for (int i = 0; i < CheckBoxList_Form_Role.Items.Count; i++)
    {
        CheckBoxList_Form_Role.Items[i].Selected = false;
    }
}
#endregion CheckBoxList

#region Action
protected void ImageButton_Close_Click(object sender, ImageClickEventArgs e)

```

```

{
    this.MPE.Hide();
}

protected void ImageButton_Delete_Click(object sender, ImageClickEventArgs e)
{
    try
    {
        foreach (GridViewRow gvr in GridView Main.Rows)
        {
            CheckBox cbItem = (CheckBox)gvr.FindControl("CheckBox_Item");
            if (cbItem.Checked)
            {
                int remainDiv = -1;
                int resultDiv = Math.DivRem(gvr.DataItemIndex, itemPerPage,
                out remainDiv);
                if (remainDiv != -1)
                {
                    string _username = Convert.ToString(GridView_Main.
                    DataKeys[remainDiv].Value);
                    MembershipUser MembershipUser =
                    Membership.GetUser( username);
                    if (_MembershipUser != null)
                    {
                        Membership.DeleteUser( MembershipUser.UserName);
                    }
                }
            }
        }
    }
    catch (Exception ex)
    {
    }

    GridView_Main.DataBind(GridView_Main.PageIndex);
    GridView_Main.SelectedIndex = -1;
    UpdatePanel Main.Update();
}

protected void ImageButton_Tambah_Click(object sender, ImageClickEventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = false;
    Button_Save.Visible = true;
    TextBox_Form_Username.Enabled = true;
    RFV_Form_Password.Enabled = true;
    RFV_Form_PasswordC.Enabled = true;
    Label_Modal_Title.Text = "Form";
    MultiView.Content.SetActiveView(View_Form);

    TextBox_Form_Username.Text = String.Empty;
    TextBox_Form_Password.Text = String.Empty;
    TextBox_Form_PasswordC.Text = String.Empty;
    TextBox_Form_NoInduk.Text = String.Empty;
    TextBox_Form>NamaLengkap.Text = String.Empty;
    TextBox_Form_Email.Text = String.Empty;
    TextBox_Form_Org.Text = String.Empty;
    TextBox_Form_Alamat.Text = String.Empty;

    CheckBoxList_Form_Roles.DataBind();
    PopulateNodes();

    UpdatePanel_Panel_Modal.Update();
    this.MPE.Show();
}

protected void Button_Save_Click(object sender, EventArgs e)
{
    string username = TextBox_Form_Username.Text;
    string roles = String.Empty;

    Membership.CreateUser(username, TextBox_Form_Password.Text,
    TextBox_Form_Email.Text);

    ProfileCommon profileCommon = Profile.GetProfile(username);
    profileCommon.Alamat = TextBox_Form_Alamat.Text;
}

```

```

        profileCommon.NamaLengkap = TextBox_Form_NamaLengkap.Text;
        profileCommon.NomorInduk = TextBox_Form_NoInduk.Text;
        profileCommon.OrganisasiID = TextBox_Form_Org.Text;
        profileCommon.Save();

        for (int i = 0; i < CheckBoxList_Form_Role.Items.Count; i++)
        {
            if (CheckBoxList_Form_Role.Items[i].Selected)
            {
                if (!String.IsNullOrEmpty(roles)) roles += ",";
                roles += CheckBoxList_Form_Role.Items[i].Value;
            }
            CheckBoxList_Form_Role.Items[i].Selected = false;
        }
        Roles.AddUserToRoles(username, roles.Split(','));

        GridView_Main_DataBind(GridView_Main.PageIndex);
        GridView_Main.SelectedIndex = -1;
        UpdatePanel_Main.Update();
        this.MPE.Hide();
    }

    protected void Button_Update_Click(object sender, EventArgs e)
    {
        Button_Edit.Visible = true;
        Button_Update.Visible = false;
        Button_Save.Visible = false;
        Label_Modal_Title.Text = "Detail";
        MultiView_Content.SetActiveView(View_Detail);

        string username = GridView_Main.SelectedValue.ToString();
        MembershipUser membershipUser = Membership.GetUser(username);
        if (membershipUser != null)
        {
            membershipUser.Email = TextBox_Form_Email.Text;
            Membership.UpdateUser(membershipUser);

            if (!String.IsNullOrEmpty(TextBox_Form_Password.Text))
            {
                membershipUser.ChangePassword(membershipUser.ResetPassword(),
                TextBox_Form_Password.Text);
            }

            ProfileCommon profileCommon = Profile.GetProfile(username);
            profileCommon.Alat = TextBox_Form_Alat.Text;
            profileCommon.NamaLengkap = TextBox_Form_NamaLengkap.Text;
            profileCommon.NomorInduk = TextBox_Form_NoInduk.Text;
            profileCommon.OrganisasiID = TextBox_Form_Org.Text;
            profileCommon.Save();

            for (int i = 0; i < CheckBoxList_Form_Role.Items.Count; i++)
            {
                if (CheckBoxList_Form_Role.Items[i].Selected.Equals(false))
                {
                    if (Roles.IsUserInRole(username,
                    CheckBoxList_Form_Role.Items[i].Value))
                    {
                        Roles.RemoveUserFromRole(username,
                        CheckBoxList_Form_Role.Items[i].Value);
                    }
                    else
                    {
                        if (!Roles.IsUserInRole(username,
                        CheckBoxList_Form_Role.Items[i].Value))
                        {
                            Roles.AddUserToRole(username,
                            CheckBoxList_Form_Role.Items[i].Value);
                        }
                    }
                }
            }

            membershipUser = Membership.GetUser(username);
            if (membershipUser != null)
            {
                Label_Value_Username.Text = membershipUser.UserName;
            }
        }
    }

```

```

        Label_Value_Email.Text = membershipUser.Email;
        Label_Value_NoInduk.Text = GetNomorInduk(username);
        Label_Value_NamaLengkap.Text = GetFullName(username);
        Label_Value_Alamat.Text = GetAlamat(username);
        Label_Value_Org.Text = GetOrganisasi(username);
        Label_Value_Roles.Text = GetRoles(username);
    }

    GridView_Main.DataBind(GridView_Main.PageIndex);
    UpdatePanel_Main.Update();
}

protected void Button_Edit_Click(object sender, EventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = true;
    Button_Save.Visible = false;
    TextBox_Form_Username.Enabled = false;
    RFV_Form_Password.Enabled = false;
    RFV_Form_PasswordC.Enabled = false;
    Label_Modal_Title.Text = "Form";
    MultiView_Content.SetActiveView(View_Form);

    CheckBoxList_Form_Roles.DataBind();
    PopulateNodes();

    string username = GridView_Main.SelectedValue.ToString();
    MembershipUser membershipUser = Membership.GetUser(username);

    TextBox_Form_Username.Text = membershipUser.UserName;
    TextBox_Form_Email.Text = membershipUser.Email;
    TextBox_Form_NoInduk.Text = GetNomorInduk(username);
    TextBox_Form_NamaLengkap.Text = GetFullName(username);
    TextBox_Form_Alamat.Text = GetAlamat(username);
    TextBox_Form_Org.Text = GetOrganisasiId(username);

    String[] arrRoles = Roles.GetRolesForUser(membershipUser.UserName);
    CheckBoxList_Form_Role.ClearSelection();
    for (int i = 0; i < arrRoles.Length; i++)
    {
        ListItem li = CheckBoxList_Form_Role.Items.FindByValue(arrRoles[i]);
        if (li != null)
        {
            CheckBoxList_Form_Role.Items.FindByValue(arrRoles[i]).Selected =
true;
        }
    }
}
#endregion Action

#region Additional Methods
protected String GetFullName(string username)
{
    ProfileCommon _ProfileCommon = Profile.GetProfile(username);
    string fullname = _ProfileCommon.NamaLengkap;
    if (String.IsNullOrEmpty(fullname.Trim())) return "-";

    return fullname;
}

protected String GetNomorInduk(string username)
{
    ProfileCommon _ProfileCommon = Profile.GetProfile(username);
    string nomorInduk = _ProfileCommon.NomorInduk;
    if (String.IsNullOrEmpty(nomorInduk.Trim())) return "-";

    return nomorInduk;
}

protected String GetAlamat(string username)
{
    ProfileCommon _ProfileCommon = Profile.GetProfile(username);
    string alamat = _ProfileCommon.Alamat;
    if (String.IsNullOrEmpty(alamat.Trim())) return "-";
}

```



```

        return alamat;
    }

    protected String GetOrganisasi(string username)
    {
        ProfileCommon _ProfileCommon = Profile.GetProfile(username);
        string orgID = _ProfileCommon.OrganisasiID;
        string orgNama = String.Empty;

        try
        {
            SPUDataAccess db = new SPUDataAccess();
            Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(orgID)).Single<Organisasi>();

            while (!org.OrganisasiInduk.Equals("0"))
            {
                orgNama = org.OrganisasiNama + "<br/>" + orgNama;
                org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(org.OrganisasiInduk)).Single<Organisasi>();
            }
        }
        catch
        {
            orgNama = "-";
        }

        return orgNama;
    }

    protected String GetOrganisasiId(string username)
    {
        ProfileCommon ProfileCommon = Profile.GetProfile(username);
        string orgID = _ProfileCommon.OrganisasiID;

        return orgID;
    }

    public String GetRoles(object username)
    {
        String returnVal = String.Empty;

        try
        {
            String[] arrRoles = Roles.GetRolesForUser((String)username);
            for (int i = 0; i < arrRoles.Length; i++)
            {
                returnVal += arrRoles[i] + ", ";
            }
            int lenghtVal = returnVal.Length;
            returnVal = returnVal.Substring(0, returnVal.Length - 2);
        }
        catch { }

        return returnVal;
    }
    #endregion Additional Methods
}

```

Pada barisan kode di atas, terdapat control GridView yang berfungsi untuk menampilkan data dalam bentuk tabular. Selain itu juga terdapat kontrol Tab, dari AJAX Control Toolkit yang berfungsi untuk membuat tab. Yang terakhir adalah control PopupControl, dari AJAX Control Toolkit yang dapat membuat textbox terlihat lebih menarik seperti yang dapat dilihat pada gambar 76.

Penjelasan Kode

Berikut ini adalah penjelasan dari barisan kode yang telah dituliskan di atas.

Daftar User

Untuk menampilkan data user, maka perlu ditambahkan control GridView pada file UserList.aspx. Berikut ini adalah kode-kode dari GridView.

