

Part 9 – Cookies And Session

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://suryodesign.asia>

www.suryodesign.wordpress.com

Lisensi Dokumen:

Copyright © 2003-2009 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Cookies Dan Session mungkin jarang di kenali oleh orang yang tidak sering berkecimpung di dunia maya seperti pengguna pasif yang menggunakan internet seperlunya saja , bahkan yang sering berkecimpung pun masih bertanya apa itu Cookies dan Session ? user awam sering melihat di browser seperti mozilla dan internet explorer , disana ada beberapa menu yang bersangkutan tapi mereka tidak mengetahui fitur apakah ini dan apa kegunaannya . disini saya akan mencoba menerangkannya secara step by step berikut dengan contohnya dan juga pengertian dari artikel ini banyak saya ambil dari MSDN karena disana merupakan sumber yang paling lengkap dan good banget buat para pengguna Visual Studio dengan bahasa programming apapun ^^ , bagi yang ingin mendownload artikel ini dapat mendownload di www.suryodesign.asia/freedownload dan bila yang ingin melihat artikel saya yang lainya dapat membaca di sini <http://ilmukomputer.org/author/suryodesign/>

ISI

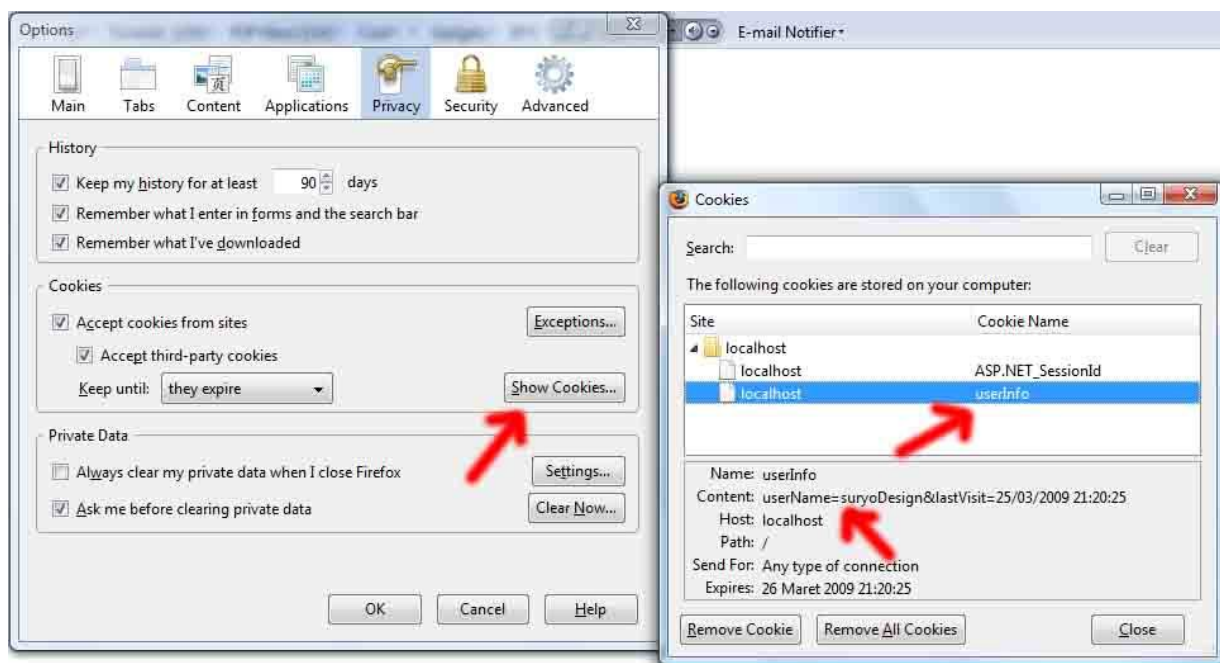
Cookies adalah file berukuran kecil yang dibuat pada sisi client untuk menyimpan data, biasanya bersifat temporary , salah satu keuntungan cookies adalah dapat menyimpan data tanpa user mesti khawatir tentang informasi yang disimpan karena cookies di enkripsi. Dan dapat langsung digunakan dengan mudah di dalam aplikasi , dan mengizinkan untuk di simpan dalam jangka waktu yang panjang . hanya saja cookies ini dibatasi oleh informasi yang bertipe string , file cookies ini tersimpan di computer dan tentu saja user dan menemukannya dan dapat membuka file tersebut , factor ini membuat cookies tidak bagus untuk menyimpan informasi private atau data dalam jumlah besar di dalam cookies.

