

Part 8 – Transferring Information

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

Perbedaan paling mendasar antara programming untuk internet dan untuk desktop adalah state management dengan kata lain perbedaannya adalah bagaimana kita mendapatkan informasi untuk user yang menggunakan dan bagaimana kita menyimpan data tersebut, dalam tradisional windows application , state diatur otomatis dan memory mencukup dan selalu tersedia , tapi dalam web application , ini berbeda cerita karena ratusan atau bahkan ribuan user terhubung secara bergantian menggunakan aplikasi yang sama dalam sebuah web server , satu dan lainnya sama2 berkomunikasi melalui HTTP dalam internet , kondisi ini yang tidak memungkinkan bila web application dibuat seperti desktop application.

Untuk membuat web application yang efisien kita akan belajar untuk menggunakan State, dan kita akan melihat dalam ASP.NET yang memiliki management features untuk menyimpan dan mengatur informasi secara hati-hati dan konsisten , kita akan explore atau mempelajari lebih jauh lagi tentang perbedaan state seperti View State , Session State , Custom Cookies . kita gunakan tergantung keadaan bagaimana kita ini mentransfer informasi dari halaman yang satu ke halaman yang lain menggunakan cross-page postring dan query string , bagi yang ingin mendownload artikel dapat mendownload di www.suryodesign.asia/freedownload dan bila yang ingin melihat artikel saya yang lainnya dapat membaca di sini <http://ilmukomputer.org/author/suryodesign/>

ISI

Dalam tradisional windows program , user berinteraksi dengan aplikasi yang di jalankan , bagian dari memory pada desktop computer di alokasikan untuk menyimpan sekumpulan informasi yang dibutuhkan . hal ini tidak akan berat bila pada desktop computer yang hanya terhubung untuk 1 pengguna .

Tapi bayangkan pada internet ??? bila ada 3000 user terhubung dan menyimpan sekumpulan data informasi user pada server ??? wah dijamin habis ini memory server ^^ , karena itu kita gunakan State Management untuk menyimpan informasi pada sisi Client , dan saya ingin menegerangkan lebih jauh lagi bahwa Web application menggunakan pola disconnect access , dimana ketika kita me-request sebuah halaman web dari client connect kepada web server dan melakukan request halaman , dan ketika halaman sudah ditampilkan , connection akan di putuskan , dan web server akan mengacuhkan semua informasi tentang client , ketika user menerima halaman, aplikasi telah berhenti bekerja karena semua data telah berhasil di tampilkan , untuk mengingat user yang telah login tadi biasanya server menggunakan beberapa step untuk mencata informasi / data tentang user yang login.

View State

Pada bab sebelumnya kita telah belajar untuk menggunakan view state untuk menyimpan informasi yang ada , informasi view state disimpan didalam hidden field dan otomatis di kembalikan ke server setiap ada postback , bagaimanapun viewstate tidak dibatasi oleh server control , kita dapat menambahkan informasi langsung ke view state dan mengambil lagi informasi itu ketika dibutuhkan, tipe informasi ini dapat disimpan dengan beberapa macam simple data types dan custom objects.

