

Part 13 – DataBinding &Data Source Control

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

Banyak Metode dan variasi yang dapat kita gunakan untuk menampilkan data , dan banyak juga komponen yang dapat digunakan untuk menampilkan data , antara metode dan komponen yang digunakan memiliki cara dan perlakuan yang sama dan ada beberapa yang berbeda , tetapi balik lagi ke user yang paling kita utamakan dalam subjek pembuatan kita dimana user sebagai pemakai yang paling banyak menentukan factor kesuksesan yang kita buat , dalam pertemuan kali ini saya akan membahas bagaimana membaca data dari database dan kemudian menampilkannya dan juga membahas tentang melakukan insert data. bagi yang ingin mendownload artikel ini beserta source codenya 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

Pada pertemuan sebelumnya kita belajar bagaimana untuk mengambil informasi dari database, dan sekarang kita akan mempelajari ini lebih dalam , tehnik ini lebih flexible ketimbang factor sebelumnya karena tehnik ini memudahkan penggunaanya , seperti contohnya kita dapat menggunakan DataSet atau DataReader untuk mengambil informasi , mengatur format nya secara manual dan menambahkan HTML table ke dalam halaman web , secara konsep ini tidak terlalu susah , tapi bagaimana pun ini masih

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2009 IlmuKomputer.Com

membutuhkan banyak proses dan mesti mengulangi beberapa code untuk membaca code dari awal hingga akhir data , ASP.Net menambahkan feature untuk melewati beberapa proses ini dan kemudian memunculkan semua data yang ada ke dalam HTML dengan format yang terstruktur yang dapat kita sebut dengan data binding yang mempercepat proses dan memudahkan proses pengambilan data.

Konsep databinding paling utama adalah memberitahu control yang kita gunakan untuk memilih datasource mana yang akan kita gunakan sebagai data utama dan bagian mana yang ingin kita display dan value nilai apa yang ingin kita dapatkan bagi user tersebut .

DataBinding sendiri memiliki 2 Type yaitu :

1.Single Value / "Simple" , DataBinding :

halamankita dapat menggunakan single value data binding untuk menambahkan informasi dibagian mana saja di dalam halaman ASP.Net. bahkan dapat menempatkan sebagian informasi ke dalam control atau plain text dalam HTML tag , single value data binding sering disebut dengan TextDinamis untuk sekedar mengubah sebagian data yang ada.

2.Repeated-Value / "List" , binding :

Repeated-value data binding memperbolehkan kita untuk menampilkan semua data yang ada ke dalam table / komponen , tidak seperti Single-Value yang dapat dengan mudah dan leluasa digunakan , Repeated-Value data binding membutuhkan control seperti CheckBoxList / ListBox atau GridView.

Data Source Control

Seperti kita ketahui tentang Data Source Control yang telah kita jelaskan pada artikel sebelumnya , ada banyak jenis Data Source Control yang dapat kita pakai untuk dapat melakukan koneksi dengan database , seperti dalam pertemuan kali ini kita akan membahas mengenai SqlDataSource modification,

Kalau anda bertanya apa sih kegunaan dari DataSource , berikut adalah kegunaan dari DataSource :

- 1.Digunakan untuk berkoneksi dengan database.
- 2.Dapat mengambil data dari datasource dan menyimpannya dalam control yang berhubungan.
- 3.Dapat update data source ketika data pada control mengalami perubahan nilai / value.

Contoh SqlDataSource :

```
<asp:SqlDataSource ID="SqlDataSource" runat="server"></asp:SqlDataSource>
```

Ada banyak contoh DataSource yang dapat kita temukan pada komponen .Net seperti :

- 1.SqlDataSource
- 2.AccessDataSource
- 3.LinqDataSource
- 4.ObjectDataSource
- 5.XmlDataSource
- 6.SiteMapDataSource

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2009 IlmuKomputer.Com

Untuk mendukung proses Modifikasi DataSource ada beberapa type Parameter yang dapat kita gunakan seperti berikut :

Source	Control Tag	Description
Control Property	<asp:ControlParameter>	Untuk mendapatkan property dari control pada halaman yang ada
Query String Value	<asp:QueryStringParameter>	Untuk mengambil nilai yang berasal dari QueryString pada URL
Session State Value	<asp:SessionParameter>	Untuk mengambil nilai dari session
Cookie Value	<asp:CookieParameter>	Untuk mengambil profile dari user
A Form Variabel	<asp:FormParameter>	Untuk mengambil value yang di post dari input control

Untuk saat ini baru 2 Control Tag yang akan kita gunakan yaitu ControlParameter dan QueryStringParameter. Nanti untuk komponen lebih lanjut pada artikel selanjutnya ^^.