```
<asp:GridView ID="GridView Main" ClientIDMode="Static" AutoGenerateColumns="false"
    DataKeyNames="UserName"
    AllowPaging="true" AllowSorting="true" Width="100%" BorderWidth="1"
    OnSelectedIndexChanged="GridView_Main_SelectedIndexChanged"
    OnSorted="GridView_Main_Sorted"
    OnPageIndexChanged="GridView Main PageIndexChanged"
    OnPageIndexChanging="GridView Main PageIndexChanging"
    runat="server">
    <EmptyDataTemplate>
        <div class="EmptyDataMessage">
            <asp:Label ID="Label_GridView_Empty" runat="server"
                Text="Data tidak ditemukan."></asp:Label>
        </div>
    </EmptyDataTemplate>
    <HeaderStyle CssClass="GridviewHeader" />
    <SelectedRowStyle CssClass="GridviewSelect" />
    <AlternatingRowStyle CssClass="GridviewAlternate" />
    <Columns>
        <asp:TemplateField>
            <HeaderTemplate>
                <div style="text-align:center">i</div>
            </HeaderTemplate>
            <ItemTemplate>
                <asp:CheckBox ID="CheckBox Item" runat="server" />
            </ItemTemplate>
            <ItemStyle HorizontalAlign="Center" VerticalAlign="Top" />
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Aksi" ItemStyle-CssClass="GridviewItem"
            ItemStyle-VerticalAlign="Top" ItemStyle-HorizontalAlign="Center"
            HeaderStyle-Width="5%" ItemStyle-Wrap="false">
            <ItemTemplate>
                <asp:LinkButton ID="LinkButton_Select" CommandName="Select"
                    runat="server">Pilih</asp:LinkButton>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="UserName" HeaderText="Username"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-Width="15%" ItemStyle-
            VerticalAlign="Top" />
        <asp:TemplateField HeaderText="Nomor Induk" ItemStyle-VerticalAlign="Top"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-Width="15%">
            <ItemTemplate>
                <%#
                    GetNomorInduk((String)DataBinder.Eval(Container.DataItem, "UserName"))
                %>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Name Lengkap" ItemStyle-VerticalAlign="Top"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-Width="20%">
            <ItemTemplate>
                <%#
                    GetFullName((String)DataBinder.Eval(Container.DataItem, "UserName"))
                %>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:BoundField DataField="Email" HeaderText="Email"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-Width="20%" ItemStyle-
            VerticalAlign="Top" />
        <asp:TemplateField HeaderText="Role" ItemStyle-VerticalAlign="Top"
            ItemStyle-CssClass="GridviewItem" HeaderStyle-Width="25%">
            <ItemTemplate>
                <%#
                    GetRoles((String)DataBinder.Eval(Container.DataItem, "UserName"))
                %>
            </ItemTemplate>
        </asp:TemplateField>
    </Columns>
    <PagerSettings Mode="NumericFirstLast">
```

```

FirstPageImageUrl="~/App_Themes/SPU/icons/icon_nav_first_on.gif"
FirstPageText="Pertama"
LastPageImageUrl="~/App_Themes/SPU/icons/icon_nav_last_on.gif"
LastPageText="Akhir"
NextPageImageUrl="~/App_Themes/SPU/icons/icon_nav_next_on.gif"
PreviousPageImageUrl="~/App_Themes/SPU/icons/icon_nav_prev_on.gif" />
<PagerStyle HorizontalAlign="Center" VerticalAlign="Middle"
ForeColor="#543000" />
</asp:GridView>

```

Pada kode diatas dapat dilihat atribut **ClientIDMode**, atribut ini adalah fitur terbaru yang dapat ditemui pada ASP.NET. Atribut ini tidak hanya terdapat pada control `GridView` saja, tetapi pada server control yang dimiliki oleh ASP.NET.

Sebagai informasi, `id` pada control ASP.NET ada kemungkinan akan berbeda saat di-render menjadi halaman HTML. Jadi apabila nilai `id` control `GridView` di atas adalah `GridView_Main`, maka setelah halaman `UserList.aspx` dilihat pada web browser, dan dilihat kode HTML dari halaman ini maka ada kemungkinan tidak akan ditemui `id=GridView_Main`.

Fitur ini memungkinkan developer untuk menentukan bagaimana `id` dari control tersebut di-render pada halaman HTML. Dengan memberikan nilai `Static` pada atribut ini maka `id` yang dilihat setelah di-render adalah sama dengan nilai `id` yang telah diberikan, yaitu `GridView_Main`.

Atribut-atribut lain dari `GridView` adalah :

- `AutoGenerateColumn` yang bernilai `false`, yang berarti developer sendiri yang akan mengatur antarmuka dari `GridView` seperti jumlah kolom yang akan ditampilkan, ukuran, nama header yang akan ditulis dan style-style dari header dan item untuk menampilkan data.
- `DataKeyNames` berisi nama field yang menjadi primary key dari data yang akan ditampilkan.
- `AllowingPaging` untuk menentukan penggunaan fitur *pagging*..
- `AllowingSorting` untuk menentukan penggunaan fitur *sorting* data.

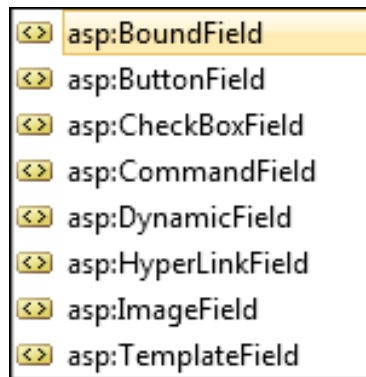
`GridView` juga mempunyai event yang dapat digunakan, pada kode di atas dapat dilihat event-event seperti berikut ini :

- `OnSelectedIndexChanged`, terjadi ketika index data berubah. Hal ini dapat terjadi ketika record data pada `GridView` dipilih.
- `OnSorted`, terjadi ketika aksi *sorting* dilakukan.
- `OnPageIndexChanged`, terjadi ketika aksi perpindahan halaman telah dilakukan.
- `OnPageIndexChanging`, terjadi ketika aksi perpindahan halaman sedang dilakukan.

Masing-masing dari event ini akan memanggil sebuah method untuk menangani aksi yang dilakukan.

Pada control `GridView` di atas juga dapat dilihat atribut `EmptyDataTemplate`, atribut ini berisi pesan atau antarmuka yang akan ditampilkan apabila data kosong.

Atribut `Column` pada `GridView` berisi item-item untuk menampilkan data. Ada beberapa cara untuk menampilkan data, yaitu seperti pada gambar di bawah ini.



Gambar 81. Control untuk menampilkan data di dalam GridView.

Pada kode GridView di atas dapat dilihat control-control berikut ini :

- a. `TemplateField`, control ini memungkinkan developer mengisi kolom dengan lebih bebas. Kolom dapat diisi dengan control lain seperti `CheckBox`, `LinkButton` atau memanggil method seperti yang dicontohkan pada barisan kode di atas. Yaitu memanggil `GetNomorInduk`, `GetFullName` dan `GetRoles`.
- b. `BoundField`, control ini untuk menampilkan data dengan cara yang telah disediakan dan lebih mudah.

Yang terakhir adalah atribut `PagerSetting`, yang berfungsi untuk mengatur antarmuka *pager*.

Selanjutnya adalah mengisi GridView dengan data user. Pada file `UserList.aspx.cs` dapat ditemui method `GridView_Main_DataBind` dengan isi sebagai berikut.

```
protected void GridView_Main_DataBind(int NewPageIndex)
{
    MembershipUserCollection allUser = Membership.GetAllUsers();

    GridView_Main.PageSize = itemPerPage;
    GridView_Main.PageIndex = NewPageIndex;
    GridView_Main.DataSource = allUser;
    GridView_Main.DataBind();
}
```

Dari baris di atas, untuk mengambil data user cukup dengan baris yang diberi tanda kuning. Selanjutnya tinggal menjadi objek `allUser` sebagai *datasource* dari GridView. Dan untuk menampilkan data pada GridView atau ketika ingin menyegarkan tampilan GridView setelah terjadi proses operasi pada data (seperti : tambah, update dan hapus data).

Detail User

Modal windows akan ditampilkan ketika event `GridView_Main_SelectedIndexChanged` terjadi, dan berikut isi dari method ini.