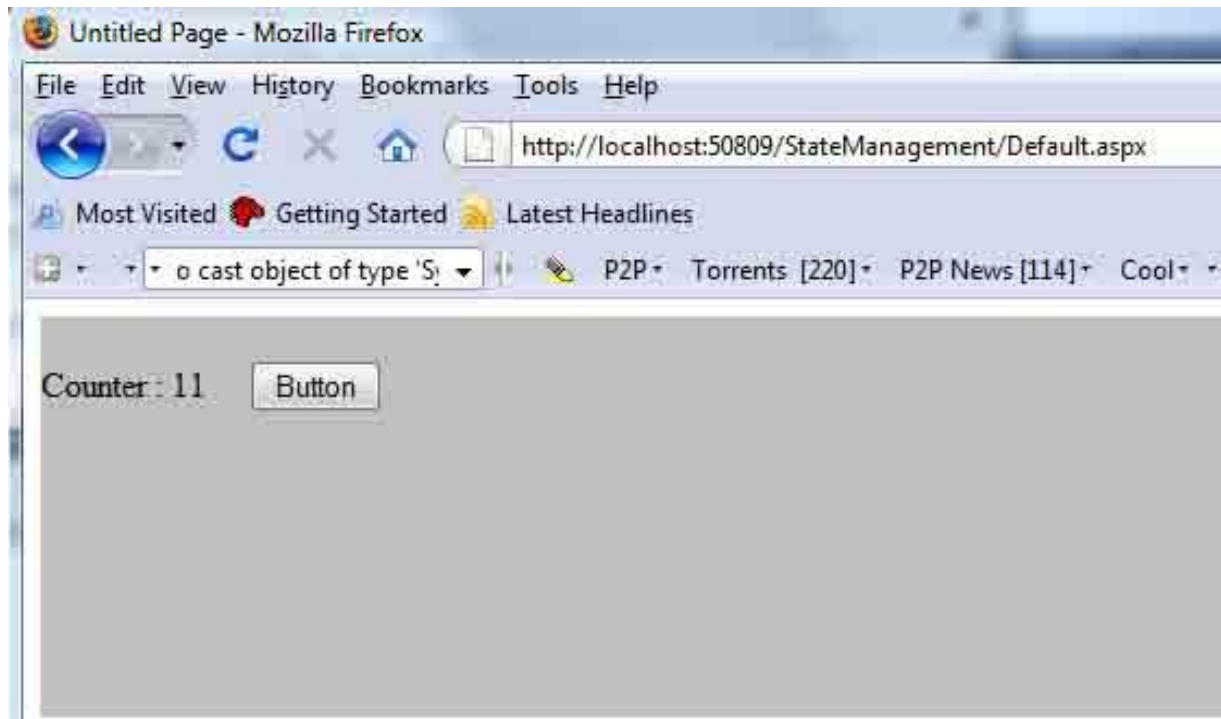
Berikut adalah contoh untuk membuat ViewState :

```
this.ViewState["CounterNilai"] = 1;
string nilai;
nilai = ViewState["CounterNilai"].ToString();
MessageBox.Show(nilai);
```

Disini kita akan membahasa ViewState dengan nama CounterNilai dan diisi dengan nilai 1 dengan tipe data integer , dan bila dalam 1 halaman ada lebih dari 1 nilai yang mendeskripsikan tentang ViewState CounterNilai maka Deskripsi yang terakhir akan menggantikan / menimpah deskripsi yang lama.

Ketika kita mengambil data dari ViewState tersebut, kita akan mengambil berdasarkan nama ViewState tersebut dan kita juga menggunakan Cast untuk mengambil value dari data tersebut , kenapa kita butuh cast ? karena ViewState Collection menyimpan semua data sebagai Generic Object yang mengizinkan kita menyimpan data dalam tipe data yang berbeda.

Sekarang kita akan mencoba membuat suatu counter Sederhana pada halaman Default.aspx, berikut contoh printscreennya :



