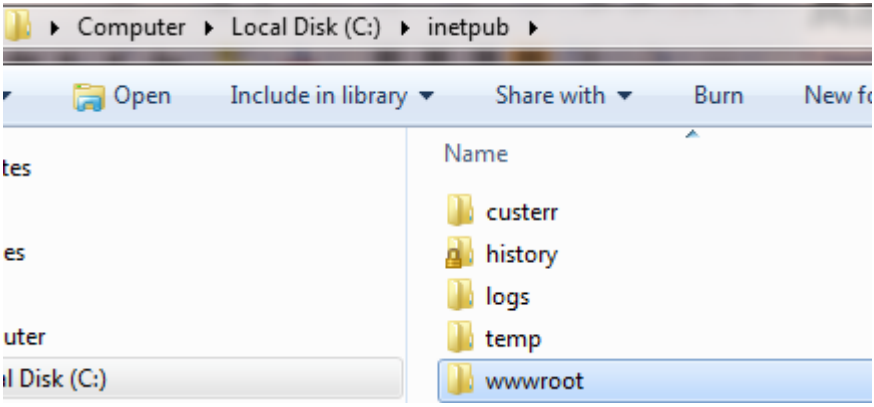
Bab 16

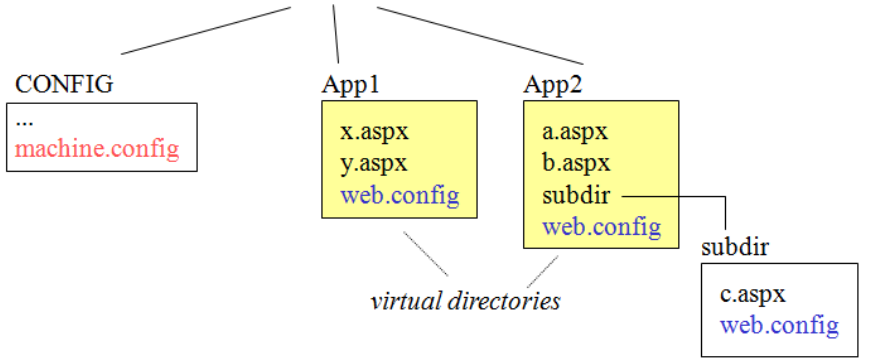
Konfigurasi Keamanan

Sebuah aplikasi dalam kerangka kerja ADO.NET disebarkan pada web server IIS milik Microsoft. Secara standar, direktori penyimpanan file-file aplikasi diletakkan pada direktori c:\inetpub dengan struktur seperti dibawah ini:



Gambar 16.1 Struktur Direktori IIS

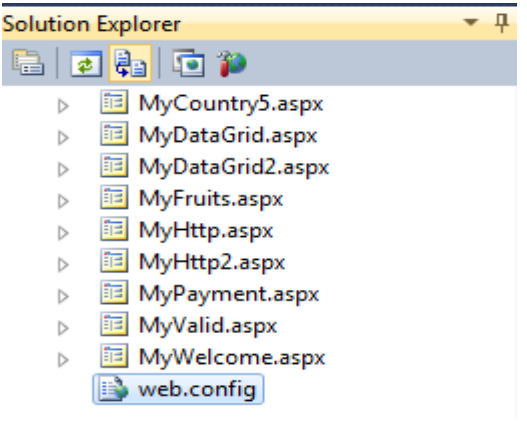
Direktori wwwroot adalah direktori utama yang disediakan oleh IIS. Tempat tsb dapat diberikan hierarki penyimpanan aplikasi dengan beberapa pengaturan konfigurasi. Dua pengaturan ini adalah machine.config dan web.config. Dua file pengatu4ran ini diletakkan pada direktori utama server, seperti yang diperlihatkan pada gambar 16.2



Gambar 16.2 Pengaturan konfigurasi

File machine.config memiliki fungsi sebagai pengaturan global, sehingga semua aplikasi akan mengikuti pengaturan yang terdapat pada global.machine.