Ada beberapa user yang men-disable cookies pada browsernya , hal ini dapat menyebabkan masalah bagi web application yang menggunakan cookies. Penggunaan cookies sudah sangat umum untuk jaman skrg ini , karena sudah di adaptasi secara luas oleh kebanyakan website. Tetapi terkadang ada beberapa masalah dengan cookies yang bersifat permanen karena tidak memiliki timer / masa hidupnya karena itu user terkadang harus menghapus secara manual cookies yang tersimpan di dalam hard drive.

Cookies banyak digunakan seperti saat user mengunjungi website kita dan kita menggunakan cookie untuk menyimpan data / keterangan user yang login seperti nama user yang login , dan informasi ini selanjutnya dapat digunakan lagi ketika user mengunjungi web tersebut tanpa mesti login terlebih dahulu. Saat user me-request sebuah halaman , sebenarnya bukan hanya Halaman Page yang dikirim balik ke kita , tetapi juga Cookies yang memiliki tanggal dan waktu , dan ketik itu juga user menerima cookie yang disimpan di dalam folder pada harddisk milik user.

Masing-masing browser memiliki tempat penyimpanan cookies yang berbeda-beda . jadi ketika anda browsing dengan internet explorer dan ada cookies yang dibuat disimpan dalam folder tertentu , maka ketika anda mencoba lagi browsing dengan Mozilla firefox dengan halaman yang sama di buka dengan internet explorer dan browser Mozilla mencoba untuk meload cookies tadi , tetapi Mozilla tidak akan mendapatkan cookies yang tadi karena Mozilla firefox dan internet explorer memiliki tempat penyimpanan cookies pada direktori folder yg berbeda.

Berikut adalah cara untuk melihat Cookies pada browser Mozilla Firefox , dapat dipilih melalui menu **Tools > Option** , dapat anda lihat Cookies tersebut berasal dari website apa , tanggal expired dan isi Content dari Cookies tersebut :



Cookies Limitation

1.Browser pada umumnya mendukung cookies hingga 4096 bytes. Karena masalah limit ini, cookies lebih baik digunakan untuk menyimpan data dalam jumlah yang kecil , atau lebih baik lagi hanya untuk menyimpan identifier , seperti userID dan name , karena userID dapat di gunakan untuk meng-identifikasi user dan membaca data informasi user dari database dengan menggunakan userID tersebut sebagai acuan pencarian data.

2.Browser juga membatasi banyaknya cookies yang dapat di simpan dalam computer, biasanya browser membatasi cookies sebanyak 20 cookies untuk sebuah website , dan bila kita mencoba untuk menyimpan lebih dari itu maka cookies yang umurnya lebih tua maka akan di buang dan diganti dengan yang baru , setiap browser juga memiliki limit penyimpanan cookies, biasanya sekitar 300 cookies untuk semua website yang pernah dikunjungi.

3.Cookies juga memiliki batasan ketika browser user di setting untuk menolak semua cookies yang ada , Sebaiknya kita berhati-hati dalam penggunaan cookies ini , dan jangan pernah membuat aplikasi yang bergantung pada cookies , karena bila cookies tidak disupport oleh browser maka aplikasi anda tentu saja tidak akan berjalan sebagaimana mestinya.

Writing Cookies

Browser bertanggung jawab untuk memanage cookies pada user system , cookies di kirim melalui browser via HttpResponse objek yang menunjukkan sekumpulan data yang disebut dengan Cookies. Kita dapat mengakses HttpResponse object sebagai Response property dari Page class. Ketika kita membuat cookies , kita mesti member Nama dan Value / Nilai pada cookies tersebut. Karena cookies disimpan dengan menggunakan nama , maka bila ada 2 cookies menggunakan nama yang sama , maka cookies dari nilai / value cookies terakhir akan menimpah nilai / value dari cookies sebelumnya. Jadi sebaiknya gunakan nama yang berbeda untuk setiap Cookies. Kita juga dapat mengatur waktu dan tanggal expired atau tanggal habis cookies .

Catatan :

User dapat menghapus cookies pada computer sewaktu-waktu walaupun cookies tersebut di set dengan waktu expired yang panjang sekali dan belum waktunya habis.