```
protected void GridView_Main_SelectedIndexChanged(object sender, EventArgs e)
{
    Button_Edit.Visible = true;
    Button_Update.Visible = false;
    Button_Save.Visible = false;
    Label_Modal_Title.Text = "Detail";
    MultiView.Content.SetActiveView(View_Detail);

    string username = Convert.ToString(GridView_Main.SelectedValue);

    MembershipUser membershipUser = Membership.GetUser(username);
    if (membershipUser != null)
```

```

{
    Label Value Username.Text = membershipUser.UserName;
    Label Value Email.Text = membershipUser.Email;
    Label Value NoInduk.Text = GetNomorInduk(username);
    Label_Value_NamaLengkap.Text = GetFullName(username);
    Label_Value_Alamat.Text = GetAlamat(username);
    Label_Value_Org.Text = GetOrganisasi(username);
    Label Value Roles.Text = GetRoles(username);
}

UpdatePanel_Panel_Modal.Update();
this.MPE.Show();
}

```

Langkah pertama adalah mengambil primary key data, sesuai dengan nilai pada atribut DataKeyNames. Cara untuk mengambilnya adalah seperti baris dengan tanda warna kuning. Sedangkan untuk mengambil data berdasarkan nama user tertentu dapat dilihat pada baris berwarna hijau.

Data yang dimiliki oleh objek membershipUser hanya sedikit, diantaranya adalah UserName dan Email. Sedangkan data seperti NoInduk, NamaLengkap, Alamat dan OrganisasiID disimpan dalam Profile. Untuk melihat bagaimana cara mengambil data pada Profile dapat dilihat pada method-method berikut ini.

```

protected String GetFullName(string username)
{
    ProfileCommon ProfileCommon = Profile.GetProfile(username);
    string fullname = ProfileCommon.NamaLengkap;
    if (String.IsNullOrEmpty(fullname.Trim())) return "-";

    return fullname;
}

protected String GetNomorInduk(string username)
{
    ProfileCommon ProfileCommon = Profile.GetProfile(username);
    string nomorInduk = ProfileCommon.NomorInduk;
    if (String.IsNullOrEmpty(nomorInduk.Trim())) return "-";

    return nomorInduk;
}

protected String GetAlamat(string username)
{
    ProfileCommon ProfileCommon = Profile.GetProfile(username);
    string alamat = ProfileCommon.Alamat;
    if (String.IsNullOrEmpty(alamat.Trim())) return "-";

    return alamat;
}

protected String GetOrganisasi(string username)
{
    ProfileCommon ProfileCommon = Profile.GetProfile(username);
    string orgID = ProfileCommon.OrganisasiID;
    string orgNama = String.Empty;

    try
    {
        SPUDataAccess db = new SPUDataAccess();
        Organisasi org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(orgID)).Single<Organisasi>();

        while (!org.OrganisasiInduk.Equals("0"))
        {
            orgNama = org.OrganisasiNama + "<br/>" + orgNama;
            org = db.SPUDataContext.Organisasis.Where(p =>
p.OrganisasiID.Equals(org.OrganisasiInduk)).Single<Organisasi>();
        }
    }
    catch

```

```

    {
        orgNama = "-";
    }

    return orgNama;
}

protected String GetOrganisasiId(string username)
{
    ProfileCommon ProfileCommon = Profile.GetProfile(username);
    string orgID = ProfileCommon.OrganisasiID;

    return orgID;
}

```

Untuk mengambil data Alamat, NoInduk, OrganisasiID dan NamaLengkap dari Profile dapat dilihat pada baris-baris berwarna biru.

Sedangkan untuk mengambil data *Role* yang dimiliki oleh user, dapat dilihat pada method berikut ini, yaitu baris yang berwarna merah muda.

```

public String GetRoles(object username)
{
    String returnVal = String.Empty;

    try
    {
        String[] arrRoles = Roles.GetRolesForUser((String)username);
        for (int i = 0; i < arrRoles.Length; i++)
        {
            returnVal += arrRoles[i] + ", ";
        }
        int lenghtVal = returnVal.Length;
        returnVal = returnVal.Substring(0, returnVal.Length - 2);
    }
    catch { }

    return returnVal;
}

```

Edit User

Setelah detail ditampilkan, selanjutnya adalah mengedit user. Dengan menekan tombol Edit, maka akan dieksekusi method berikut ini.

```

protected void Button Edit Click(object sender, EventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = true;
    Button_Save.Visible = false;
    TextBox Form Username.Enabled = false;
    RFV Form Password.Enabled = false;
    RFV_Form_PasswordC.Enabled = false;
    Label_Modal_Title.Text = "Form";
    MultiView Content.SetActiveView(View Form);

    CheckBoxList Form Roles.DataBind();
    PopulateNodes();

    string username = GridView_Main.SelectedValue.ToString();
    MembershipUser membershipUser = Membership.GetUser(username);

    TextBox Form Username.Text = membershipUser.UserName;
    TextBox Form Email.Text = membershipUser.Email;
    TextBox_Form_NoInduk.Text = GetNomorInduk(username);
    TextBox_Form_NamaLengkap.Text = GetFullName(username);
    TextBox_Form_Alamat.Text = GetAlamat(username);
    TextBox_Form_Org.Text = GetOrganisasiId(username);
}

```

```
String[] arrRoles = Roles.GetRolesForUser(membershipUser.UserName);
CheckBoxList Form Role.ClearSelection();
for (int i = 0; i < arrRoles.Length; i++)
{
    ListItem li = CheckBoxList_Form_Role.Items.FindByValue(arrRoles[i]);
    if (li != null)
    {
        CheckBoxList Form Role.Items.FindByValue(arrRoles[i]).Selected = true;
    }
}
}
```

Pada method ini akan dipanggil method `CheckBoxList_Form_Roles_DataBind` untuk mengisi data *Role* pada control `CheckBoxList`. Sedangkan method `PopulateNodes` berfungsi untuk mengisi data Organisasi pada control `TreeView`, sehingga dapat dilihat form seperti berikut ini.

Gambar 82. Control CheckBoxList dan TreeView.

Selanjutnya menekan tombol `Update`, dan method `Button_Update_Click` akan dieksekusi, berikut ini adalah isi dari method ini.

```
protected void Button_Update_Click(object sender, EventArgs e)
{
    Button Edit.Visible = true;
    Button_Update.Visible = false;
    Button_Save.Visible = false;
    Label_Modal_Title.Text = "Detail";
    MultiView Content.SetActiveView(View_Detail);

    string username = GridView_Main.Selected.Value.ToString();
    MembershipUser membershipUser = Membership.GetUser(username);
    if (membershipUser != null)
    {
        membershipUser.Email = TextBox_Form_Email.Text;
        Membership.UpdateUser(membershipUser);

        if (!String.IsNullOrEmpty(TextBox_Form_Password.Text))
        {
            membershipUser.ChangePassword(membershipUser.ResetPassword(),
            TextBox_Form_Password.Text);
        }

        ProfileCommon profileCommon = Profile.GetProfile(username);
        profileCommon.Alamat = TextBox_Form_Alamat.Text;
        profileCommon>NamaLengkap = TextBox_Form>NamaLengkap.Text;
        profileCommon.NomorInduk = TextBox_Form>NoInduk.Text;
        profileCommon.OrganisasiID = TextBox_Form_Org.Text;
        profileCommon.Save();

        for (int i = 0; i < CheckBoxList_Form_Role.Items.Count; i++)
        {
            if (CheckBoxList_Form_Role.Items[i].Selected.Equals(false))
            {

```

```

        if (Roles.IsUserInRole(username,
CheckBoxList_Form_Role.Items[i].Value))
        {
            Roles.RemoveUserFromRole(username,
CheckBoxList_Form_Role.Items[i].Value);
        }
    }
    else
    {
        if (!Roles.IsUserInRole(username,
CheckBoxList_Form_Role.Items[i].Value))
        {
            Roles.AddUserToRole(username,
CheckBoxList_Form_Role.Items[i].Value);
        }
    }
}

membershipUser = Membership.GetUser(username);
if (membershipUser != null)
{
    Label_Value_Username.Text = membershipUser.UserName;
    Label_Value_Email.Text = membershipUser.Email;
    Label_Value_NoInduk.Text = GetNomorInduk(username);
    Label_Value>Nama_Lengkap.Text = GetFullName(username);
    Label_Value_Alamat.Text = GetAlamat(username);
    Label_Value_Org.Text = GetOrganisasi(username);
    Label_Value_Roles.Text = GetRoles(username);
}

GridView_Main.DataBind(GridView_Main.PageIndex);
UpdatePanel_Main.Update();
}
}

```

Pada barisan kode di atas terdapat beberapa baris yang penting, yaitu :

- Baris berwarna merah muda, baris ini berfungsi untuk mengupdate data user, dalam kasus ini data yang diupdate adalah email.
- Sedangkang apabila ingin mengubah password maka caranya dapat dilihat pada baris yang berwarna merah tua.
- Untuk menyimpan perubahan data pada Profile dapat dilihat pada baris berwarna biru.
- Untuk menambah atau menghapus Role yang dimiliki oleh seorang user dapat dilihat caranya pada baris-baris berwarna coklat.

Tambah User

Form untuk menambah user akan ditampilkan ketika tombol Tambah yang ada pada sisi kanan ditekan dan akan mengeksekusi method berikut ini.