Mungkin kita langsung masuk saja ke latihan kali ini untuk menerapkan teori / konsep yang telah kita pelajari , berikut kita buat halaman pertama yaitu **ShowData.aspx** :

The screenshot shows a web application with a header 'BELAJAR BERSAMA SURYO' and a sub-header 'Menu Searching Data'. Below the header is a search form with a text input 'Nama Barang :', a 'Cari Data' button, and a table of results. The table has columns: KdBarang, NamaBarang, Stok, Harga, and KdDistributor. The results show 5 items: Gelas, Piring, Sendok, Mangkok, and Lap Tangan. A pagination bar at the bottom of the table shows '1 2 3'. Below the table is a label 'Label Jumlah Data : 5'. Red arrows point to various elements with labels: 'Menu Searching Data' points to the header, 'Menu yang ditampilkan dari Web.SiteMap' points to the left sidebar, 'GridView yang berasal dari DataSet kemudian Di Bind' points to the table, and 'Jumlah Data Yang ditampilkan Dari GridView' points to the 'Label Jumlah Data : 5'.

KdBarang	NamaBarang	Stok	Harga	KdDistributor
1	Gelas	200	5000	2
2	Piring	500	7500	3
3	Sendok	100	3000	3
4	Mangkok	175	10000	2
5	Lap Tangan	11	6000	1

Label Jumlah Data : 5

Halaman **ShowData.aspx** , disini ada beberapa point yang kita pelajari :

- 1.Menyimpan data dari database ke dalam dataset.
- 2.Binding data ke dalam GridView dengan data yang berasal dari DataSet.
- 3.Membuat fungsi private.
- 4.Melakukan Pencarian / Searching Data.
- 5.Menghitung Data yang ditampilkan dari datagrid.
- 6.Mengaktifkan Paging pada DataGrid dan mengatur jumlah data yang tampil.

Oke , sekarang langsung saja kita buat , untuk komponen yang digunakan , anda pasti sudah mengetahui komponen apa saja yang terlibat di dalamnya bila anda melihat sekilas apalagi bila anda mempelajari artikel-artikel yang sebelumnya yang sudah menggunakan komponen ini satu-persatu.

Pertama kita siapkan dulu objek dan variabel yang akan kita gunakan agar dapat di gunakan pada semua bagian di form ini, ini kita letakkan di dalam behind code , ingat letaknya bukan di dalam form load , tetapi di luar form load , jangan sampai salah :

```
public partial class Services : System.Web.UI.Page
{
    //mengambil connection string dari Web.Config
    private string SuryoCon =
    ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
    //membuat objek SqlDataAdapter
    SqlDataAdapter da ;
    //membuat objek DataSet
    DataSet ds;
    //Membuat objek SqlCommand
    SqlCommand com = new SqlCommand();
}
```

Berikut isi BehindCode dari ShowData .aspx ketika halaman di Load :

```
protected void Page_Load(object sender, EventArgs e)
{
    //PostBack = fitur untuk memilih data dan kemudian
    //akan di kembalikan / di direct ke halaman itu
    if (IsPostBack)
    {
        //memanggil fungsi untuk mencari data
        //dan melempar sebuah string yang akan di cari
        panggilData(txtCari.Text.Trim());
    }
    else
    {
        //memanggil fungsi untuk mencari data
        //dan melempar string kosong di dalamnya
        //hasilna akan menampilkan semua data
        panggilData("");
    }
    //menghitung data yang ditampilkan dari dataGrid
    LabelJumlahData.Text = "Label Jumlah Data : " +
    GridView1.Rows.Count.ToString();
}
```

