

Part 12 – ADO.NET & SQL Server 2005

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

Website dalam menampilkan data memerlukan suatu DataSource yang berupa database , database sendiri dalam website umumnya digunakan MYSQL bagi yang di buat dengan PHP dan berskala kecil , dan untuk yang skala menengah ke atas digunakan SQL2005 dan Oracle yang dibuat dengan ASP.NET , Kecepatan dalam mengakses , Integritas Data , Concurrency dan terutama pada Security merupakan faktor yang mesti di perhatikan, dalam artikel ini saya akan mencoba untuk membahas Fundamental dari akses dan manipulasi data dengan menggunakan ADO.NET , bagi yang ingin mendownload artikel ini 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

ASP.Net merupakan salah satu Microsoft .Net Platform yang dapat membuat web application dengan cross language integration , dan komponen2 yang baru dan juga memiliki Class Library yang mengizinkan kita untuk handling error hingga analisis XML Document , dalam pertemuan kali ini kita akan membahas mengenai ADO.Net .

Berikut beberapa fitur ASP.Net yang menjadi kelebihan :

- A page and controls framework
- The ASP.NET compiler
- Security infrastructure

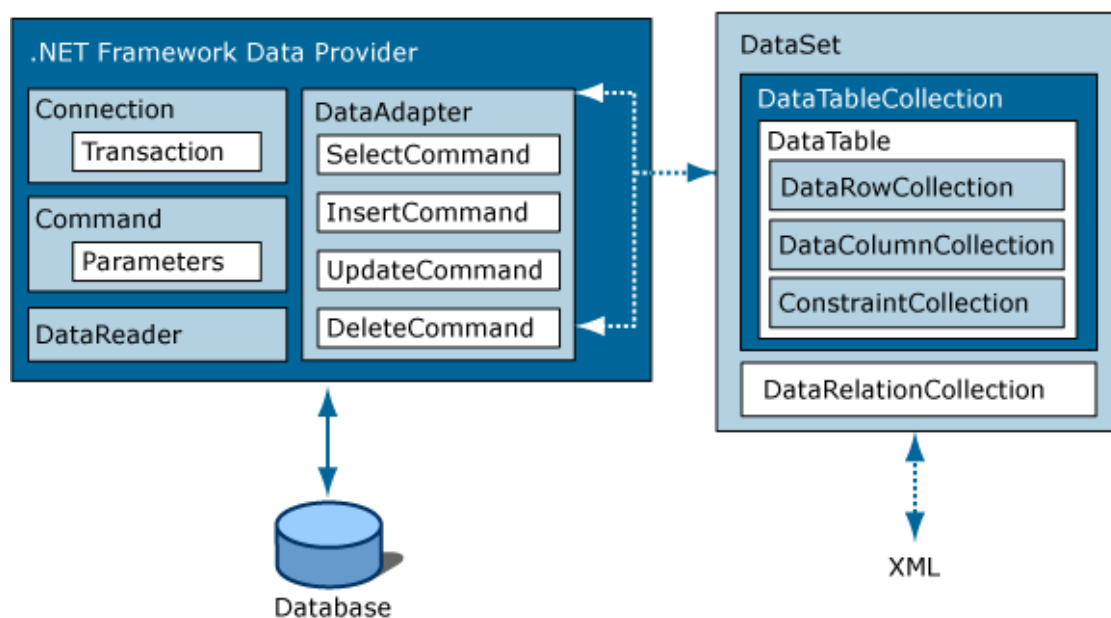
Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2009 IlmuKomputer.Com

- State-management facilities
- Application configuration
- Health monitoring and performance features
- Debugging support
- An XML Web services framework
- Extensible hosting environment and application life cycle management
- An extensible designer environment
- ADO.Net support (*ini yang akan menjadi pembahasan kita saat ini*)

ADO.Net memungkinkan kita untuk berinteraksi dengan relational database dan data source lainnya , ADO.Net merupakan salah satu teknologi yang kita gunakan untuk berkomunikasi dengan database , kita akan dapat menampilkan data , membuat log , serta proses manipulasi lainnya. ADO.NET merupakan sekumpulan class yang expose data akses servis bagi .Net Programmer , ADO.Net menyediakan sekumpulan komponen yang kaya akan fitur untuk membuat , data-sharing application dan merupakan bagian dari .Net Framework , menyediakan akses relational , XML dan application data , ADO.Net juga mendukung berbagai jenis development yang dibutuhkan , termasuk dalam pembuatan front-end database client dan middle-tier business object application , tools , languages dan browser internet. Kalau anda bertanya apa sih fitur yang terbaru di ADO.NET yaitu LINK atau Language Integrated Query (LINQ) , LINK To DataSet , Link To SQL , sabar saja nanti saya pada jug akan menerangkan tentang ini ^^