Ada 2 cara untuk menulis cookies, cara ke 1 :

```
Response.Cookies["Nama"].Value = "SuryoDesign";  
Response.Cookies["Nama"].Expires = DateTime.Now.AddDays(5);
```

Cara ke 2 untuk menulis cookies :

```
HttpCookie sDesignCookie = new HttpCookie("lastVisit");  
sDesignCookie.Value = DateTime.Now.ToString();  
sDesignCookie.Expires = DateTime.Now.AddDays(1);  
Response.Cookies.Add(sDesignCookie);
```

Dapat kita lihat dari 2 cara di atas , ada 2 cookies yang dibuat , pertama dengan nama "Nama" dan kemudian dengan nama lastVisit , cookie pertama merupakan Cookies collection yang nilainya di set secara langsung (directly) , kita dapat menggunakan cara ini secara langsung karena cookies di turunkan dari specialized collection dengan type NameObjectCollectionBase.

Dan cara ke 2 . kode dibuat dengan instance dari HttpCookie, kemudian kita atur property dari cookies tersebut , seperti tanggal sekarang dan waktu expires selama 1 hari. Dan kemudian kita add cookies tersebut dengan Add Method . ketika kita inisialisasi HttpCookie kita mesti memberi nama Cookie sebagai constructornya .

Jadi dapat kita ambil kesimpulan bahwa 2 cara penulisan memiliki cara yang sama hanya saja yang 1 langsung menggunakan NameObjectCollectionBase cookies dan yang 1 nya kita buat constructor via HTTP , cara yang mana saja terserah anda ingin menggunakan yang mana.

Cookies with More Than One value

Kita dapat menyimpan nilai di dalam cookie seperti kita buat sebelumnya yaitu Nama dan lastVisit , kita juga dapat membuat multiple name-value dalam sebuah cookies, jadi kalau sebelumnya kita membuat 2 cookies dengan nama cookies Nama dan lastVisit dan sekarang kita akan membuat sebuah Cookies dimana di dalamnya mengandung Nama dan lastVisit . atau biasa kita sebut dengan single cookies yang memiliki subkeys Nama dan lastVisit.

Kita menggunakan cookies dengan subkeys karena beberapa alasan :

1.Kita ingin mengelompokkan informasi yang sama di dalam sebuah cookies termasuk tanggal expired , dan informasi lainnya.

2.Subkeys dapat membantu kita dalam mengatasi "Cookie Limitations" yang sudah di jelaskan pada bahasan sebelumnya , cookies biasanya dibatasi hingga 406 bytes dan maximal 20 cookies di simpan dari sebuah situs , dengan menggunakan single cookie with subkeys , kita dapat membuat 1 keys dengan 20 subkeys , ini dapat membantu sekali dalam menyelesaikan masalah "limitations cookies" ,

Berikut adalah contoh untuk membuat Cookies dengan Subkey dengan C# Behind Code:

```
Response.Cookies["userInfo"]["Nama"] = "SuryoDesign";
Response.Cookies["userInfo"]["lastVisit"] = DateTime.Now.ToString();
Response.Cookies["userInfo"].Expires = DateTime.Now.AddDays(1);

HttpCookie sDesignCookie = new HttpCookie("userInfo");
sDesignCookie.Values["userName"] = "patrick";
sDesignCookie.Values["lastVisit"] = DateTime.Now.ToString();
sDesignCookie.Expires = DateTime.Now.AddDays(1);
Response.Cookies.Add(sDesignCookie);
```

Controlling Cookie Scope

Defaultnya atau pada standarnya , semua cookies pada website disimpan pada client dan semua cookies di terima tanpa adanya request dari site itu sendiri atau dengan kata lain semua cookies otomatis dikirimkan ketika anda membuka halaman website itu sendiri, kita dapat membatasi scope cookies tersebut dengan dua cara :

- 1.Membatasi Scope Cookies ke dalam folder pada server , yang mana mengizinkan kita untuk membatasi cookies pada sebuah website
- 2.Mengatur Scope dari domain yang mengizinkan kita untuk mengatur subdomain dari domain mana saja yang boleh mengakses cookie yang ada