Ketika halaman di Load , ada fungsi PanggilData() , berikut code nya :

```
private void panggilData(string TerimaData)
{
    //membuat koneksi
    SqlConnection con = new SqlConnection(SuryoCon);
    try
    {
        //mengisi perintah command
        com.CommandText = "select * from msbarang where NamaBarang like '%" +
        TerimaData + "%'";
        //mendefinisikan koneksi yang akan digunakan
        com.Connection = con;
        //melakukan eksekusi perintah dari command
        da = new SqlDataAdapter(com);
        //membuat dataset untuk menyimpan data
        ds = new DataSet();
        //menyimpan data yang di select ke dalam
        //dataset dengan nama Tables barang
        da.Fill(ds, "barang");
        //memilih datasource yang digunakan
        GridView1.DataSource = ds;
        //memilih DataMember dari Tables pada dataSet
        GridView1.DataMember = "barang";
        //melakukan binding untuk menampilkan data
        GridView1.DataBind();
        //menutup koneksi
        con.Close();
    }
    catch (Exception kesalahan)
    {
        //menampilkan pesan kesalahan yang ada
        MessageBox.Show(kesalahan.ToString());
    }
    finally
    {
        //menghapus koneksi yang ada
        con.Dispose();
    }
}
```

Kemudian kita bisa mencari data dengan mengisi data pada TextBox yang ada dan menekan button cari data , berikut isi dari button cari data :

```
protected void btnCari_Click(object sender, EventArgs e)
{
    //memanggil fungsi untuk mencari data
    //dan melempar sebuah string yang akan di cari
    panggilData(txtCari.Text.Trim());
}
```

Sekarang kita telah dapat menampilkan data pada GridView dan sudah dapat melakukan searching data , tapi dapat kita lihat data yang tampil akan tampil semua , dan paging tidak diaktifkan , berikut property yang mesti kita ubah , klik Paging dan cari Properties berikut :

```
AllowPaging = true;  
PageSize = 5;
```

Atau anda juga dapat melakukannya melalui Code seperti berikut :

```
//mengaktifkan paging  
GridView1.AllowPaging = true;  
//set paging menjadi 1 halaman memiliki 5 data  
GridView1.PageSize = 5;
```

Oke sekarang anda telah berhasil membuat halaman **Insert.aspx** , sekarang saatnya membuat halaman Insert :

BELAJAR BERSAMA SURYO DESIG

Home-> Insert
Insert Data
Home
Show Data
Insert
System.Data
Binding

Nama Barang :
Harga :
Stok :
KdDistributor : PT. A ▼

☒ Insert Dengan Parameter
checked = query with parameter
not checked = not with parameter

Insert Data Reset Data

Mesti Number

Di atas adalah tampilan dari halaman **Insert.aspx** , berikut adalah beberapa point yang akan anda dapat dari halaman ini :

1. Menentukan Tipe data dengan TryParse
2. Insert Data dengan ExecuteNonQuery
3. Insert Data dengan ExecuteNonQuery dengan parameter
4. Menghitung data yang sudah ada dengan ExecuteScalar

ada beberapa rule yang ada di dalam insert disini :

1. Textbox tidak boleh ada yang kosong
2. TextBox Stok dan harga mesti dalam format number
3. Nama barang tidak boleh sama dengan nama yang sudah ada pada database

Oke langsung saja , buat variable global di dalam form :

```
public partial class Monitor : System.Web.UI.Page  
{  
    private bool isBoolTrue = true;  
}
```

Kemudian buat fungsi untuk mengecek tipe data apakah number atau bukan :

```
private void CekString(string TerimaNilai, string TerimaPesan)
{
    //integer i disini sebagai penyimpan nilai dan pembanding
    int i;
    //buat tipe data boolean yang akan menyimpan hasil
    //cara kerja dari int.TryParse adalah :
    //mencoba konversi data yang di terima yang berformat string
    //dan di coba konversi ke dalam Integer ( out i )
    //kemudian bila dapat di konversi ke integer
    //berarti True dan hasil akan di simpan ke variabel result
    bool result = int.TryParse(TerimaNilai, out i);
    if (!result)
    {
        MessageBox.Show(TerimaPesan);
        isBoolTrue = false;
    }
    else
    {
        isBoolTrue = true;
    }
}
```