Berikut adalah gambaran Architecture dari ADO.Net (gambar saya ambil dari MSDN):



Mengakses sebuah database dalam internet merupakan scenario yang berbeda dengan mengakses database yang berada pada desktop atau client/server program. Kebanyakan developers mengembangkan aplikasi database pada desktop application dan kemudian dengan serius mencoba untuk menyelesaikan masalah apa yang mereka hadapi dengan lingkup yang ada pada internet , dimana yang hanya pada computer local yang hanya memiliki sedikit client dan jaringan yang umumnya terhubung dengan kabel LAN tergolong bagus dan selalu terkoneksi , bagaimana dengan aplikasi yang berada di Internet ? logika coding yang terlalu panjang dan memakan banyak waktu akan membuat delay semakin lambat , koneksi internet yang kurang bagus dan halaman page yang terlalu besar untuk di Load pun menjadi hambatan , dan banyak nya user yang terhubung dan mengakses database bersama-sama adalah masalah yang paling banyak dihadapi . Web Application bukan hanya digunakan oleh ratusan bahkan hingga ribuan user yang secara bergantian ,ini memakan memory pada server , oleh karena itu kita mesti limit resource seperti database connection dan mesti memiliki server yang memadai tentunya untuk hal ini .

Bila kita Design ASP.NET Application yang membutuhkan database connecton dan kita melakukan Hold terhadap database itu selama beberapa menit saja , user mungkin menyadari ada perlambatan atau database di Lock secara keseluruhan sehingga tidak dapat diakses , kita mesti berhati-hati dalam menentukan database concurrency , seperti permasalahan recork lock dan konflik ketika sedang meng-update data dan untuk menyediakan data yang konsisten kepada semua user.

Seperti ketahui , HTTP merupakan stateless protocol , dimana ketika user me-request halaman ke dalam server , koneksi kemudian dibuat , kode di press dan kemudian di balikkan ke user menjadi sebuah halaman HTML , dan koneksi kemudian di putus setelah data kembali kepada user , dengan data-driven maka kitadisini akan mencoba menggunakan ADO.Net yang memiliki banyak fitur dan memudahkan kita untuk menampilkan data (display/ retrieve), manipulasi , dan lain2.

ADO.Net memiliki beberapa perbedaan dengan pendahulunya seperti ADO , sekedar informasi saja , ADO banyak digunakan pada ASP Classic (ASP pendahulunya).

Yang paling utama dari ADO.Net yaitu memiliki **DataSet**, **Disconnected Access** , **XML Integration dll** ,

DataSet : DataSet ini sendiri berfungsi sebagai cache / penampung dari informasi yang telah di ambil dari database berdasarkan query yang kita miliki "seperti select * from msuser", setelah kita mendapatkan data ini , data ini akan di simpan di dalam dataset , kita dapat memberikan nama berdasarkan kategori dengan dataset.Table dari data yang kita ambil dan kita juga dapat membuat relasi dari database yang ada.

DisconnectedAccess : Disconnected Akses adalah salah satu fitur yang paling penting yang dimiliki oleh ADO.Net dan diharapkan sebagai contoh terbaik pada teknologi .Net dalam mengakses data yang ada. pada database sebelumnya, untuk mengakses data , kita membuka koneksi kemudian Hold Connection tersebut ke database server ketika code kita bekerja , ini biasa kita sebut dengan Live Connection yang selalu digunakan untuk memastikan data terbaru , dan bahkan kita dapat melihat pergantian yang dilihat oleh user secara real time, tetapi permasalahnya database server hanya menyediakan user yang terbatas yang dapat berkoneksi secara bersama-sama (limited connections) sebelum mereka menutup

/ reject koneksi yang ada . selama koneksi di buka , besar kemungkinan user lain yang ingin mengakses data tidak akan dapat mengaksesnya karena data tersebut di LOCK oleh user yang sedang membukanya. ADO.Net memiliki sedikit perbedaan dengan metode di atas yang di miliki oleh ADO pendahulunya, dalam ADO.Net ketika kita membuat koneksi database , kemudian setelah koneksi dibuat kita dapat menutup koneksi jauh lebih cepat dibanding pendahulunya karena data yang kita miliki kita Fill ke dalam DataSet object dengan menyain informasi yang ada dari database , kemudian kita emnutup koneksi tersebut dan mengakses data yang ada melalui dataset tadi , dan kemudian kita dapat melakukan manipulasi data tanpa ragu-ragu, dan bila kita melakukan update data maka kita akan melakukan reconnect ke database untuk menyetujui (commit) semua perubahan yang terjadi.