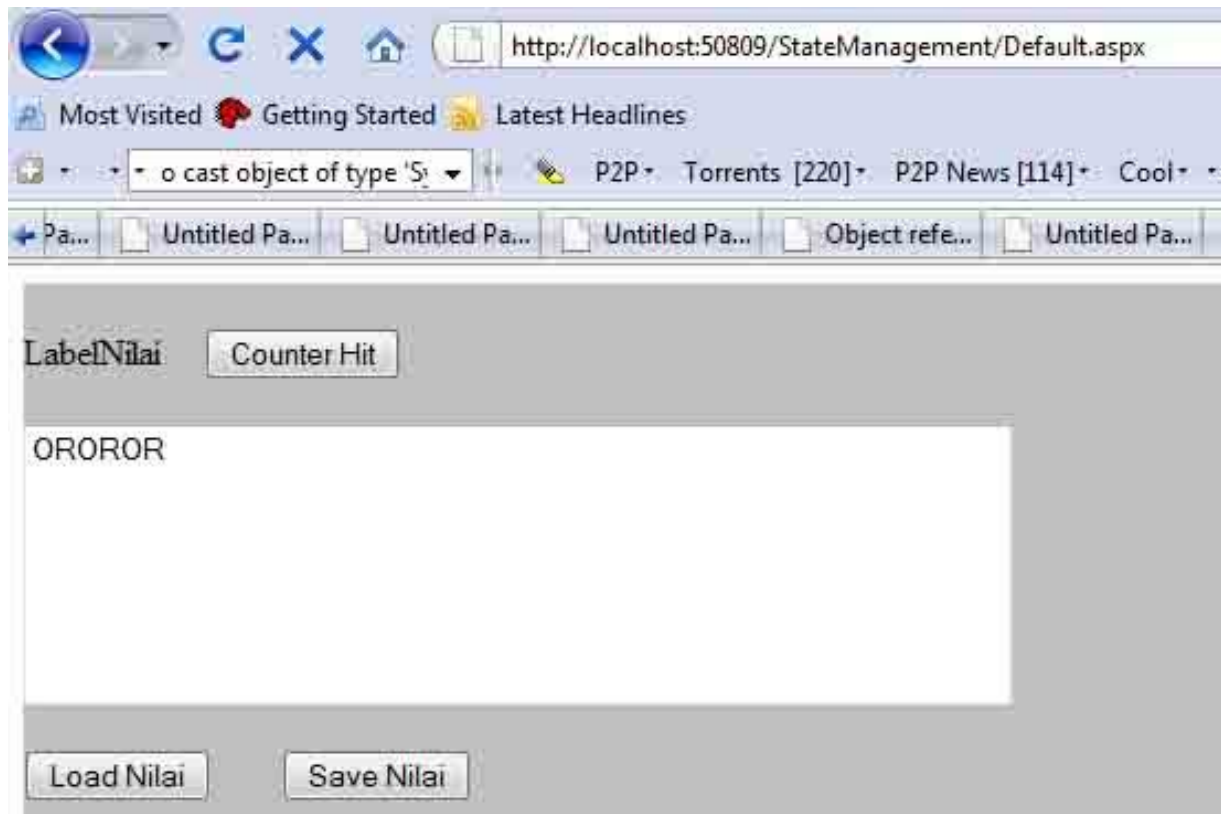
Dan berikut adalah behind code dengan C# dari Default.aspx.cs :

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        int counterValue;
        if (ViewState["CounterNilai"] == null)
        {
            counterValue = 1;
        }
        else
        {
            counterValue = (int)ViewState["CounterNilai"] + 1;
        }
        ViewState["CounterNilai"] = counterValue;
        Label1.Text = "Counter : " + counterValue.ToString();
    }
}
```

Dalam kode di atas kita ingin memastikan apakah pernah ada ViewState sebelumnya pada halaman itu , bila belum ada ViewState maka kita set variable counterValue dengan nilai 1 , tapi bila ada maka nilai sekarang akan di tambahkan dengan + 1 , kita dapat dengan mudah mengecek ViewState sudah ada atau belum menggunakan Null Reference exception .

Kita dapat juga menggunakan metode yang lain untuk menyimpan variable , pada contoh di atas kita menyimpan nilai di dalam variable dan ViewState , tapi bisa juga kita menyimpannya di dalam label , kemudian nanti kita membaca nilai dari label berikut apakah memiliki nilai atau tidak , ini hanyalah sebuah logika / tehnik yang dapat kita buat berdasarkan pola pikir kita.

Sekarang kita akan mencoba menambah kan sedikit informasi untuk meload data dari ViewState dan menyimpan dari ViewState seperti berikut :



Berikut adalah Source Codenya :