Kemudian buat fungsi untuk membersihkan textbox yang ada ketika setelah insert data kemudian akan clear TextBox :

```
private void clearTextbox()
{
    txtNamaBarang.Text = "";
    txtHarga.Text = "";
    txtStok.Text = "";
}
```

Kemudian ketika Form di load :

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!IsPostBack)
    {
        //memanggil fungsi distributor
        TampilDistributor();
        //Memilih data pertama pada DropDownList$
        DropDownList1.SelectedIndex = 0;
    }
}
```


Kemudian buat fungsi untuk menampilkan data distributor ke dalam dropDownList

```
private void TampilDistributor()
{
    //mengambil connection string dari Web.config
    string SuryoCon =
    ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
    //membuat koneksi
    SqlConnection con = new SqlConnection(SuryoCon);
    try
    {
        //membuka koneksi
        con.Open();
        //mengisi perintah ke dalam command
        SqlCommand com = new SqlCommand("select * from msdistributor", con);
        //membuat objek SqlDataReader
        SqlDataReader rd;
        //membaca data dari command
        rd = com.ExecuteReader();
        //membaca data hingga data habis
        while (rd.Read())
        {
            //membuat list item untuk menyimpan data
            ListItem DaftarItem = new ListItem();
            //menampilkan data distributor ke dalam dropdownlist
            //daftar Item yang ditampilkan
            DaftarItem.Text = rd["NamaDistributor"].ToString();
            //daftar item yang akan menjadi value
            DaftarItem.Value = rd["KdDistributor"].ToString();
            //add item ke dalam dropdownlist
            DropDownList1.Items.Add(DaftarItem);
        }
        //tutup datareader
        rd.Close();
        //hapus objek datareader
        rd.Dispose();
    }
    catch (Exception kesalahan)
    {
        MessageBox.Show(kesalahan.ToString());
    }
    finally
    {
        //tutup koneksi dan hapus koneksi
        con.Close();
        con.Dispose();
    }
}
```


Kemudian buat fungsi untuk melakukan cek data pada database seperti berikut :

```
private void CekDataPadaDatabase(string TerimaNilai, string TerimaPesan)
{
    string SuryoCon =
    ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
    SqlConnection con = new SqlConnection(SuryoCon);
    try
    {
        int i ;
        con.Open();
        SqlCommand com = new SqlCommand("select count(namabarang) from msbarang where
        namabarang = '" + TerimaNilai + "'",con);
        //execute scalar digunakan untuk menghitung aggregate function
        //yang hanya membalikkan 1 nilai seperti max , min , count , avg
        i = Convert.ToInt16(com.ExecuteScalar());
        //bila i > 0 berarti maksudnya disini data itu
        //sudah ada pada database
        //maka tampilkan pesan bahwa data sudah ada
        if ( i > 0 )
        {
            MessageBox.Show(TerimaPesan);
            isBoolTrue = false;
        }
        else
        {
            isBoolTrue = true;
        }
        com.Dispose();
    }
    catch (Exception kesalahan)
    {
        MessageBox.Show(kesalahan.ToString());
    }
    finally
    {
        con.Close();
        con.Dispose();
    }
}
```

Kemudian berikut adalah Isi perintah dari Insert Data :

```
protected void Button1_Click(object sender, EventArgs e)
{
    //tidak boleh ada data yang kosong
    if (txtNamaBarang.Text.Trim() == "" || txtStok.Text.Trim() == "" ||
        txtHarga.Text.Trim() == "")
    {
        MessageBox.Show("tidak boleh ada kolom kosong");
        return;
    }
    //cek harga dan stok mesti number
    CekString(txtHarga.Text.Trim(), "Harga Mesti Number");
    if (!isBoolTrue) return;
    CekString(txtStok.Text.Trim(), "Stok Mesti Number");
    if (!isBoolTrue) return;