XML Integration

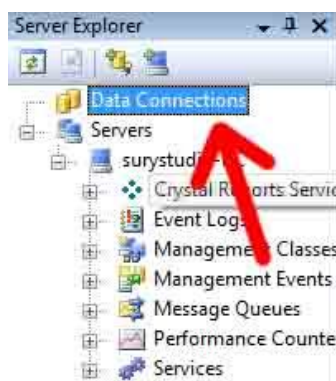
ADO.NET sangat support XML dengan berbagai fitur yang ada , karena pada umumnya website untuk membaca XML selalu membaca secara langsung (direct) , tapi dalam ADO.Net kita memiliki DataSet untuk menampilkan semua perubahan yang ada serta menampilkan data yang berasal dari XML document , kita dapat modifikasi values , remove rows dan menambah row (record baru) dan DataSet akan otomatis meng-updatenya.

Saat ini kita akan mencoba meng-integrasikan ADO.Net dengan database SQL Server 2005 Express Edition. SQL Server 2005 Express Edition dapat kita download dengan Cuma-Cuma di situs Microsoft. SQL Server 2005 Express ini Free untuk digunakan dan di-distribuksikan tetapi memiliki batasan seperti berikut :

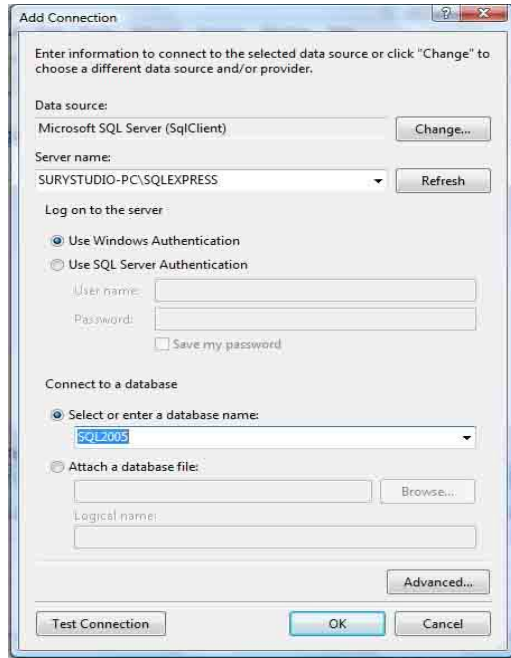
1. Hanya dapat menggunakan 1 CPU
2. Maximum 1 GB Ram
3. Database Maximum 4GB
4. Graphic Tools tidak disediakan
5. Masih banyak fitur lain selain di atas , anda dapat mencari sendiri yang lainnya ^^

Sekedar pengetahuan saja , kita tidak mesti mengedit database melalui SQL Server Management Studio tetapi kita dapat memodifikasi nya melalui Visual Studio , Berikut adalah caranya :

1. Klik ToolBar View > Pilih Server Explorer
2. Akan Muncul Menu Server Explorer , Klik Kanan Data Connections dan Add New Connections

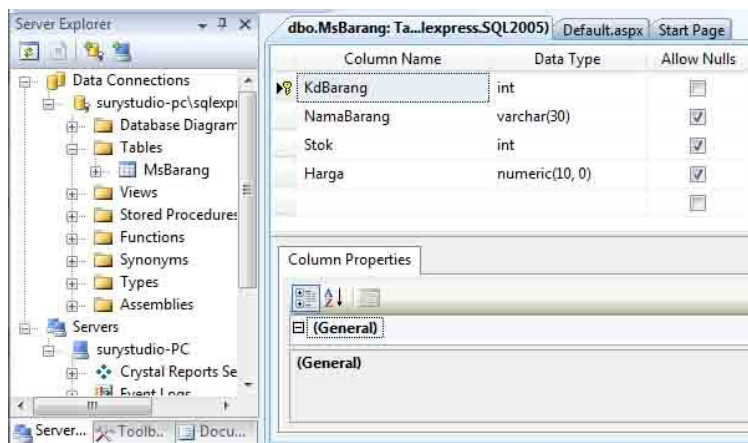


3. Kemudian Pilih Server Name dan pilih DatabaseName dan untuk yang menggunakan SQLEXPRESS jangan lupa menambahkan SQLEXPRESS di belakangnya.