Berbeda dengan file machine.config, file web.config berada pada masing-masing aplikasi. File ini dapat meng-*overwrite* aturan yang terdapat pada file config.machine. Pada apliasi VS2010, secara otomatis file web.config dibuat ketika membuat aplikasi situs web, hal ini dapat diperlihatkan pada gambar 16.2



Gambar 16.3 File web.config pada VS2010

Secara standar, isi dari file web.config pada VS 2010, terdapat sekrip pengaturan untuk debug seperti diperlihatkan di bawah 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>
       <system.web>
              <compilation debug="true" targetFramework="4.0"/>
       </system.web>
</configuration>
```

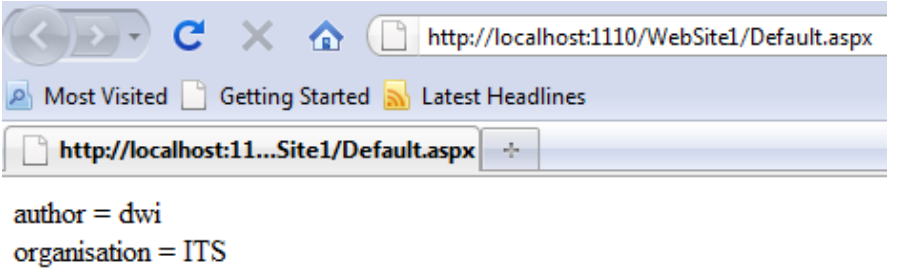
File standar dapat ditambahkan dengan konfigurasi lain seperti setting aplikasi, contoh isi dari file web.config:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <appSettings>
    <add key="author" value="dwi" />
    <add key="organisation" value="ITS" />
  </appSettings>
</configuration>
```

Konfigurasi file web.config ditulis dalam format XML, untuk dapat membaca isi dari file web.config dapat memanfaatkan perintah `ConfigurationSettings.AppSettings.` Contoh untuk membaca file diatas, sekrip tsb dituliskan dalam bahasa C# dibawah ini:

```
<%@Page Language="C#" %>
<%@ Import Namespace="System.Configuration" %>
<html><head></head>
<body>
<%= "author = " +
       ConfigurationSettings.AppSettings["author"] %><br>
<%= "organisation = " +
       ConfigurationSettings.AppSettings["organisation"] %><br>
</body></html>
```

Pada .Net versi yang lebih tinggi, perintah tsb sudah diganti dengan perintah yang lebih komples, namun secara esensi, perintah tsb adalah sama. Hasil pembacaan file web.config diperlihatkan pada gambar 16.4



Gambar 16.4 Pembacaan file web.config

16.1 Autorisasi

Direktori aplikasi harus dilindungi dari pemakai yang tidak berhak. Perlindungan ini dapat memanfaatkan file konfigurasi: global.machine atau web.config

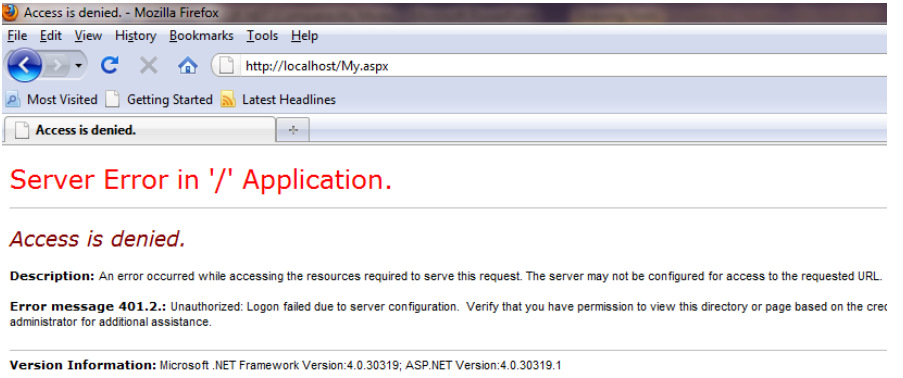
Pengaturan konfigurasi sebagai bentuk autoriasi terhadap siapa yang berkunjung ke situs pada direktori yang ditentukan. Format pengaturan terletak pada kata kunci siapa yang diijinkan yaitu (*allow*) dan siapa yang tidak diperbolehkan (*deny*)