    //cek namabarang tidak boleh ada pada database
    CekDataPadaDatabase(txtNamaBarang.Text.Trim(), "Nama Barang Sudah ada");
    if (!isBoolTrue) return;

    string SuryoCon =
    ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
    SqlConnection con = new SqlConnection(SuryoCon);
    try
    {
        con.Open();
        //bila checkbox di centang atau bernilai true
        //lakukan perintah dalam if untuk melakukan insert data dengan paramter
        if (CheckBox1.Checked)
        {
            //buat string queryInsert untuk melakukan insertData
            //parameter diawali dengan lambang @
            string queryInsert = "insert into msbarang(NamaBarang,Stok,Harga ,
            KdDistributor)" + "values(@p_nama ,@p_stok , @p_harga , @p_dis)";
            SqlCommand com = new SqlCommand(queryInsert, con);
            //isi parameter dengan value
            com.Parameters.AddWithValue("@p_nama" , txtNamaBarang.Text.Trim());
            com.Parameters.AddWithValue("@p_stok" , txtStok.Text.Trim() );
            com.Parameters.AddWithValue("@p_harga", txtHarga.Text.Trim() );
            com.Parameters.AddWithValue("@p_dis", DropDownList1.SelectedValue);
            //jalankan / eksekusi perintah
            com.ExecuteNonQuery();
        }
        else
        {
            SqlCommand com = new SqlCommand("insert into
            msbarang(NamaBarang,Stok,Harga,KdDistributor) values('" +
            txtNamaBarang.Text.Trim() + " ' ,'" + txtStok.Text.Trim() + " ' , '" +
            txtHarga.Text.Trim() + " ' , '" + DropDownList1.SelectedValue + " ')", con);
            com.ExecuteNonQuery();
        }
    }
    catch (Exception kesalahan)
    {
        MessageBox.Show(kesalahan.ToString());
    }
}
```

```

}
finally
{
    MessageBox.Show("Insert Data sukses");
    con.Close();
    con.Dispose();
}
}

```

Selamat akhirnya anda telah berhasil membuat halaman **Insert Data.aspx**, sekarang saatnya membuat halaman **System.Data.aspx**, halaman ini terpotong menjadi 2 bagian karena terlalu panjang ke bawah :

BELAJAR BERSAMA SURYO DESIGN

Home-> System.Data
 Halaman DataRelation

- Home
- Show Data
- Insert
- System.Data
- Binding

data pada data set barang pada data ke 2 sebelumnya : Suryo
 data pada data set barang pada data ke 2 saat ini : SuryoDesign

Nama Kolom yang ada pada table Distributor :
 Kolom ke : 1 adalah : KdDistributor
 Kolom ke : 2 adalah : NamaDistributor

Data Distributor :

KdDistributor	NamaDistributor
1	PT. A
2	PT. B
3	PT. C

← **DataRow dari datatable**
 ← **NamaColumn dari dataset.Tables**
 ← **data dari dataset TableDistributor**

Data Barang :

KdBarang	NamaBarang	Stok	Harga	KdDistributor
1	Gelas	200	5000	2
2	Piring	500	7500	3
3	Sendok	100	3000	3
4	Mangkok	175	10000	2
5	Lap Tangan	11	6000	1

← **Data dari dataset Tables Barang**

Data Pegawai :

Suryo
Design
Ario
Creative

← **Binding ke List Box**

Data Pembeli :

titi kamal

← **Binding ke DropDownList**

Disini ada beberapa Point yang akan kita dapat dari halaman **System.Data.aspx** ini, bila sebelumnya kita hanya coba binding ke Gridview disini kita akan melakukan ke pada komponen yang lainnya , berikut point nya :

- 1.Menggunakan DataRow.
- 2.Menggunakan DataColumn
- 3.Menggunakan DataRelation
- 3.Binding Ke ListBox
- 4.Binding ke DropDownList

Oke berikut adalah behind code dari halaman **System.data.aspx.cs** :

Seperti biasa , buat variable dan objek yang akan digunakan :

```
public partial class Support : System.Web.UI.Page
{
    private string SuryoCon =
    ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
    SqlDataAdapter da;
    DataSet ds;
    SqlCommand com = new SqlCommand();
}
```

Kemudian buat function panggil data :