Contoh dari Scope Cookies dari nomor 1 yang menggunakan folder / path :

```
HttpCookie appCookie = new HttpCookie("AppCookie");  
appCookie.Value = "written " + DateTime.Now.ToString();  
appCookie.Expires = DateTime.Now.AddDays(1);  
appCookie.Path = "/Application1";  
Response.Cookies.Add(appCookie);
```

Halaman tersebut merupakan halaman yang berada di bawah site root atau virtual root. Penempatan halaman Path ini akan berpengaruh terhadap Cookie yang dapat dilihat oleh halaman dari Application1 folder atau virtual root. Sebagai contoh : bila kita memiliki situs dengan nama www.suryodesign.asia maka cookie yang dibuat sebelumnya akan dapat di lihat dengan path : <http://www.suryodesign.asia/Application1> dan cookies tidak akan dapat dilihat dari halaman lainnya seperti dari halaman <http://www.suryodesign.asia/Application2> dan <http://www.suryodesign.asia/Application3> .

Perlu diperhatikan dalam browser biasanya Path / alamat direktori bersifat case sensitive. Kita tidak dapat mengendalikan bagaimana user mengetik URL pada browser tetapi bila aplikasi bergantung kepada cookies dengan Path spesifik , maka kita mesti memastikan bahwa semua URL dan hyperlink mesti match dengan yang telah kita buat.

Contoh dari Scope Cookies dari nomor 2 yang menggunakan subdomain

Secara Default atau standard , cookies di gunakan dengan menggunakan spesifik domain dari user tersebut , sebagai contoh bila website kita www.suryodesign.asia maka cookie yang akan dibuat akan menggunakan nama website tersebut. Tapi bagaimana bila website kita memiliki subdomain seperti <http://gallery.suryodesign.asia> maka kita dapat mengautr cookies tersebut dengan specific subdomain :

```
Response.Cookies["domain"].Value = DateTime.Now.ToString();  
Response.Cookies["domain"].Expires = DateTime.Now.AddDays(1);  
Response.Cookies["domain"].Domain = "gallery.suryodesign.com";
```

Ketika domain telah di atur seperti di atas , maka cookie akan tersedia untuk subdomain yang di spesifikasikan dan kita juga dapat menggunakan domain property untuk sharing di antara multiple subdomains maka anda dapat membuat domain dengan mengatur nama domain sebanyak anda mau membuat multiple subdomains yang ada.

Membaca Single Cookies

```
Response.Cookies["Nama"].Value = "SuryoDesign";
Response.Cookies["userName"].Expires = DateTime.Now.AddDays(5);

HttpCookie sDesignCookie = new HttpCookie("lastVisit");
sDesignCookie.Value = DateTime.Now.ToString();
sDesignCookie.Expires = DateTime.Now.AddDays(1);
Response.Cookies.Add(sDesignCookie);

if (Request.Cookies["Nama"] != null)
Label1.Text = Server.HtmlEncode(Request.Cookies["Nama"].Value);

if (Request.Cookies["Nama"] != null)
{
HttpCookie aCookie = Request.Cookies["Nama"];
Label2.Text = Server.HtmlEncode(aCookie.Value);
}
```

Membaca Subkeys pada Cookies :

```
Response.Cookies["userInfo"]["userName"] = "suryoDesign";
Response.Cookies["userInfo"]["lastVisit"] = DateTime.Now.ToString();
Response.Cookies["userInfo"].Expires = DateTime.Now.AddDays(1);

HttpCookie aCookie = new HttpCookie("userInfo");
aCookie.Values["userName"] = "suryoDesign";
aCookie.Values["lastVisit"] = DateTime.Now.ToString();
aCookie.Expires = DateTime.Now.AddDays(1);
Response.Cookies.Add(aCookie);

if (Request.Cookies["userInfo"] != null)
{
    Label1.Text =
        Server.HtmlEncode(Request.Cookies["userInfo"]["userName"]);

    Label2.Text =
        Server.HtmlEncode(Request.Cookies["userInfo"]["lastVisit"]);
}
```

Dalam contoh di atas kita membaca value dari subkey lastVisit dimana lastVisit di tunjukkan dengan tipe data DateTime value , dan cookies di simpan dengan string , jadi bila kita ingin

menggunakan date time value sebagai tipe data date maka kita mesti konversi type data tersebut menjadi dateTime seperti berikut :

```
dt = DateTime.Parse(Request.Cookies["userInfo"]["lastVisit"]);
Label2.Text = dt.ToString();
```