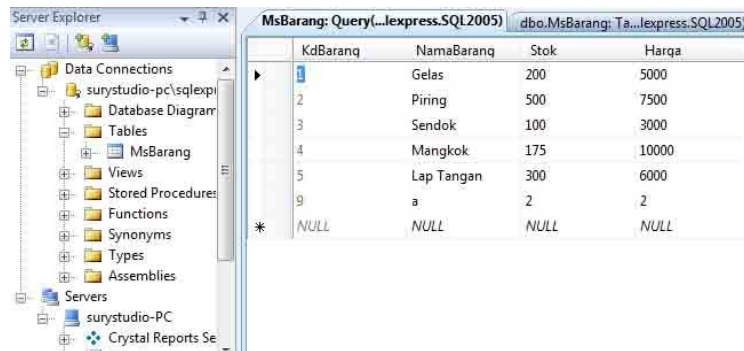


4. Pilih Test Connection dan bila Sukses kemudian OK , bila sukses maka di Menu bagian kiri akan muncul server yang tadi anda buat

5. Expand Node yang ada , pilih Server yang telah kita add, masuk ke menu Tables , dan pastikan anda sudah memiliki Sebuah Table dengan data di dalamnya . dan kemudian anda dapat merubah structure yang ada



6. Anda bukan hanya dapat merubah Structure yang ada , anda dapat insert , update , hingga delete data dan anda juga dapat menambah trigger dan lain-lain , benar2 seperti masuk SQL Management Studio bukan ^^.



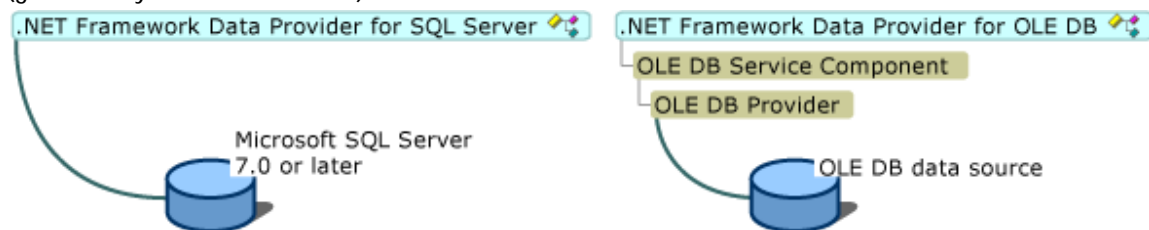
KdBarang	NamaBarang	Stok	Harga
1	Gelas	200	5000
2	Piring	500	7500
3	Sendok	100	3000
4	Mangkok	175	10000
5	Lap Tangan	300	6000
9	a	2	2
*	NULL	NULL	NULL

Nanti kita akan mempelajari beberapa kegunaan seperti **DataSet** , **DataTable** , **DataRow**, dan **DataRelation** dan semua semua akan kita gunakan untuk berkoneksi dengan data yang lebih spesifik dengan data source yang akan menggunakan , connection , command dan **DataReader**.

Penting untuk mengetahui macam-macam data Provider yang ada, Microsoft memiliki empat data provider seperti berikut :