```
private void panggilData()
{
    SqlConnection con = new SqlConnection(SuryoCon);
    try
    {
        com.CommandText = "select * from msbarang";
        com.Connection = con;
        da = new SqlDataAdapter(com);
        ds = new DataSet();
        //fill data ke dalam ds barang
        da.Fill(ds, "barang");
        com.CommandText = "select * from msdistributor";
        //fill data ke dalam ds distributor
        da.Fill(ds, "distributor");
        //membuat DataColumn
        DataColumn parentChild =
        ds.Tables["barang"].Columns["KdDistributor"];
        DataColumn parentColumn =
        ds.Tables["distributor"].Columns["KdDistributor"];
        //membuat dataRelation
        DataRelation relasiParent;
        //mendefinisikan isi dari dataRelation
        //perlu kita perhatikan disini
        //"NewOrder" merupakan relation name
        //kemudian kolom ke 2 merupakan column di mana primary key berada
        //dan kolom ke 3 merupakan foreign key yg menunjuk PK,
```

```
//bila anda salah meletakkan akan menyebabkan error
relasiParent = new DataRelation("NewOrder", parentColumn,
parentChild);
//membuat relasi
ds.Relations.Add(relasiParent);

//menentukan datasource, data member
//kemudian membinding data
GridView1.DataSource = ds;
GridView1.DataMember = "barang";
GridView1.DataBind();
GridView2.DataSource = ds;
GridView2.DataMember = "distributor";
GridView2.DataBind();
//menutup koneksi
con.Close();
}
catch (Exception kesalahan)
{
    MessageBox.Show(kesalahan.ToString());
}
finally
{
    con.Dispose();
}
}
```

Kemudian behind Code ketika halaman **System.data.aspx.cs** di load :

```
protected void Page_Load(object sender, EventArgs e)
{
    //memanggil function panggilData
    panggilData();

    //mengganti isi dari nama barang pd ds.tables[0] merupakan indeks
    dimana
    //data table yang pertama kali dibuat , yaitu Tables Barang
    //, dan tables distributor menjadi indeks ke 1 karena
    //diadd setelah tables barang

    ds.Tables[0].Rows[2]["NamaBarang"] = "Suryo";
    LabelHeader.Text = "data pada data set barang pada data ke 2
    sebelumnya : " + ds.Tables[0].Rows[2]["NamaBarang"].ToString() + "<br/>";
    ds.Tables[0].Rows[2]["NamaBarang"] = "SuryoDesign";
    LabelHeader.Text += "data pada data set barang pada data ke 2 saat
    ini : " + ds.Tables[0].Rows[2]["NamaBarang"].ToString() + "<br/>";
    LabelHeader.Text += "<br/>Nama Kolom yang ada pada table Distributor
    : <br/>";
    int i = 1;
    //membaca column yang ada pada Tables distributor
    foreach (DataColumn column in ds.Tables["distributor"].Columns)
    {
        LabelHeader.Text += "Kolom ke : " + i + " adalah : " +
        column.ColumnName + "<br/>";
        i++;
    }
}
```

```
}  
  
    //membuat tables baru pada dataset  
    ds.Tables.Add("Artis");  
    //buat column baru pada dataset  
    ds.Tables["Artis"].Columns.Add("Nama");  
    ds.Tables["Artis"].Columns.Add("Daerah");  
  
    //mendefinisikan data row dan membuat row baru  
    DataRow rowBaru = ds.Tables["Artis"].NewRow();  
    //mengisi row baru dengan data yang ada  
    rowBaru["Nama"] = "Suryo";  
    rowBaru["Daerah"] = "Lampung";  
    ds.Tables["Artis"].Rows.Add(rowBaru);  
  
    rowBaru = ds.Tables["Artis"].NewRow();  
    rowBaru["Nama"] = "Design";  
    rowBaru["Daerah"] = "Jakarta";  
    ds.Tables["Artis"].Rows.Add(rowBaru);  
  
    rowBaru = ds.Tables["Artis"].NewRow();  
    rowBaru["Nama"] = "Ario";  
    rowBaru["Daerah"] = "Jawa";  
    ds.Tables["Artis"].Rows.Add(rowBaru);  
  
    rowBaru = ds.Tables["Artis"].NewRow();  
    rowBaru["Nama"] = "Creative";  
    rowBaru["Daerah"] = "Bandung";  
    ds.Tables["Artis"].Rows.Add(rowBaru);  
  