Kita juga memiliki cara lain untuk mengambil subkey collection dan kemudian memindahkannya subkey value dengan nama subkey tersebut dengan menggunakan type dari NameValueCollection , contoh :

```
HttpCookie aCookie = new HttpCookie("userInfo");
aCookie.Values["userName"] = "suryoDesign";
aCookie.Values["lastVisit"] = DateTime.Now.ToString();
aCookie.Expires = DateTime.Now.AddDays(1);
Response.Cookies.Add(aCookie);

if (Request.Cookies["userInfo"] != null)
{
    System.Collections.Specialized.NameValueCollection UserInfoCookieCollection;
    UserInfoCookieCollection = Request.Cookies["userInfo"].Values;
    Label1.Text = Server.HtmlEncode(UserInfoCookieCollection["userName"]);
    Label2.Text = Server.HtmlEncode(UserInfoCookieCollection["lastVisit"]);
}
```

Dan kita juga dapat membaca semua object cookies yang ada dengan cara berikut :

```
HttpCookie aCookie = new HttpCookie("userInfo");
aCookie.Values["userName"] = "suryoDesign";
aCookie.Values["lastVisit"] = DateTime.Now.ToString();
aCookie.Expires = DateTime.Now.AddDays(1);
Response.Cookies.Add(aCookie);

System.Text.StringBuilder output = new System.Text.StringBuilder();
HttpCookie aCookie2;

for (int i = 0; i < Request.Cookies.Count; i++)
{
    aCookie2 = Request.Cookies[i];
    output.Append("Cookie name = " + Server.HtmlEncode(aCookie2.Name)
        + "<br />");
    output.Append("Cookie value = " + Server.HtmlEncode(aCookie2.Value)
        + "<br /><br />");
}
Label1.Text = output.ToString();
```


Bila anda ingin mengubah Cookies seperti mengubah Expired nya saja dapat diubah dengan metode seperti berikut ini :

```
Response.Cookies["counter"].Expires = DateTime.Now.AddDays(1);
```

Dan bila anda ingin menghapusnya anda tinggal mengurangi waktu expirednya seperti berikut :

```
Response.Cookies["counter"].Expires = DateTime.Now.AddDays(-1);
```

Sekarang saya akan coba meng-implementasikan nya secara sederhana ke dalam sebuah aplikasi :

Berikut halaman dari Default.aspx ketika cookies belum ditemukan / blm dibuat :



Berikut halaman dari Default.aspx ketika Cookies telah dibuat / ditemukan :



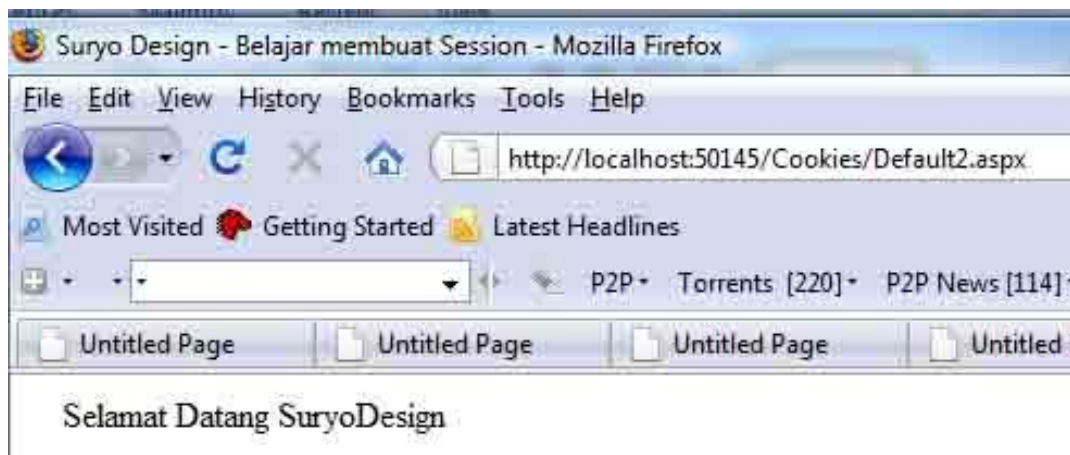
Source dari halaman 1.aspx :

```
protected void Page_Load(object sender, EventArgs e)
{
    Label1.Visible = true ;
    Label2.Visible = false;
    ButtonLogin.Visible = false;
    if (Request.Cookies["Nama"].Value != "")
    {
        Label1.Visible = false;
        Label2.Visible = true;
        ButtonLogin.Visible = true;
    }
}

protected void btnCreateCookies_Click(object sender, EventArgs e)
{
    if (txtNama.Text == "")
    {
        MessageBox.Show("Nama Tidak Boleh Kosong", "Pesan Kesalahan",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        Response.Cookies["Nama"].Value = txtNama.Text ;
        Response.Redirect("Default.aspx");
    }
}

protected void ButtonLogin_Click(object sender, EventArgs e)
{
    Response.Redirect("Default2.aspx");
}
```