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.web>
    <authorization>
      <allow users="admin" />
      <deny users="?" />
    </authorization>
  </system.web>
</configuration>
```

Sekrip *allow users* dapat berisi lebih dari satu pemakai, penulisan dipisah dengan tanda koma (,) dan terletak pada blok string " …. "

seperti contoh `users="user1, user2, ..."`. Tanda asterisk (*) mewakili keseluruhan, sehingga jika pengaturan user diberikan sebagai `users="*"` berarti secara aturan itu berlaku secara keseluruhan bagi pemakai. Tanda ? dipakai untuk menunjukkan pemakai yang tidak dikenal atau anonymous users. Secara default file konfigurasi global.machine berisi sintak `<allow users="*" />`. Walau secara eksplisit hal itu tidak dituliskan, semua pengguna dapat mengakses situs sampai kemudian ditemukan file web.config yang mengatur siapa saja yang berhak mengkases situs.

Ketika pengaturan konfigurasi diberikan, seseorang pengguna mencoba untuk mengakses sebuah situs, maka sebuah pemberitahuan pesan kesalahan akan muncul seperti yang terlihat pada gambar 16.5:



Gambar 16.5 Pesan Error Autorisasi

16.2 Autentikasi

Pengaturan konfigurasi keamanan dapat membatasi akses pada sumber sumber yang dilindungi. Pada kasus lain, beberapa pemakai dapat memiliki peran yang berbeda sebagai bagian dari tugas mereka, sehingga harus diberikan hak untuk mengakses halaman yang dilindungi dari pemakain umum. Pengataturan peran ini terhadap pemakai satu dengan pemakai lain dikenal sebagai autentifkasi. Terdapat empat jenis autentifikasi yaitu:

1. None
   Tidak ada autentifikasi, yang berarti semua pemakai adalah anonymous users
2. Windows
   Autentifikasi memakai nama user dan password yang dipakai pada login windows. Mode ini biasanya dipakai untuk akses secara local, dimana server secara langsung terhubung ke computer yang mengaksesnya.
3. Passport
   Pemakai akan di *redirect* ke halaman login dari suatu server yang menyimpan informasi pemakai dan passwordnya
4. Forms
   Pemakai di autenfikasi lewat sebuah halaman login

Dari ke-empat jenis autentifikasi, biasanya yang paling sering dipakai adalah tipe Forms. Pemakai dan passwordnya di daftarkan pada file web.config atau pada suatu database. Contoh autenfikasi lewat jenis Forms yang diatur lewat file web.config:

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration>
  <system.web>
    <authorization>
      <deny users="?" />
    </authorization>
    <authentication mode="Forms">
       <forms loginUrl="Login.aspx"
              name="mycookie" protection="All" timeout="20">
        <credentials passwordFormat="MD5">
          <user name="darlis"
              Password="…"/>
          <user name="imam"
              password=".."/>
        </credentials>
      </forms>
    </authentication>
    ...
  </system.web>
  ...
</configuration>
```

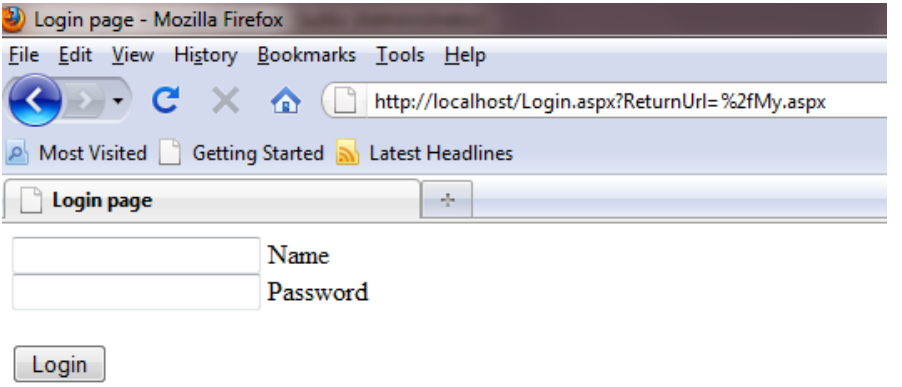
Pengaturan file web.config memakai mode forms dengan loginUrl adalah file Login.aspx dan format password memakai MD5. Pencatatan sesi login ditempatkan pada file mycookie dan siklus umur dari sesi diatur pada 20 menit.