```

protected void ImageButton_Tambah_Click(object sender, ImageClickEventArgs e)
{
    Button_Edit.Visible = false;
    Button_Update.Visible = false;
    Button_Save.Visible = true;
    TextBox_Form_Username.Enabled = true;
    RFV_Form_Password.Enabled = true;
    RFV_Form_PasswordC.Enabled = true;
    Label_Modal_Title.Text = "Form";
    MultiView_Content.SetActiveView(View_Form);

    TextBox_Form_Username.Text = String.Empty;
    TextBox_Form_Password.Text = String.Empty;
    TextBox_Form_PasswordC.Text = String.Empty;
    TextBox_Form_NoInduk.Text = String.Empty;
    TextBox_Form>Nama_Lengkap.Text = String.Empty;
}

```



```

        TextBox_Form_Email.Text = String.Empty;
        TextBox_Form_Org.Text = String.Empty;
        TextBox_Form_Alamat.Text = String.Empty;

        CheckBoxList_Form_Roles.DataBind();
        PopulateNodes();

        UpdatePanel_Panel_Modal.Update();
        this.MPE.Show();
    }

```

Dan antarmuka seperti ini akan ditampilkan.

Gambar 83. Form tambah.

Setelah form diisi dan tombol Simpan ditekan maka selanjutnya akan dieksekusi method berikut ini.

```

protected void Button_Save_Click(object sender, EventArgs e)
{
    string username = TextBox_Form_Username.Text;
    String roles = String.Empty;

    Membership.CreateUser(username, TextBox_Form_Password.Text,
        TextBox_Form_Email.Text);

    ProfileCommon profileCommon = Profile.GetProfile(username);
    profileCommon.Alamat = TextBox_Form_Alamat.Text;
    profileCommon.NamaLengkap = TextBox_Form_NamaLengkap.Text;
    profileCommon.NomorInduk = TextBox_Form_NoInduk.Text;
    profileCommon.OrganisasiID = TextBox_Form_Org.Text;
    profileCommon.Save();

    for (int i = 0; i < CheckBoxList_Form_Role.Items.Count; i++)
    {
        if (CheckBoxList_Form_Role.Items[i].Selected)
        {
            if (!String.IsNullOrEmpty(roles)) roles += ",";
            roles += CheckBoxList_Form_Role.Items[i].Value;
        }
        CheckBoxList_Form_Role.Items[i].Selected = false;
    }
    Roles.AddUserToRoles(username, roles.Split(','));

    GridView_Main.DataBind(GridView_Main.PageIndex);
    GridView_Main.SelectedIndex = -1;
    UpdatePanel_Main.Update();
    this.MPE.Hide();
}

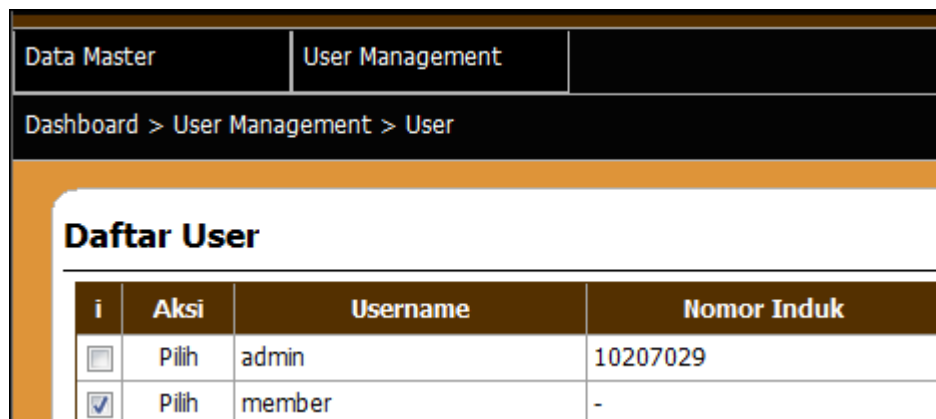
```

Ada tiga hal penting dari kode di atas, yaitu :

- Membuat user, caranya dapat dilihat pada baris berwarna kuning.
- Setelah user dibuat, maka data dapat disimpan pada Profile dengan cara seperti pada baris-baris berwarna hijau.
- Selanjutnya adalah menambah Role pada user yang dapat dilakukan dengan menulis baris berwarna biru.

Hapus User

Untuk menghapus data user, terlebih dahulu harus dipilih user yang akan dihapus dengan mencentang CheckBox pada user yang akan dihapus. Setelah hal itu dilakukan, user dapat menekan tombol Hapus yang terdapat pada sisi kanan.



Gambar 84. CheckBox pada daftar user.

Setelah tombol Hapus ditekan, maka akan dijalankan method berikut ini untuk menghapus user yang telah dipilih.

```
protected void ImageButton Delete Click(object sender, ImageClickEventArgs e)
{
    try
    {
        foreach (GridViewRow gvr in GridView_Main.Rows)
        {
            CheckBox cbItem = (CheckBox)gvr.FindControl("CheckBox Item");
            if (cbItem.Checked)
            {
                int remainDiv = -1;
                int resultDiv = Math.DivRem(gvr.DataItemIndex, itemPerPage, out
remainDiv);
                if (remainDiv != -1)
                {
                    string _username =
Convert.ToString(GridView_Main.DataKeys[remainDiv].Value);
MembershipUser MembershipUser =
Membership.GetUser(_username);
                    if (MembershipUser != null)
                    {
                        Membership.DeleteUser(_MembershipUser.UserName);
                    }
                }
            }
        }
    }
    catch (Exception ex)
    {
    }
}

GridView_Main_DataBind(GridView_Main.PageIndex);
```

```

GridView_Main.SelectedIndex = -1;
UpdatePanel Main.Update();
}

```

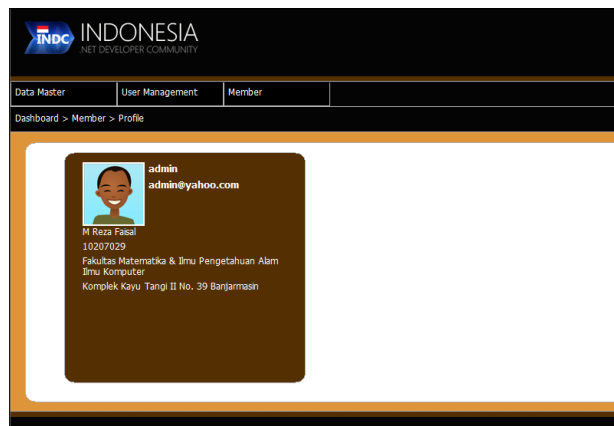
Pada method ini terdapat dua langkah penting untuk menghapus user, yaitu :

- Mengambil objek user seperti yang terlihat pada baris berwarna kuning.
- Menghapus user dengan cara yang dituliskan pada baris berwarna hijau.

Profile User

Profile user akan disimpan pada file `Profile.aspx`. Halaman ini untuk menampilkan profile user yang sedang login. Bila user yang login adalah admin, maka user ini akan melihat user dirinya sendiri.

Tampilan dari halaman `Profile.aspx` ini adalah sebagai berikut.



Gambar 85. Profile user.

Isi dari file `Profile.aspx` adalah sebagai berikut.

```

<%@ Page Title="" Language="C#"
MasterPageFile="~/Shared/MasterPages/Member.master" AutoEventWireup="true"
CodeFile="Profile.aspx.cs" Inherits="Dashboard_Profile" %>

<asp:Content ID="Content1" ContentPlaceHolderID="Head Member" Runat="Server">
    <style type="text/css">
        .style1
        {
            width: 4px;
        }
    </style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder Member"
Runat="Server">
    <div id="PanelContainer" style="padding-top:0;margin-top:0;color:#ffffff">
        <asp:Panel ID="Panel_Login" CssClass="PanelLogin" runat="server">
            <div id="LoginContainer">
                <div id="AvatarContainer">
                    <asp:Image ID="Image Avatar"
                        ImageUrl="~/App_Themes/SPU/avatar/avatar.png"
                        runat="server" />
                </div>
                <div id="UserDataContainer">
                    <div class="UserDataItem">
                        <asp:Label ID="Label_Value_Username" SkinID="Profile"
                            runat="server" Text="Label"></asp:Label>
                    </div>
                    <div class="UserDataItem">
                        <asp:Label ID="Label_Value_Email" SkinID="Profile"

```

```

        runat="server" Text="Label"></asp:Label>
    </div>
</div>
<div class="Separator"></div>
<div class="UserDataItem">
    <asp:Label ID="Label_Value_NameLengkap" runat="server"
        Text="Label"></asp:Label>
</div>
<div class="UserDataItem">
    <asp:Label ID="Label Value NoInduk" runat="server"
        Text="Label"></asp:Label>
</div>
<div class="UserDataItem">
    <asp:Label ID="Label Value Org" runat="server"
        Text="Label"></asp:Label>
</div>
<div class="UserDataItem">
    <asp:Label ID="Label Value Alamat" runat="server"
        Text="Label"></asp:Label>
</div>
</div>
</asp:Panel>
<ajaxToolkit:RoundedCornersExtender ID="rce"
    TargetControlID="Panel_Login"
    Radius="13"
    Corners="All"
    BorderColor="#b3b3b3"
    runat="server" />
</div>
</asp:Content>

```

Dan berikut ini adalah isi dari file Profile.aspx.cs.

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.Security;

using System.Web.Configuration;
using System.Data;
using System.Data.Linq;

using SPU.DataAccess;
using SPU.Entities;

public partial class Dashboard Profile : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        bool isLogin = HttpContext.Current.User.Identity.IsAuthenticated;
        if (isLogin)
        {
            string username = HttpContext.Current.User.Identity.Name;
            MembershipUser user = Membership.GetUser(username);

            Label_Value_Username.Text = username;
            Label Value Email.Text = user.Email;
            Label Value NameLengkap.Text = GetFullName(username);
            Label Value NoInduk.Text = GetNomorInduk(username);
            Label_Value_Org.Text = GetOrganisasi(username);
            Label_Value_Alamat.Text = GetAlamat(username);
        }
    }

    #region Additional Methods
    protected String GetFullName(string username)
    {
        ProfileCommon _ProfileCommon = Profile.GetProfile(username);
        string fullname = _ProfileCommon>NamaLengkap;
        if (String.IsNullOrEmpty(fullname.Trim())) return "-";
    }
}