```
public partial class _Default : System.Web.UI.Page
{
    private string simpanNilai = "";
    protected void Page_PreRender(object sender, EventArgs e)
    {
        ViewState["simpanNilai"] = simpanNilai;
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        if (this.IsPostBack)
        {
            simpanNilai = (string)ViewState["simpanNilai"];
        }
    }
    protected void btnLoad_Click(object sender, EventArgs e)
    {
        txtNilai.Text = simpanNilai;
    }

    protected void btnSave_Click(object sender, EventArgs e)
    {
        simpanNilai = txtNilai.Text;
        txtNilai.Text = "";
    }

    protected void btnCounter_Click(object sender, EventArgs e)
    {
        int counterValue;
        if (ViewState["CounterNilai"] == null)
        {
            counterValue = 1;
        }
        else
        {
            counterValue = (int)ViewState["CounterNilai"] + 1;
        }
        ViewState["CounterNilai"] = counterValue;
        Label1.Text = "Counter : " + counterValue.ToString();
    }
}
```

Berikut Saya akan membuat perbandingan terhadap masing-masing event ketika dijalankan , karena disini prosesnya terlihat ribet untuk bagi yang baru belajar.

Halaman Di Load Pertama Kali	<pre>private string simpanNilai = ""; protected void Page_PreRender(object sender, EventArgs e) { ViewState["simpanNilai"] = simpanNilai; }</pre>
Button Save Di Klik	<pre>private string simpanNilai = ""; protected void Page_Load(object sender, EventArgs e) { if (this.IsPostBack) { simpanNilai = (string)ViewState["simpanNilai"]; } } protected void btnSave_Click(object sender, EventArgs e) { simpanNilai = txtNilai.Text; txtNilai.Text = ""; } protected void Page_PreRender(object sender, EventArgs e) { ViewState["simpanNilai"] = simpanNilai; }</pre>
Button Load Di Klik	<pre>private string simpanNilai = ""; protected void Page_Load(object sender, EventArgs e) { if (this.IsPostBack) { simpanNilai = (string)ViewState["simpanNilai"]; } } protected void btnLoad_Click(object sender, EventArgs e) { txtNilai.Text = simpanNilai; } protected void Page_PreRender(object sender, EventArgs e) { ViewState["simpanNilai"] = simpanNilai; }</pre>

Dapat anda lihat pada table di atas urutan ketika aplikasi di jalankan , perhatikan pada saat pertama kali form di Load . anda pasti berpikir kenapa fungsi Page_Load tidak dijalankan ??? bukannya fungsi ini pasti di jalankan pada setiap program di eksekusi ? jawabannya perintah ini dijalankan juga pada formload tetapi di dalamna terdapat `if (this.IsPostBack)` tetapi karena halaman ini pertama kali di load dan bukan merupakan PostBack atau bukan halaman yang pernah di buka kembali maka perintah di dalam IF ini tidak di jalankan sehingga form Load dianggap tidak berisi apa-apa dan tidak ada yang di eksekusi di dalamnya .

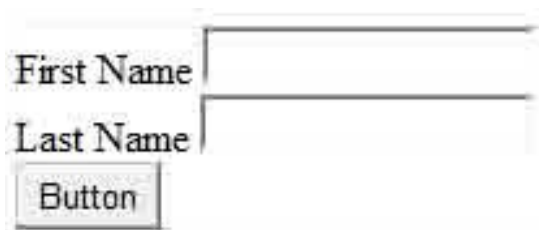
Perhatikan sekali lagi di situ ada `protected void Page_PreRender` , kita akan berpikir apakah guna dari event ini ? anda perhatikan dengan seksama pada setiap proses , event ini selalu di jalankan pada akhir dari semua proses yup2 ini memang dijalankan pada setiap akhir proses , seperti pada saat button saved di click , maka event ini dijalankan setelah isi dari button Code itu sendiri di eksekusi.

Transferring Information

Salah satu batasan dari View State adalah hanya bergantung pada halaman tertentu di mana ViewState di definisikan dan tidak berlaku pada halaman lain, berbeda dengan session yang aktif atau dapat digunakan pada setiap halaman, bila kondisinya user melakukan navigasi ke halaman lain , maka data informasi yang disimpan itu akan hilang , dalam kasus ini kita akan melakukan 2 tehnik basic untuk mentransfer informasi antara seperti : cross-page postring dan Query String.

Cross-Page Posting

Salah satu pendekatan baru dalam ASP.NET 2.0 adalah untuk men-trigger postback pada halaman lainnya , infrastructure yang mendukung cross-page postback ini dinamakan PostBack URL yang dapat didefinisikan melalui ImageButton , Link Button dan Button. Untuk Menggunakan Cross-Postring , kita dapat men-set PostBackUrl menjadi nama web lain . ketika user meng-klik salah satu button dan kemudian button tersebut akan menuju halaman lain dan dalam string URL itu akan berisi value baru. Untuk Lebih jelasnya akan saya contohkan seperti berikut contoh tampilan dari Default2.aspx :



Dan berikut adalah design Form dari Default2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default2.aspx.cs"
Inherits="Default2" %>
<!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>Default2.aspx</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            First Name
            <asp:TextBox ID="txtFirst" runat="server"></asp:TextBox>
            <br />
            Last Name
            <asp:TextBox ID="txtLast" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="btnSubmit" runat="server"
PostBackUrl="~/Default3.aspx"
```

```
        Text="Button" />
    <br />
    <br />
    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>
</html>
```