File Login.aspx berisi sebuah inputan nama dan passwordnya yang dicocokkan pada file web.config:

```
<%@ Page Language="C#" %>
<%@ Import Namespace="System.Web.Security" %>
<html><head>
<title>Login page</title>
<script Language="C#" Runat="server">
 void Authenticate (object sender, EventArgs e) {
   if (FormsAuthentication.Authenticate(user.Text, pwd.Text) ||
user.Text == "Dwi")

       FormsAuthentication.RedirectFromLoginPage(user.Text,
false);
       else
              msg.Text = "-- authentication failed";
   }
</script></head>
<body>
<form id="Form1" Runat="server">
       <asp:TextBox ID="user" Runat="server"/> Name<br>
       <asp:TextBox ID="pwd" TextMode="Password"
Runat="server"/> Password<br><br>
       <asp:Button ID="button" Text="Login"
OnClick="Authenticate" Runat="server" />
       <br><br>
       <asp:Label ID="msg" Runat="server" />
</form></body></html>
```

Semua halaman web akan dilindungi dari akses yang tidak berhak, dimana ketika pemakai mencoba mengakses semua halaman web, secara default halaman Login.aspx yang akan muncul pertama kalinya. Contohnya ketika halaman My.aspx akan di akses tanpa melalui login terlebih dahulu, maka server akan menngalihkan permintaan tsb ke halaman Login.aspx seperti yang diperlihatkan pada gambar 16.6



Gambar 16.6 Halaman Login.aspx

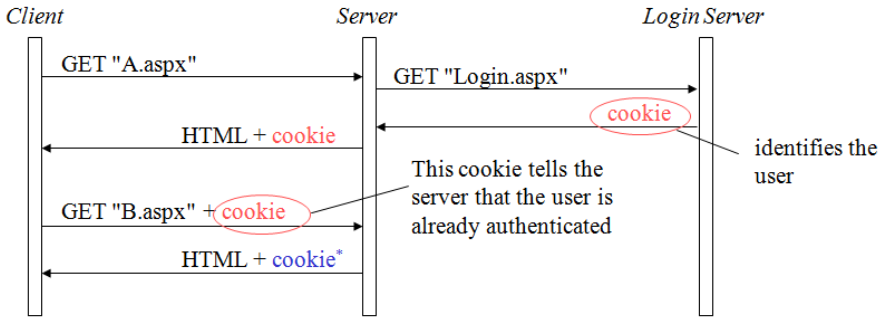
Halaman Login.aspx yang muncul, meminta pemakai untuk mengisikan nama dan password, jika pemakai tidak memilikinya, maka akses ke server ditolak, sebaliknya jika pemakai memiliki user dan password yang sesuai, maka halaman yang diminta akan muncul. Pemakaian tipe enkripsi password MD5 merupakan pilihan enkripsi yang terbanyak dipakai, pembuatan password tsb dapat memakai perintah

```
string encryptedPwd =
       FormsAuthentication.HashPasswordForStoringInConfigFile("
       myPwd", "MD5");
```

Selama waktu yang telah ditentukan, sesi pemakai dicatat, sehingga ketika pemakai berhasil login, maka pada waktu tertentu tsb, sesi akan terus di-ingat.