```

```

        return fullname;
    }

    protected String GetNomorInduk(string username)
    {
        ProfileCommon _ProfileCommon = Profile.GetProfile(username);
        string nomorInduk = _ProfileCommon.NomorInduk;
        if (String.IsNullOrEmpty(nomorInduk.Trim())) return "-";

        return nomorInduk;
    }

    protected String GetAlamat(string username)
    {
        ProfileCommon ProfileCommon = Profile.GetProfile(username);
        string alamat = ProfileCommon.Alat;
        if (String.IsNullOrEmpty(alamat.Trim())) return "-";

        return alamat;
    }

    protected String GetOrganisasi(string username)
    {
        ProfileCommon _ProfileCommon = Profile.GetProfile(username);
        string orgID = ProfileCommon.OrganisasiID;
        string orgNama = String.Empty;

        try
        {
            SPUDataAccess db = new SPUDataAccess();
            Organisasi org = db.SPUDataContext.Organisasi.Where(p =>
                p.OrganisasiID.Equals(orgID)).Single<Organisasi>();

            while (!org.OrganisasiInduk.Equals("0"))
            {
                orgNama = org.OrganisasiNama + "<br/>" + orgNama;
                org = db.SPUDataContext.Organisasi.Where(p =>
                    p.OrganisasiID.Equals(org.OrganisasiInduk)).Single<Organisasi>();
            }
        }
        catch
        {
            orgNama = "-";
        }

        return orgNama;
    }
}
#endregion Additional Methods
}

```

Yang penting di file ini adalah cara untuk mengambil nilai username dari user yang sedang login. Caranya cukup dengan menggunakan baris yang berwarna kuning.

Apa yang telah dipelajari?

Pada bab ini ada beberapa hal yang dipelajari :

- Menggunakan class `Membership` untuk mengelola user.
- Mengelola atribut tambahan pada data user dengan menggunakan `Profile`.
- Menampilkan data dengan `GridView`.
- Control `PopupControl`, `AJAX Control Toolkit`.
- Control `Tab`, `AJAX Control Toolkit`.
- Cara mengambil username yang login.

Pada bab ini adalah langkah akhir dalam pembuatan Sistem Pengelolaan User ini. Ada beberapa yang akan dibahas pada bab ini diantaranya adalah :

- a. Security.
- b. Globalization.

Security

Pada bab Pembangunan Fondasi, control Menu sudah menampilkan menu sesuai *Role* yang dimiliki oleh user yang login, seperti yang telah di atur pada file Web.sitemap, seperti yang bisa dilihat pada gambar di bawah ini.

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0" >
  <siteMapNode url="" title="Dashboard" description="Dashboard" roles="">

    <siteMapNode url="" title="Data Master" description="Data Master" roles="Administrator">
      <siteMapNode url="~/DataMaster/OrganisasiList.aspx" title="Organisasi" description="Daftar Organisasi" roles="Administrator"/>
    </siteMapNode>

    <siteMapNode url="" title="User Management" description="User Management" roles="Administrator">
      <siteMapNode url="~/UserManagement/UserList.aspx" title="User" description="Daftar User" roles="Administrator"/>
    </siteMapNode>

    <siteMapNode url="" title="Member" description="Member" roles="Member">
      <siteMapNode url="~/Dashboard/Profile.aspx" title="Profile" description="Profile" roles="Member"/>
    </siteMapNode>

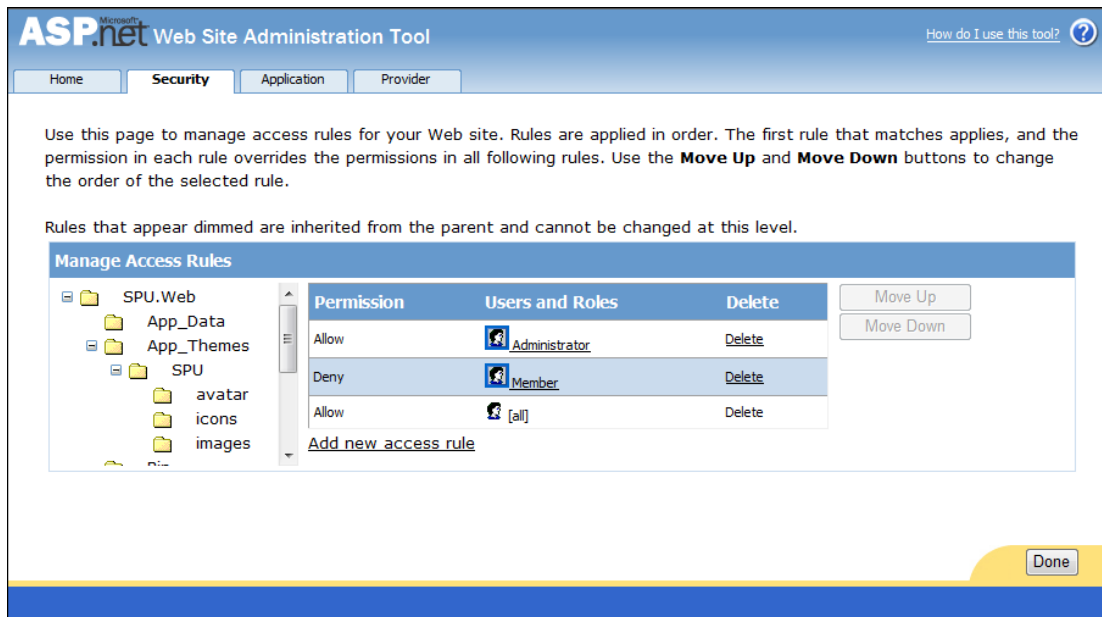
  </siteMapNode>
</siteMap>
```

Gambar 86. Web.sitemap.

Dari gambar di atas, dapat diketahui bahwa user dengan Role Member hanya dapat melihat menu Member saja dan tidak bisa melihat menu Data Master dan User Management. Hal ini bukan berarti user dengan Role Member tidak bisa mengakses halaman pada menu tersebut. Bila user ini menulis langsung alamat menuju file `OrganisasiList.aspx` dan `UserList.aspx` pada address bar, maka user tersebut dapat mengakses kedua halaman tersebut.

Untuk menghindari hal ini terjadi, maka perlu pembenahan pada pengaturan hak akses. ASP.NET sudah memiliki fitur yang dapat mengatur hak akses atau Access Rule. Pengaturan hak akses meliputi pengaturan hak untuk mengakses folder pada Web Site Project atau Web Application Project. Bila suatu folder diatur untuk tidak bisa diakses oleh user tertentu atau role tertentu maka secara otomatis file-file yang ada pada folder tersebut tidak dapat diakses pula.

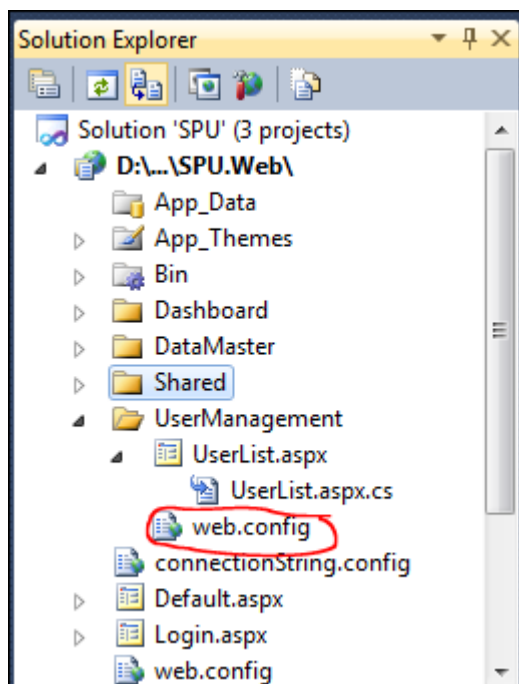
Untuk melakukan pengaturan hak akses ini dapat dilakukan dengan mengakses halaman ASP.NET Web Site Administration Tool.



Gambar 87. Mengelola Access Rules.

Hasil dari konfigurasi yang dilakukan di halaman ini adalah file `web.config` pada folder yang sedang diatur hak aksesnya.

Sebagai contoh, apabila hak akses yang diatur adalah pada folder `UserManagement` agar tidak bisa diakses oleh user dengan role `Member`, tetapi bisa di akses oleh user dengan role `Administrator` maka dapat dilihat folder `web.config` pada folder ini.



Gambar 88. Web.config untuk mengatur folder.

Dan berikut ini adalah isi dari file `web.config` ini.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <authorization>
      <allow roles="Administrator" />
    </authorization>
  </system.web>
</configuration>
```

```
<deny roles="Member" />
</authorization>
</system.web>
</configuration>
```

Dari isi file di atas dapat dilihat bahwa user dengan role Administrator dapat mengakses folder ini tapi tidak untuk user yang hanya memiliki role Member.

Apabila dimiliki user yang mempunyai role Administrator dan Member, maka user tersebut masih dapat mengakses folder ini, hal ini terjadi karena urutan authorisasi seperti di atas.