Dapat Kita lihat pada Page di atas tidak berisi code untuk mengolah / eksekusi data hanya Berisi tampilan data untuk mengisi firstname dan lastname saja, dan kita akan membuat sebuah behind code dengan C# untuk menyimpan data dari textbox di atas. Berikut Default2.aspx.cs

```
public partial class Default2 : System.Web.UI.Page
{
    public string FullName
    {
        get { return txtFirst.Text + " " + txtLast.Text; }
    }
}
```

Dapat kita lihat pada syntax di atas terpadaPostBack dimana halaman akan di arahkan :

```
<asp:Button ID="btnSubmit" runat="server" PostBackUrl="~/Default3.aspx"
```

Ketika button di submit maka halaman akan di-redirect ke halaman Default3.aspx, dan pada halaman Default3.aspx kita tambahkan syntax berikut :

```
<%@ PreviousPageType VirtualPath="~/Default2.aspx" %>
```

Syntax ini memiliki Virtual Path atau bisa dikatakan hanya halaman yang berasal dari halaman Default2.aspx yang dapat mengakses halaman yg memiliki Virtual Path ini.

Berikut adalah isi halaman BehindCode dengan C# dari Default3.aspx untuk menampilkan Keterangan FullName , dan beberapa data User seperti Title halaman pada halaman sebelumnya , judul title halaman skrg , IP USER , Browser User , Hostname , berikut adalah printscreen dari halaman Default3.aspx :

```
Selamat Datang : suryo design
Anda datang dari halaman : Default2.aspx
Dan Judul Page Title Sekarang adalah Menampilkan Informasi User
Dan IP Anda adalah : 127.0.0.1
Dan Browser anda adalah : Mozilla/5.0 (Windows; U; Windows NT 6.0; en-US; rv:1.9) Gecko/2008052906 Firefox/3.0
Dan HostName anda adalah : 127.0.0.1
```

Cross-Page Post

Berikut adalah Behind Code dari halaman Default3.aspx

```
protected void Page_Load(object sender, EventArgs e)
{
    if (PreviousPage != null)
```



```
{
LabelPrevPage.Text = "Selamat Datang : " + PreviousPage.FullName + "<br>" +
"Anda datang dari halaman : " + PreviousPage.Header.Title + "<br>" +
"Dan Judul Page Title Sekarang adalah " + Page.Header.Title + "<br>" +
"Dan IP Anda adalah : " + Page.Request.UserHostAddress + "<br>" +
"Dan Browser anda adalah : " + Page.Request.UserAgent + "<br>" +
"Dan HostName anda adalah : " + Page.Request.UserHostName + " <br>";
if (PreviousPage.IsCrossPagePostBack == true)
{
LabelCrossPage.Text = "Cross-Page Post";
}
else
{
LabelCrossPage.Text = "Bukan Cross-Page Post";
}
}
```