    //binding data dari dataset yang baru dibuat  
    //ke dalam listbox  
    ListBox1.DataSource = ds.Tables["Artis"];  
    ListBox1.DataTextField = "Nama";  
    ListBox1.DataBind();  
  
    //membuat arraylist  
    ArrayList artis = new ArrayList();  
    artis.Add("titi kamal");  
    artis.Add("asmiranda");  
    artis.Add("sheila");  
    artis.Add("marshanda");  
    artis.Add("leoni");  
    artis.Add("baby");  
  
    //binding arrayList ke dalam dropDownList  
    DropDownList1.DataSource = artis;  
    DropDownList1.DataBind();  
}
```

Kemudian kita akan membuat bagian terakhir dari Latihan kali ini yaitu halaman dari **Binding.aspx** :

[Home](#) -> [Binding](#)
Halaman Binding
Home
Show Data
Insert
System.Data
Binding

Selamat Belajar ASP.NET Mr : Suryo / SuryoDesign
Teruslah Mencoba dan jgn pernah menyerah ^^.

Single Value Data Binding

Pilih data dengan PostBack

Gelas
KdBarang 1
NamaBarang Gelas
Stok 200
Harga 5000
KdDistributor 2

Show data
dengan Postback

Pilih data dengan QueryString

Mangkok
Pilih Data
KdBarang 4
NamaBarang Mangkok
Stok 175
Harga 10000
KdDistributor 2

Show Data Dengan Query String

Apa yang akan kita pelajari pada halaman Binding.aspx ini ? berikut adalah point-pointnya :

- 1.SingleValueBinding
- 2.Show Data dengan PostBack
- 3.Show data dengan QueryString
- 4.Modifikasi SqlDataSource
- 5.Menggunakan `<asp:QueryStringParameter>`
- 6.Menggunakan `<asp:ControlParameter>`

Untuk bagian ini mungkin sedikit membingungkan , karena itu saya akan memberikan sedikit pengertian melalui kata-kata .

SingleValueBinding intinya tidak lebih dari sekedar dynamic text yang dapat menampilkan text yang berasal dari behindcode atau variable yang sudah di definisikan , atau sekedar teks string yang langsung ingin di definisikan di dalamnya , dan dapat juga menampilkan keterangan lainnya.

Berikut adalah contoh SingleValueBinding yang terletak pada halaman sisi design pada .aspx dan bukan terletak pada behindCode :

```
<%# "Selamat Belajar ASP.NET Mr : " %>
<%# namaPengguna + " / " %>
<% PesanTersembunyi(); %>
```

Beberapa variable dan fungsi tersebut kita dapat dari behindcode berikut :

```
public partial class _Default : System.Web.UI.Page
{
    public string namaPengguna;
    public void PesanTersembunyi()
    {
        Response.Write("SuryoDesign. <br / >");
        Response.Write("Teruslah Mencoba dan jgn pernah menyerah ^^.");
    }

    protected void Page_Load(object sender, EventArgs e)
    {
        namaPengguna = "Suryo";
    }
}
```

Dan kemudian selanjutnya adalah memodifikasi SqlDataSource, pertama drag komponen SqlDataSource dari toolbar , kemudian anda akan melihat code nya seperti berikut dalam keadaan normal dan belum di modifikasi :

```
<asp:SqlDataSource ID="SqlDataSource" runat="server"></asp:SqlDataSource>
```

Dan kita modifikasi menggunakan Connection string yang berasal dari Web.Config, berikut String pada Web.Config :