Apabila isi file `web.config` di atas diubah menjadi seperti berikut ini, maka user yang mempunyai role Administrator dan Member tersebut tidak akan dapat melakukan akses ke folder tersebut.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <authorization>
      <deny roles="Member" />
      <allow roles="Administrator" />
    </authorization>
  </system.web>
</configuration>
```

Hal itu disebabkan aturan di atas akan menolak user dengan role Member (`<deny roles="Member" />`) terlebih dahulu, maka user tersebut akan langsung dilemparkan keluar folder ini tanpa sempat membaca aturan yang memperbolehkan user dengan role Administrator (`<allow roles="Administrator" />`).

Dari contoh di atas maka dapat diketahui bahwa urutan penulisan hak akses sangat berpengaruh.

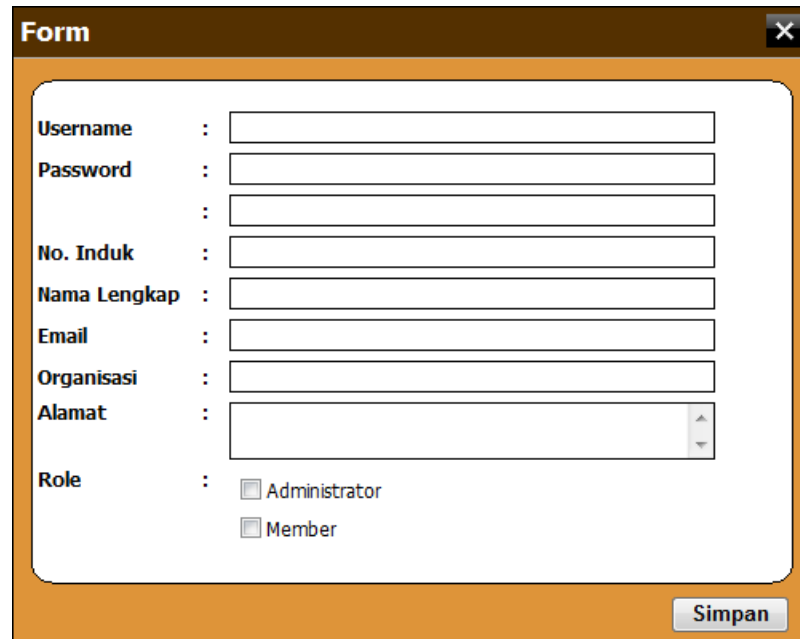
Contoh lain penulisan hak akses dapat dilihat pada isi file `web.config` di bawah ini.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <authorization>
      <allow user="Reza" />
      <deny user="*" />
    </authorization>
  </system.web>
</configuration>
```

Dari contoh di atas, folder ini hanya bisa diakses oleh user Reza tapi tidak untuk user lainnya.

Resource dan Globalization

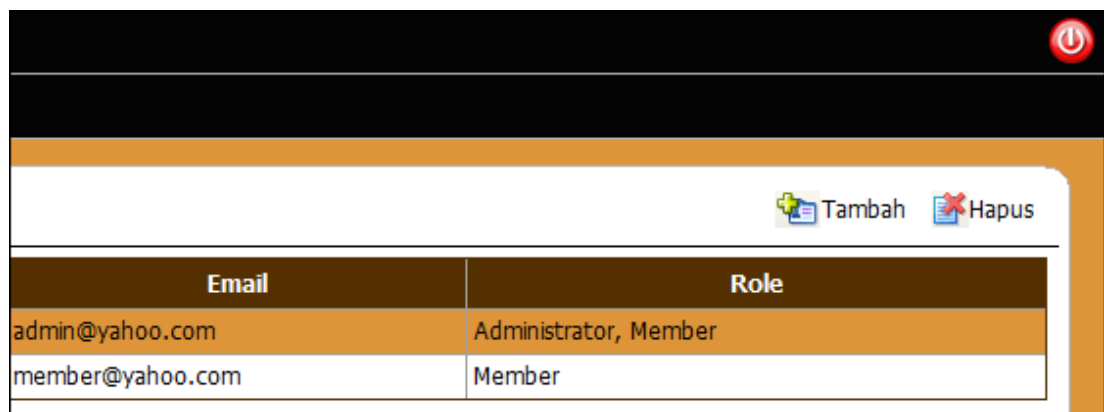
Berikut ini adalah antarmuka yang telah dibuat.



The screenshot shows a window titled "Form" with a close button in the top right corner. Inside the window, there are several input fields and checkboxes. The fields are labeled: Username, Password, No. Induk, Nama Lengkap, Email, Organisasi, and Alamat. Each label is followed by a colon and an input field. The "Role" field has two checkboxes: "Administrator" and "Member". At the bottom right of the window is a button labeled "Simpan".

Gambar 89. Form tambah user.

Atau antarmuka berikut ini.



The screenshot shows a user management interface. At the top right is a power button. Below it is a table with two columns: "Email" and "Role". There are two rows of data. Above the table are two buttons: "Tambah" (Add) and "Hapus" (Delete).

Email	Role
admin@yahoo.com	Administrator, Member
member@yahoo.com	Member

Gambar 90. Daftar user.

Bila aplikasi web ini hanya digunakan oleh user yang mengerti bahasa Indonesia tentu tidak akan ada kendala dalam menggunakannya. Namun bagaimana bisa ternyata aplikasi web ini dilirik oleh pengguna dari belahan dunia lain yang tidak mengerti bahasa Indonesia, maka harus ada perubahan yang dilakukan oleh developer agar aplikasi web ini dapat digunakan oleh pengguna lain yang tidak menggunakan bahasa Indonesia.

ASP.NET mempunyai fitur yang memungkinkan text-text yang ditampilkan pada halaman-halaman aplikasi web dapat diganti dengan mudah. Fitur ini memungkinkan developer menyimpan text-text yang akan ditampilkan tersebut dalam sebuah 'kamus', jadi text-text tersebut tidak dituliskan langsung pada halaman aplikasi web.

'Kamus' yang disebutkan di atas dapat disebut *Resource*. Resource mempunyai dua tipe, yaitu :

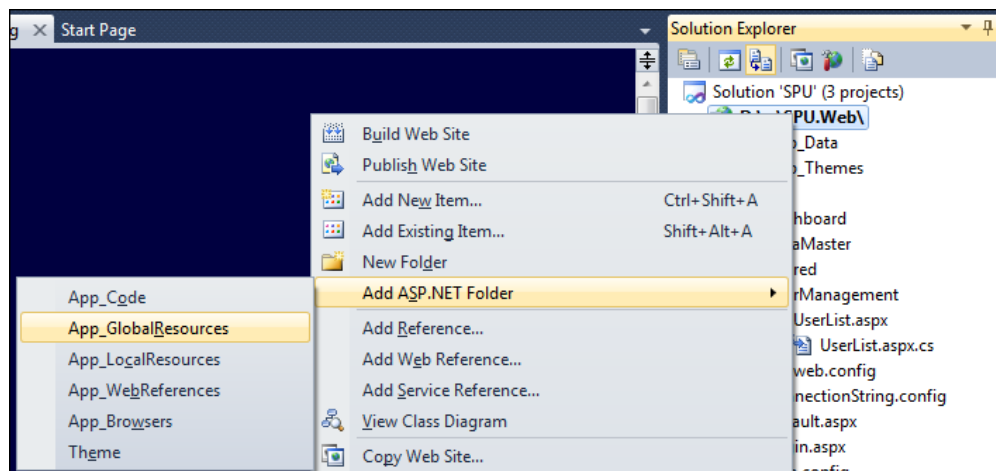
- a. Global Resource, data yang disimpan akan dapat digunakan pada semua halaman.
- b. Local Resource, data hanya dapat digunakan oleh satu halaman saja.

Global Resource

Pada gambar 89 dan gambar 90, dapat ditunjuk text seperti Simpan pada Button, text Tambah dan Hapus pada Label adalah text-text yang dapat disimpan sebagai Global Resource. Hal ini dikarenakan control Button dengan text Tambah atau control Label dengan text Tambah dan Hapus akan dapat ditemui pada banyak halaman.

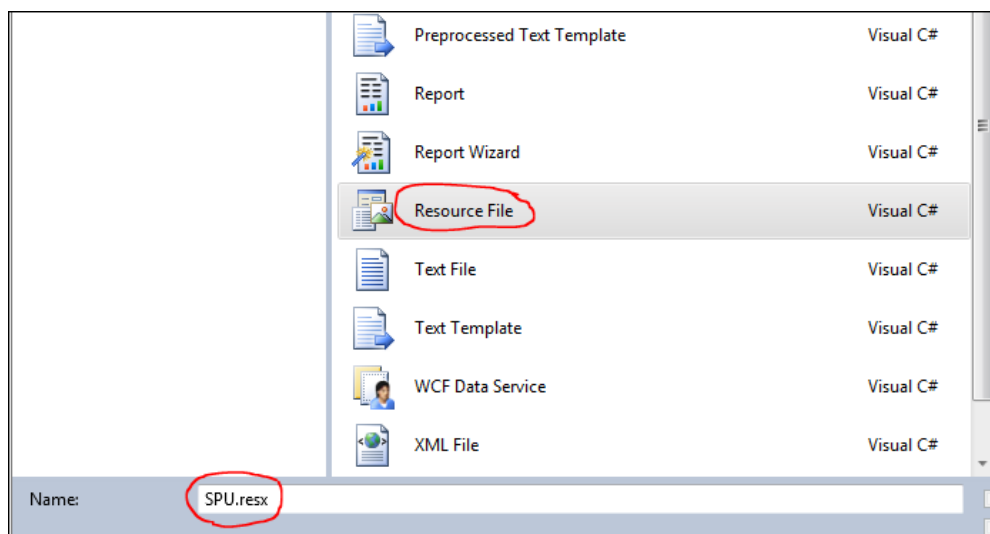
Untuk membuat 'kamus' yang bersifat global dapat dilakukan dengan melakukan langkah-langkah berikut ini.