Berikut adalah halaman Default 2.aspx ketika berhasil Login :



Source dari halaman 2.aspx

```
protected void Page_Load(object sender, EventArgs e)
{
    if (Request.Cookies["Nama"].Value != "")
    {
        Label1.Text = "Selamat Datang " +
        Server.HtmlEncode(Request.Cookies["Nama"].Value);
        Response.Cookies["Nama"].Value = "";
    }
    else if (Request.Cookies["Nama"].Value == "")
    {
        Response.Redirect("Default.aspx");
    }
}
```

Oke Sekarang anda telah mengerti mengenai Cookies, sekarang kita akan membahas mengenai Session , Sekarang saya akan menerangkan secara global mengenai perbedaan Cookies , session dan ViewState.

Cookies : selalu aktif di client selama File cookies masi ada dan waktu expired blm habis, bisa digunakan di semua halaman.

Session : Aktif selama Browser tidak di close dan timeout blm habis, dan bisa digunakan pada semua halaman

ViewState : Hanya aktif di halaman tertentu saja dimana ViewState di definisikan, dan akan hilang bila browser di close

Session

Oke sekarang kita akan membahas mengenai Session, apa itu Session ?

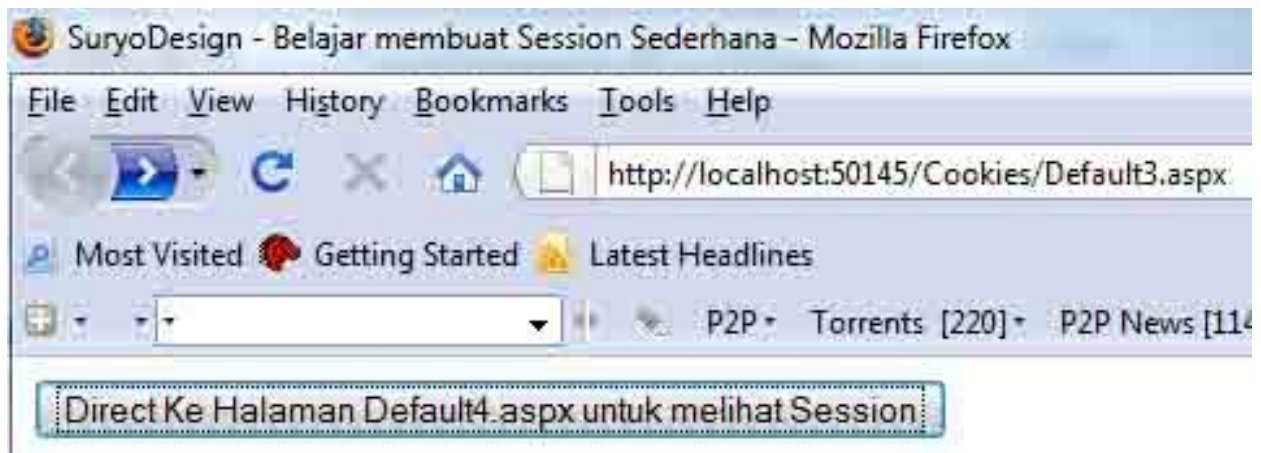
Session adalah data yang di simpan di dalam memory client sering digunakan untuk kebutuhan web tersebut dan akan hilang bila browser tersebut di close.

Session akan hilang bila kondisi berikut terjadi :

- 1.User menutup atau restart browser.
- 2.User yang mengakses halaman yang sama dengan browser yang berbeda.
- 3.Session timeout.
- 4.Programmer mengakhiri Code Session yang ada.