```
<connectionStrings>
<add name="SuryoConnection" connectionString="Data Source=SURYSTUDIO-
PC\SQLEXPRESS;Initial Catalog=SQL2005;Integrated Security=True"/>
</connectionStrings>
```

Kemudian setelah drag Toolbar dan membuat connection string pada Web.Config , saatnya modifikasi SqlDataSource , ada 2 SqlDataSource yang kita gunakan pada halaman Binding.aspx , yang pertama kita gunakan untuk POSTBACK , dan yang ke 2 untuk QueryString .

berikut SqlDataSource dengan postback , dapat kita lihat disana terdapat perintah select Command yang memilih data dari parameter @KdBarang, kemudian di mana kita mengisi parameter ini pertanyaanya ? kita mengisi nya dengan SelectParamater yang memiliki Name="KdBarang" dan memiliki PropertyName SelectedValue untuk memilih data yang berasal dari ControlID DropDownList2 :

```
<asp:SqlDataSource ConnectionString="<%$ ConnectionStrings:SuryoConnection
%>"
ProviderName="System.Data.SqlClient"
SelectCommand= "select * from msbarang where KdBarang=@KdBarang"
ID="SqlDataSource1" runat="server">
<SelectParameters>
<asp:ControlParameter ControlID="DropDownList2" Name="KdBarang"
PropertyName="SelectedValue" />
</SelectParameters>
</asp:SqlDataSource>
```

Berikut SqlDataSource2 dengan QueryString, masih sama dengan yang pertama kita memilih KdBarang berdasarkan parameter @KdBarang , kita mendapatkan ini dari QueryString yang berasal dari URL yang kita dapatkan dari URL contoh seperti www.default.aspx?ID=1 , dari URL berikut kita tangkap variable ID dan nilainya dengan QueryStringParameter seperti berikut :

```
<asp:SqlDataSource ConnectionString="<%$ ConnectionStrings:SuryoConnection
%>"
ProviderName="System.Data.SqlClient"
SelectCommand= "select * from msbarang where KdBarang=@KdBarang"
ID="SqlDataSource2" runat="server" >
<SelectParameters>
<asp:QueryStringParameter Name="KdBarang" QueryStringField="id"
DefaultValue="1" />
</SelectParameters>
</asp:SqlDataSource>
```

Yup2 tidak begitu susah bukan ? awalnya memang terlihat membingungkan tapi akhirnya nanti akan menjadi biasa bila kita sering menggunakannya ^^.

berikut adalah behind Code dari halaman **Binding.aspx.cs** ketika halaman di load :

```
protected void Button1_Click(object sender, EventArgs e)
{
//direct ke halaman Binding.aspx dengan ID
//yg berasal dari DropDownList yang dipilih
Response.Redirect("Binding.aspx?ID="+ DropDownList3.SelectedValue + "");
}
```

Kemudian buat objek dan komponen yang dibutuhkan dan buat function PesanTersembunyi :

```
public partial class _Default : System.Web.UI.Page
{
public string namaPengguna;
public void PesanTersembunyi()
{
Response.Write("SuryoDesign. <br / >");
Response.Write("Teruslah Mencoba dan jgn pernah menyerah ^^.");
}
}
```

Berikut merupakan code ketika halaman **binding.aspx** di Load :

```
protected void Page_Load(object sender, EventArgs e)
{
    namaPengguna = "Suryo";
    string SuryoCon =
ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
    SqlConnection con = new SqlConnection(SuryoCon);

    con.Open();
    if (!IsPostBack)
    {
        SqlCommand com = new SqlCommand("select * from msbarang", con);
        DropDownList2.DataSource = com.ExecuteReader();
        DropDownList2.DataTextField = "NamaBarang";
        DropDownList2.DataValueField = "KdBarang";
    }
    this.DataBind();

    con.Close();
    if (!IsPostBack)
    {
        //menangkap variabel yang berasal dari QueryString pada URL
        string Pilih = Request.QueryString["ID"];
        con.Open();
        SqlCommand com2 = new SqlCommand("select * from msbarang", con);
        DropDownList3.DataSource = com2.ExecuteReader();
        DropDownList3.DataTextField = "NamaBarang";
        DropDownList3.DataValueField = "KdBarang";
        DropDownList3.DataBind();
        con.Close();
        if (Pilih != null)
        {
            DropDownList3.SelectedValue = Pilih;
        }
    }
}
```

Selamat anda telah berhasil membuat halaman ShowData.aspx , Insert.aspx ,System.Data.aspx dan Binding.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/>