Membuat folder App_GlobalResources dengan melakukan klik kanan pada Web Site Project kemudian pilih Add ASP.NET Folder > App_GlobalResources.



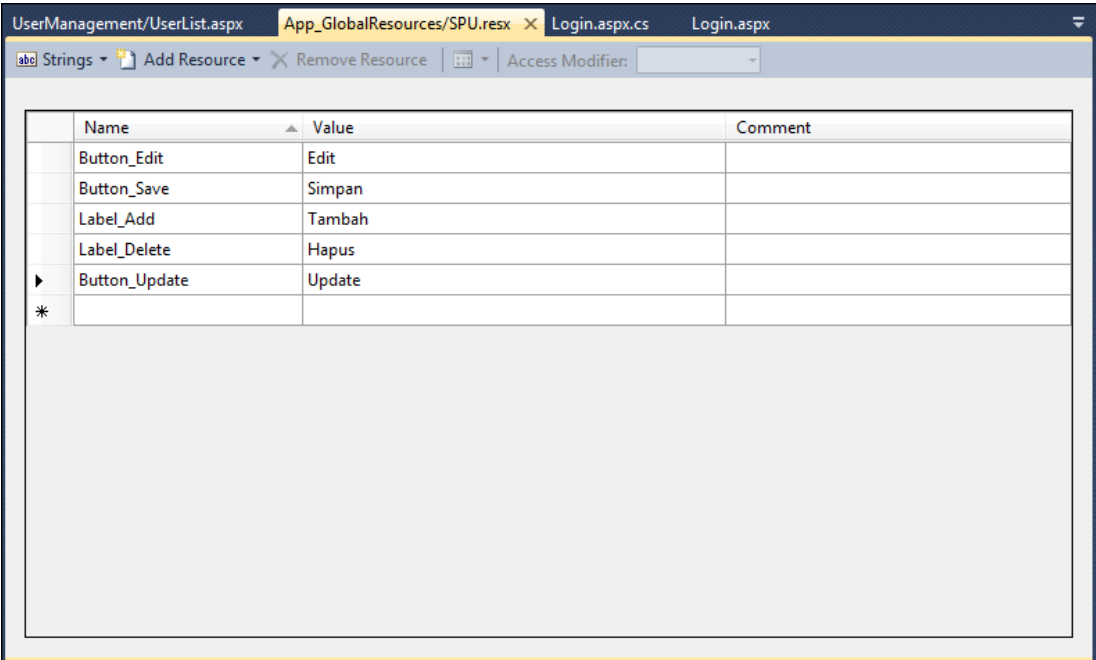
Gambar 91. Menambah folder App_GlobalResources.

Selanjutnya adalah menambahkan file Resource dengan nama SPU.resx pada folder tersebut.



Gambar 92. Membuat file SPU.resx.

Bila file `SPU.resx` dibuka maka akan dilihat tampilan seperti berikut ini.



	Name	Value	Comment
	Button_Edit	Edit	
	Button_Save	Simpan	
	Label_Add	Tambah	
	Label_Delete	Hapus	
▶	Button_Update	Update	
*			

Gambar 93. Isi file SPU.resx

Field Name, adalah kunci yang dapat dipanggil untuk mengambil nilai yang tersimpan pada field Value.

Berikut ini adalah contoh penggunaan Global Resource untuk mengganti atribut Text pada Button.

```
<asp:Button ID="Button_Edit"
    SkinID="ButtonModal"
    runat="server"
    Text="<%%$ Resources:SPU, Button_Edit %>"
    onclick="Button Edit Click" />

<asp:Button ID="Button_Update"
    SkinID="ButtonModal"
    ValidationGroup="Default"
    runat="server"
    Text="<%%$ Resources:SPU, Button_Update %>"
    onclick="Button Update Click" />

<asp:Button ID="Button_Save"
    SkinID="ButtonModal"
    ValidationGroup="Default"
    runat="server"
    Text="<%%$ Resources:SPU, Button_Save %>"
    onclick="Button Save Click" />
```

Dan berikut ini perubahan nilai atribut Text pada control-control Label.

```
<asp:Label ID="Label_Delete"
    runat="server"
    AssociatedControlID="ImageButton_Delete"
    Text="<%%$ Resources:SPU, Label_Delete %>"
/>

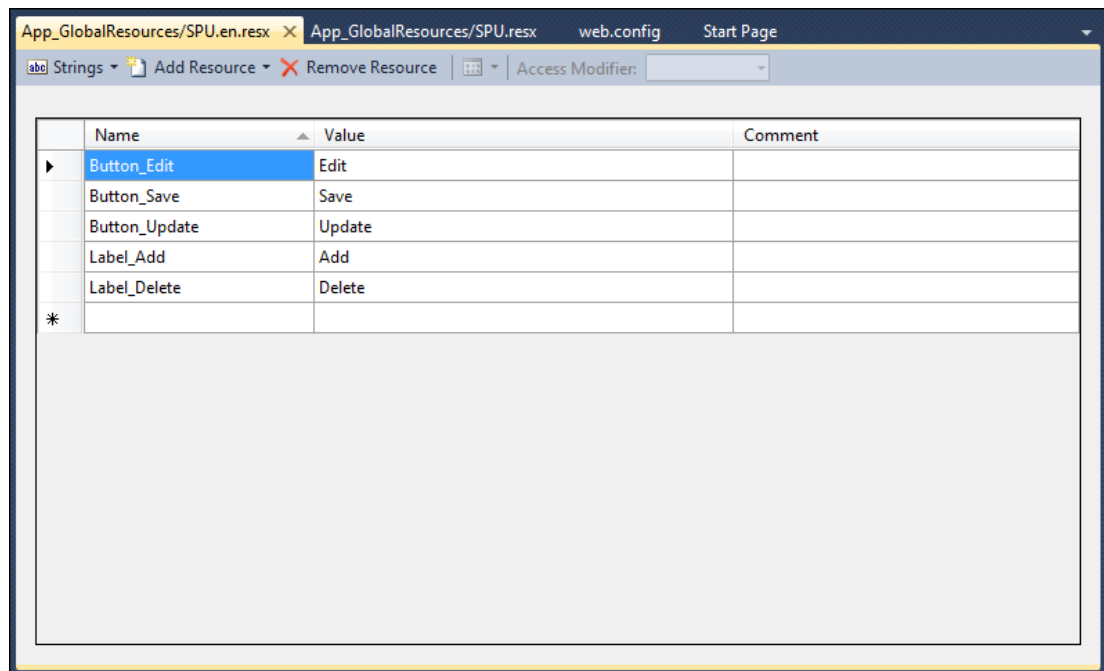
<asp:Label ID="Label_Tambah"
    AssociatedControlID="ImageButton_Tambah"
    runat="server"
    Text="<%%$ Resources:SPU, Label_Add %>"
/>
```

Dari contoh di atas, cara untuk menggunakan Global Resource dapat dilihat pada baris-baris berwarna kuning. Pada baris tersebut ada dua pemanggilan, yaitu :

- a. Nama file Global Resource, yaitu `SPU`.
- b. Nilai Name yang disimpan pada Global Resource, seperti `Button_Edit`, `Button_Update` dan `Button_Save`.

File `SPU.resx` adalah file Global Resource utama yang akan dibaca untuk memberikan nilai-nilai pada atribut-atribut `Text` yang diinginkan. File `SPU.resx` ini berisi nilai-nilai berbahasa Indonesia. Bila ingin membuat 'kamus' berbahasa Inggris maka perlu dibuat sebuah file Global Resource lagi dengan nama `SPU.en.resx`.

Berikut adalah isi dari file `SPU.en.resx`.



Gambar 94. Global resource bahasa Inggris.

Setelah file ini dibuat dan halaman di-refresh (dalam kasus ini perubahan dilakukan pada file `UserList.aspx`) maka dapat dilihat perubahan bahasa menjadi bahasa Inggris. Hal ini terjadi karena Sistem Operasi dan Web Browser yang digunakan setting bahasa Inggris.

Agar halaman dapat menampilkan text dalam bahasa Indonesia maka perlu dibuat file dengan nama `SPU.id.resx` dengan isi seperti pada gambar 93.