Pada point 1 dan 2 , ini dapat mengakhiri session karena session sebenarnya di simpan di dalam memory dan ketika browser di tutup maka memory itu akan dihapus juga atau dengan browser yang berbeda akan memiliki alamat memory yang berbeda pada browser sehingga session tidak dapat digunakan pada browser tersebut.

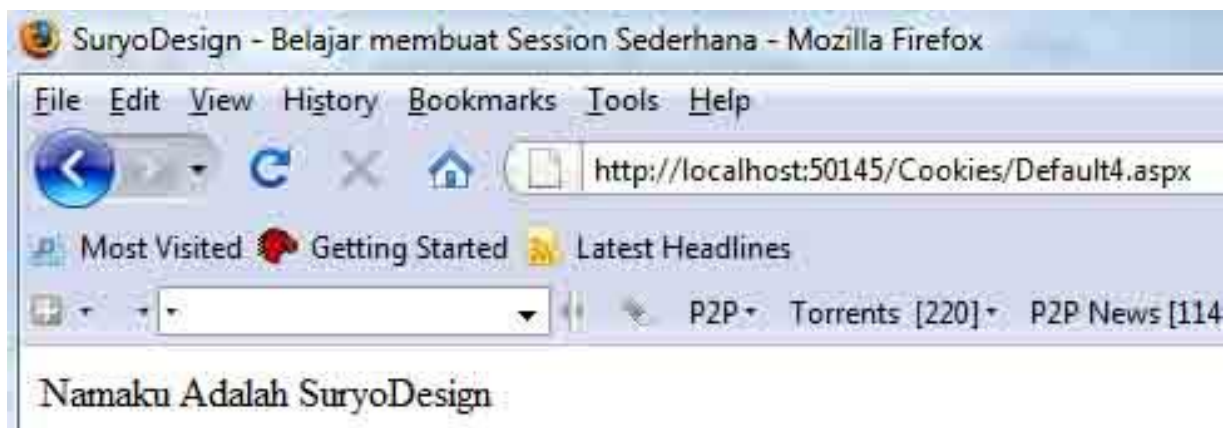
Berikut adalah Contoh Halaman Default3.aspx dimana session dibuat :



Source Code dengan C# dari halaman Default3.aspx :

```
protected void Page_Load(object sender, EventArgs e)
{
    Session["satu"] = "Namaku ";
    Session["dua"] = "Adalah ";
    Session["tiga"] = "SuryoDesign";
}
```

Dan ketika button dari halaman 3 di tekan maka akan di direct ke halaman Default4.aspx , berikut adalah tampilan dari halaman Default4.aspx :



Berikut Behind Code dari halaman Default4.aspx :

```
protected void Page_Load(object sender, EventArgs e)
{
    Label1.Text = Session["satu"].ToString() + Session["dua"].ToString() +
    Session["tiga"].ToString();
}
```

Penutup

Tidak banyak kata saya sampaikan pada penutup kali ini . 3 Tips Dan Trik aja dari saya :

1. Terus Mencoba dan berpikir lah jernih agar lebih tenang dalam mengerjakan sesuatu !!!
2. Gunakan MSDN untuk membantu anda.
3. Googling untuk mencari referensi yang dapat membantu anda, karena pasti bukan hanya anda yang pernah mengalami masalah ini , di jamin pasti ketemu dengan googling ^^

REFERENSI :

- MSDN 2008
- <http://msdn.microsoft.com>
- <http://wikipedia.com/>
- <http://www.microsoft.com>
- www.w3schools.com/
- <http://asp.net-tutorials.com/>
- Ebook : press.Beginning.ASP.NET.2.0.in.C.Sharp.2005

Biografi Penulis



M.Suryo Pranoto – Alumni Mahasiswa Perguruan Tinggi Universitas Bina Nusantara , Aktif dalam beberapa komunitas komputer , dan beberapa project terutama berbasis aplikasi seperti VB.Net atau C#, dan sedang berusaha keras untuk menabung dan berencana untuk melanjutkan Cisco CCNP setelah menyelesaikan CCNA , memiliki hobby untuk sharing mengenai komputer mulai dari software hingga jual beli hardware maupun modding komputer.

Ym : suryolovetyka ,

Email : suryodesign@yahoo.co.id

Website : www.suryodesign.asia

Blog : www.suryodesign.wordpress.com

Ilmu Komputer : <http://ilmukomputer.org/author/suryodesign/>