Query String

Query String merupakan pendekatan yang paling sering dipakai untuk mentransfer informasi melalui URL , seperti pada contohnya bila anda perhatikan pada google ketika anda melakukan searching anda dengan keyword suryo design , tanpa sadar anda akan mendapatkan URL seperti berikut :

```
http://www.google.com/search?q=suryo+design
```

Query string merupakan potongan dari URL yang ditandai dengan penanda untuk memisahkan variable yang ada di dalamnya , dalam kasus ini , kita mendefinisikan dengan nama variable q , yang mengandung string suryo+design , Beberapa batasan dari querystring ini dapat anda lihat seperti berikut :

- 1.Information di batasi hanya untuk simple string dan tidak boleh mengandung URL dan legal character
- 2.Information dapat di lihat oleh jelas oleh user mulai dari nama variable dan isi variable tersebut
- 3.User yang berpengalaman dapat langsung mengubah query string dan memasukkan value yang baru , ini merupakan metode yang tidak kita harapkan dan kita ingin melindungi / mem-protect web dari kejadian seperti ini
- 4.Beberapa browser membatasi panjang dari URL ini (biasanya antara 1 KB hingga 2 KB). Karena alasan tersebut kita tidak dapat menempatkan informasi dalam jumlah yang besar di dalam query string dan mesti memastikan kompatibilitas dengan browser yang ada.

Berikut adalah beberapa contoh untuk menggunakan query string

```
Response.Redirect("default.aspx?ID=1");
```

Dan kita juga dapat mengirim beberapa parameter melalui QueryString dengan memisahkan parameter tersebut dengan ampersand(&) :

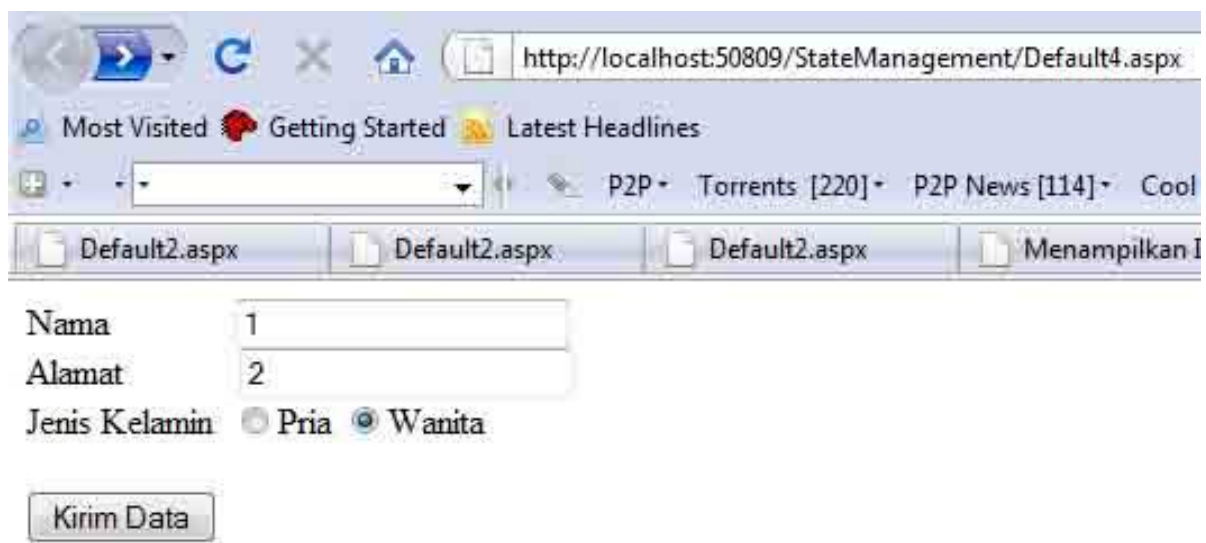
```
Response.Redirect("default.aspx?ID=1&mode=full");
```

Dan untuk mengambil nilai dari query string berikut ketika query string di terima oleh halaman tertentu kita dapat menggunakan perintah Request.QueryString["ID"] seperti berikut :

```
String ID = Request.QueryString["recorded"];
```

Perlu kita perhatikan dalam penerimaan data , semua informasi yang diterima selalu dalam tipe string yang mana perlu kita konversi ke dalam tipe data lainnya. Value dalam query string selalu ditandai dengan nama variable, bila kita bandingkan dengan view state , informasi dalam query string terlihat jelas (clearly visible) dan tidak di enkripsi , jadi saran saya jangan gunakan query string untuk informasi yang perlu di hidden.