Setelah file `SPU.id.resx` dibuat, langkah selanjutnya adalah 'mengaktifkan file ini' agar control `Button` dan `Label` menampilkan teks dengan bahasa Indonesia. Caranya adalah dengan menambahkan baris berikut ini pada file `web.config`, di dalam `<system.web>` ...

```
<globalization uiCulture="id" culture="id-ID"/>
```

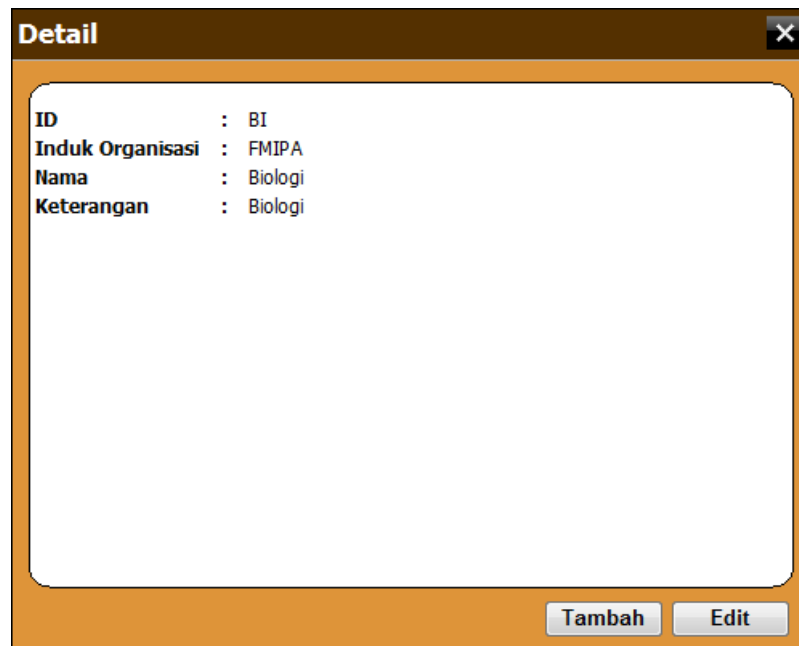
Setelah baris di atas ditambah dan halaman `UserList.aspx` di-refresh maka akan dapat dilihat teks pada `Label` dan `Button` akan berbahasa Indonesia kembali sesuai dengan isi file `SPU.id.resx`.

Bila ingin menggunakan bahasa yang ditulis pada file `SPU.en.resx` maka cukup dengan mengubah baris di atas menjadi seperti berikut ini.

```
<globalization uiCulture="en" culture="en-US"/>
```

Local Resource

Local Resource adalah 'kampus' yang bersifat lokal dan hanya berlaku untuk satu halaman ASPX saja. Sebagai contoh antarmuka di bawah ini.



The screenshot shows a web application window titled "Detail" with a close button (X) in the top right corner. The window has an orange border and a white background. Inside, there is a table with the following data:

ID	: BI
Induk Organisasi	: FMIPA
Nama	: Biologi
Keterangan	: Biologi

At the bottom right of the window, there are two buttons: "Tambah" and "Edit".

Gambar 95. Detail Organisasi

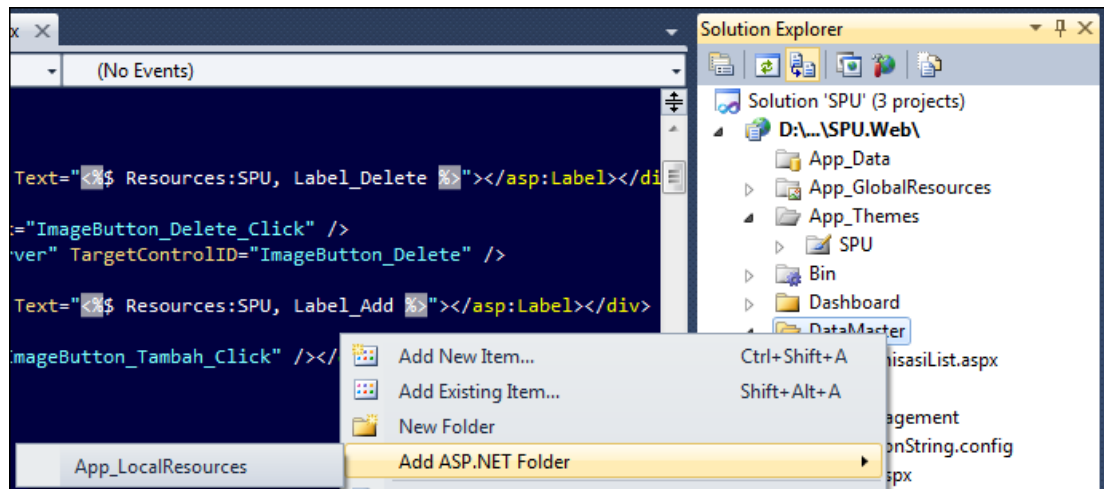
Antarmuka di atas adalah bagian dari file `OrganisasiList.aspx`, pada halaman ini dapat dilihat control-control `Label` dengan teks, yaitu :

- ID.
- Induk Organisasi.
- Nama.
- Keterangan.

Karena control `Label` dengan atribut `Text` tersebut hanya ada di file `OrganisasiList.aspx` maka tidak perlu nilai-nilai tersebut tidak perlu dimasukkan ke dalam Global Resource, dalam kasus ini adalah file `SPU.resx`, `SPU.en.resx` dan `SPU.id.resx`.

Nilai-nilai ini cukup dimasukkan ke dalam 'kamus lokal' saja yang disebut Local Resource. Untuk membuat Local Resource yang akan digunakan oleh file `OrganisasiList.aspx` maka harus dilakukan langkah-langkah berikut ini.

Buat folder `App_LocalResources` di dalam folder `DataMaster`, dengan cara klik kanan pada folder `DataMaster` kemudian pilih `Add ASP.NET Folder > App_LocalResources`.



Gambar 96. Membuat folder App_LocalResources.

Setelah folder dibuat maka dibuat file Resource dengan nama `OrganisasiList.aspx.resx`, `OrganisasiList.aspx.id.resx` dan `OrganisasiList.aspx.en.resx`. Dapat dilihat bawah aturan untuk membuat file Local Resource adalah sebagai berikut :

```
[nama file].[kode negara].resx
```

Berikut ini adalah isi dari `OrganisasiList.aspx.resx` dan `OrganisasiList.aspx.id.resx`.

DataMaster/App_LocalResources/OrganisasiList.aspx.id.resx			
	Name	Value	Comment
▶	Text_ID	ID	
	Text_OrgDescription	Keterangan	
	Text_OrgName	Nama	
	Text_OrgParent	Induk Organisasi	

Gambar 97. Data di dalam OrganisasiList.aspx.resx dan OrganisasiList.aspx.id.resx.

Dan berikut adalah isi dari file `OrganisasiList.aspx.en.resx`.

DataMaster/App_LocalResources/OrganisasiList.aspx.en.resx			
	Name	Value	Comment
	Text_ID	ID	
	Text_OrgDescription	Description	
▶	Text_OrgName	Name	
	Text_OrgParent	Organization Parent	
*			

Gambar 98. Data di dalam OrganisasiList.aspx.en.resx.

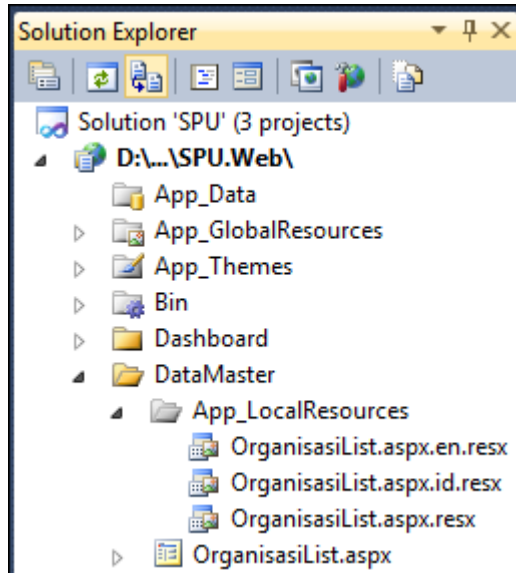
Ada perbedaan penulisan cara penulisan penggunaan antara Global Resource dan Local Resource, berikut ini adalah contoh penggunaan Local Resource.

```
<asp:Label ID="Label_Detail_ID"
    SkinID="LabelDetail"
    runat="server"
    Text="<%$ Resources:Text_ID %>">
```

```
</asp:Label>
```

Dari baris di atas dapat dilihat cara penulisan cara memanggil data yang ada di dalam file Local Resource. Pada baris di atas tidak perlu memanggil nama file Resource seperti halnya cara memanggil file Global Resource.

Berikut ini adalah struktur folder dan file-file untuk mendukung penggunaan Local Resource.



Gambar 99. Struktur folder App_LocalResources.

7

Penutup

Pada ebook ini telah dijelaskan control-control dan fitur-fitur ASP.NET 4.0 dan AJAX Control Toolkit secara singkat dan sederhana. Penjelasan masing-masing fitur dan control tersebut saat digunakan pada pembuatan Sistem Pengelolaan User.

Pembuatan Sistem Pengelolaan User ini akan terdiri dari beberapa tahap, yaitu :

- a. Persiapan database.
- b. Pembuatan fondasi aplikasi web. Pada fondasi aplikasi web ini akan diterangkan konfigurasi yang pertama kali dilakukan agar terkoneksi dengan database yang telah disiapkan. Pada bagian ini juga dipaparkan pembuatan Data Access Layer untuk mempermudah akses database yang akan digunakan pada aplikasi web ini. Selain itu juga dipaparkan untuk melakukan pengamanan terhadap aplikasi web yang telah dibuat serta implementasi fitur Resource dan Globalization untuk membuat aplikasi web yang multi-bahasa.
- c. Pembuatan modul Data Master untuk mengelola data Organisasi dengan menggunakan Data Access Layer yang telah dibuat.
- d. Pembuatan modul User Management dengan menggunakan fitur Membership.
- e. Pembuatan modul Profile yang dapat menampilkan data user yang login.

Sebagai penutup, saya ucapkan terima kasih telah membaca ebook ini. Semoga buku ini dapat bermanfaat bagi web developer untuk membuat aplikasi web.