Pada konfigurasi diatas, semua pemakai yang akan mengakses halaman pada situs dengan server IIS mengalami perlakuan pemeriksaan pada file web.config, jika belum login, maka halaman akan dialihkan ke halaman login. Setelah berhasil login, sesi identifikasi pemakai dicatat pada cookie, server mengembalikan permintaan sebuah halaman dengan sebuah halaman HTML + cookie yang tercatat pada browser pemakai. Ketika pemakai mengakses halaman lain, maka secara otomatis browser akan mencantumkan pula cookie yang telah tercatat

sebelumnya, sehingga server akan menerimanya sebagai autentifikasi yang legal, hal ini diperlihatkan pada gambar 16.7



Gambar 16.7 Siklus Cookie pada Login.aspx

Pengaturan cookie ditempatkan pada file web.config dengan format:

```
<forms loginUrl="Login.aspx" name="mycookie" protection="All"
timeout="20" >
```

Sebuah cookie bernama mycookie, dengan pilihan proteksi secara keseluruhan yang berarti data cookie akan di enkripsi. Pilihan timeout berarti umur dari data cookie tsb adalah 20 menit, setelah itu cookie akan invalid.

16.3 Kontrol Login

Pengaturan autentifikasi melalui halaman login yang telah dijelaskan pada subab sebelumnya, pada .NET versi yang lebih tinggi, pengaturan tsb ditambahkan beberapa kemampuan.

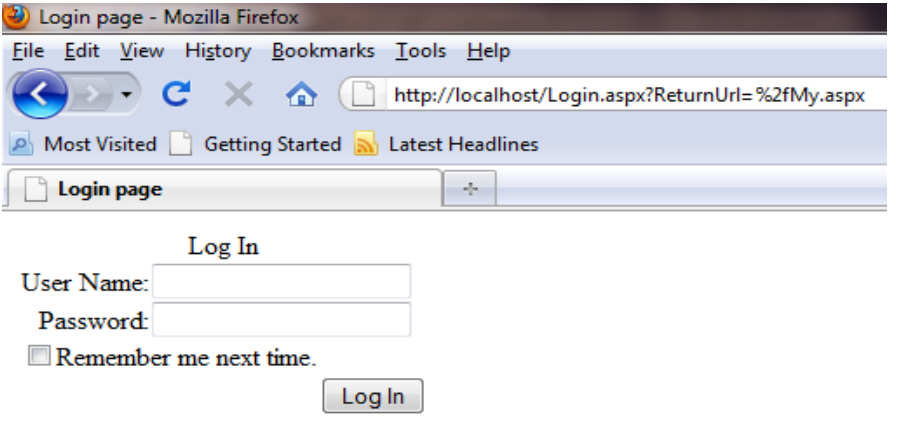
1. **Status Login**
   Sebuah pencatatan pada keadaan sebuah halaman, jika belum login, akan muncul sebuah link referensi ke halaman login, sedangkan ketika sudah login, muncul sebuah referensi ke logout. Format perintah ini adalah: `<asp:LoginStatus Runat="server" />`
2. **Login Name dan Login View**
   Status login yang memperlihatkan sebuah referensi dapat diganti menjadi sebuah informasi berupa text. Format penulisan

```
<asp:LoginView Runat="server">
   <AnonymousTemplate>
     You are not logged in. Click the Login link to sign in.
   </AnonymousTemplate>
   <LoggedInTemplate>
     You are logged in. Welcome,
     <asp:LoginName Runat="server" />!
   </LoggedInTemplate>
</asp:LoginView>
```

Pengaturan autentifikasi pada subab sebelumnya dapat dipersingkat dengan sebuah kontrol login yang telah disediakan pada pustaka ASP.NET 2.0 ke atas. Modifikasi program login.aspx diatas dapat dipersingkat dengan sekrip dibawah ini:

```
<html><head>
<title>Login page</title></head>
<body>
<form id="Form1" Runat="server">
       <asp:Login ID="Login1" Runat="server" />
</form>
</body></html>
```

Sebuah halamam login yang standar akan muncul dengan isian username dan password seperti yang diperlihatkan pada gbr 16.8



Gambar 16.8 Halaman memakai kontrol login

Pustaka System.Web.Security memiliki beberapa fungsi yang dapat dipakai untuk mengatur pemakai. Secara standar pustaka tsb terdapat:

```
static MembershipUser CreateUser(string name, string password)
{...}
static MembershipUser GetUser(string name) {...}
static void UpdateUser(MembershipUser user) {...}
static bool DeleteUser(string name) {...}
static bool ValidateUser(string name, string password) {...}
```