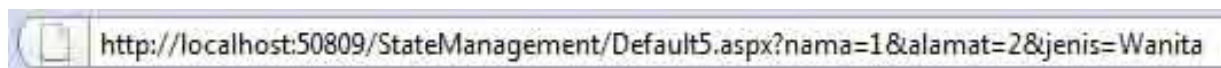
Berikut adalah Contoh Dari halaman display pada Default4.aspx untuk mengirim data ke halaman Default5.aspx



The screenshot shows a web browser window with the address bar displaying `http://localhost:50809/StateManagement/Default4.aspx`. Below the browser window, a form is visible with the following elements:

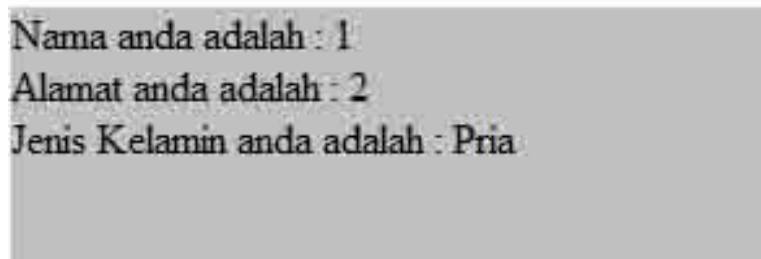
- Nama**: A text input field containing the value `1`.
- Alamat**: A text input field containing the value `2`.
- Jenis Kelamin**: Two radio buttons labeled `Pria` and `Wanita`. The `Wanita` radio button is selected.
- Kirim Data**: A button to submit the form data.

Dapat kita lihat pada halaman Default5.aspx terdapat variable nama , alamat , dan jenis beserta value dari masing-masing variable pada URL nya seperti berikut



The screenshot shows the address bar of a web browser with the following URL: `http://localhost:50809/StateManagement/Default5.aspx?nama=1&alamat=2&jenis=Wanita`

Dan Berikut adalah Contoh tampilan ketika string ditampilkan pada halaman Default5.aspx :



The screenshot shows a web page with the following text displayed:

- Nama anda adalah : 1**
- Alamat anda adalah : 2**
- Jenis Kelamin anda adalah : Pria**

Berikut adalah BehindCode untuk menangkap variable dari URL :

```
Label1.Text = "Nama anda adalah : " + Request.QueryString["nama"] + "<br>" +
```

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2009 IlmuKomputer.Com

```
"Alamat anda adalah : " + Request.QueryString["alamat"] + "<br>" +  
"Jenis Kelamin anda adalah : " + Request.QueryString["jenis"] + "<br>" ;
```

Bila anda mencoba untuk mengisi value dengan nama yang memiliki spasi seperti "Suryo Design" maka pada saat di kirim akan di encode menjadi "Suryo%Design", % disini akan menggantikan spasi dari Suryo Design. Selamat sampai disini semoga anda telah mengerti ^^.

Berikut halaman dan isinya :

Default.aspx = Counter + Load & Save Ke dalam ViewState

Default2.aspx = Merupakan halaman yg berisi Design Form

Default3.aspx = Menampilkan data dari halaman Default2.aspx dengan cross-page dan juga keterangan mengenai browser user , hostname , IP Address dan judul halaman pada Previous page / halaman sebelumnya.

Default4.aspx = Berisi Halaman Design Form Pengisian Data tentang Nama , Alamat , Jenis Kelamin

Default5.aspx = Berisi tentang Halaman menampilkan data QueryString dari halaman Default4.aspx

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/>