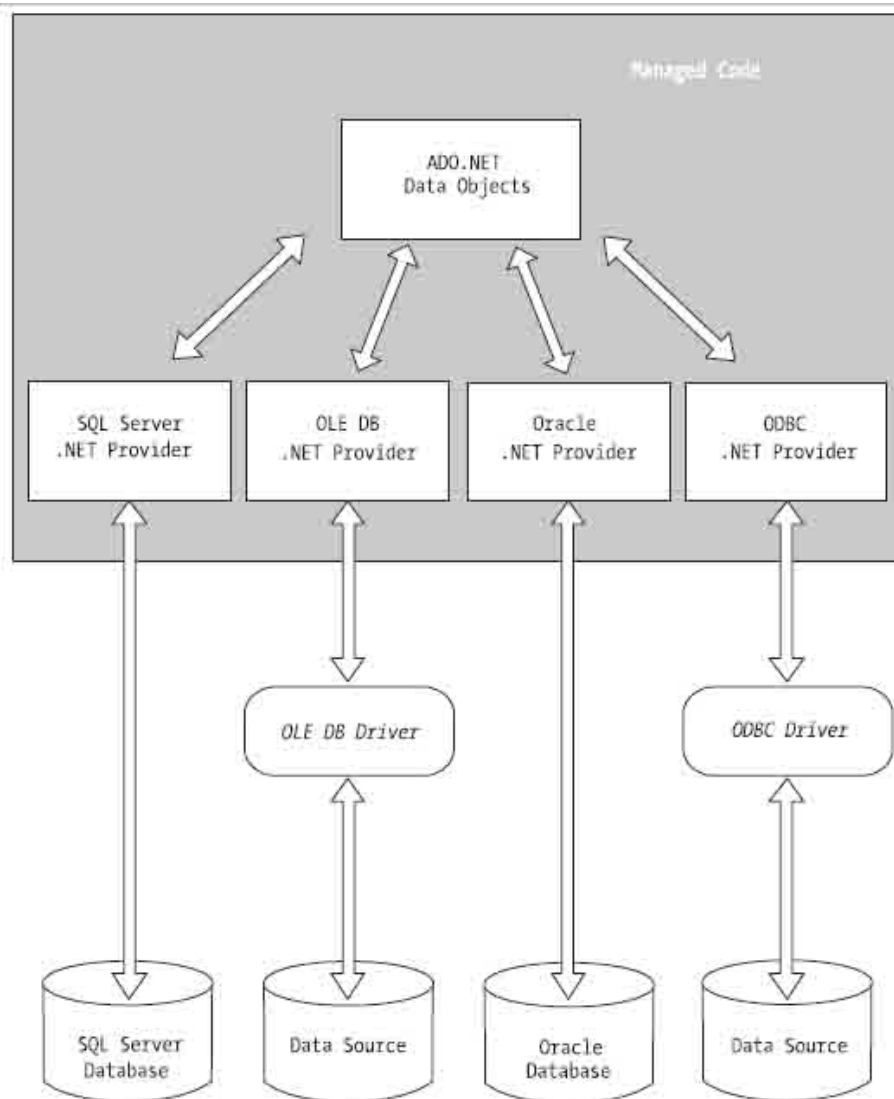
1. SQL Server provider : menyediakan akses yang optimal untuk SQL Server database (versi 7 ke atas)
2. OLE DB Provider : menyediakan akses ke datasource yang dapat support OLE DB , seperti akses
3. Oracle provider : Menyediakan akses yang optimal ke apada Oracle database (version 8i or later)
4. ODBC provider : menyediakan akses ke dalam data source yang memiliki ODBC (Open Database Connectivity) Driver

Berikut adalah Gambaran dari SQLDriver dibandingkan dengan OLEDB Driver :
(gambar saya ambil dri MSDN)



Dari Vendor pembuat database dan Third-Party developers telah membuat masing-masing ADO.Net provider yang dapat digunakan dengan cara yang sama yang terdapat di dalam .NET Framework , ketika memilih provider, anda mesti memastikan dataprovider yang terbaik bagi database anda , apabila tidak ada provider yang cocok , anda dapat menggunakan OLDE DB provider , karena OLE DB teknologi telah dikembangkan beberapa tahun lalu sebagai bagian dari ADO , jadi dipastikan semua datasource menyediakan OLE DB Driver seperti SQL , ORACLE , ACCESS , MYSQL , DB2 dan lain-lain.

Berikut adalah diagram yang dari Data Provider yang ada (gambar diambil dari PDF yang ada pada referensi saya)



Untuk menggunakan ADO.NET anda mesti lebih dahulu mengetahui Namespaces dari masing-masing provider :

Namespace	Purpose
System.Data	Mengandung fundamental class / class dasar dari ADO.Net, seperti DataSet , DataRelation , DataTable dan lain-lain yang mengizinkan kita untuk memanipulasi data, class ini kudu ! (Wajib) diperlukan dalam ADO.Net ^^
System.Data.Common	Class ini digunakan oleh provider yang melakukan inherit dari data ini
System.Data.OleDb	Mengandung class yang digunakan untuk melakukan koneksi dengan OLE DB data source , termasuk OleDbCommand dan OleDbConnection
System.Data.SqlClient	Mengandung class untuk melakukan koneksi dengan SQL Server (versi 7 ke atas) , seperti SqlCommand dan SqlConnection , memiliki semua properties dan methods yang sama seperti System.Data.OleDb namespace. Perbedaannya hanya lebih optimal untuk SQL Server dan menyediakan performance yang lebih baik ketimbang dengan OLE DB layer.
System.Data.SqlTypes	Mengandung struktur dari SQL Server , dan spesifik data types seperti SqlMoney dan SqlDateTime, anda dapat menggunakan type ini untuk bekerja dengan SQL Server tanpa menginginkan konversi ke dalam nilai standard yang equivalent seperti Decimal dan DateTime , untuk menghindari masalah pada konversi data
System.Data.OracleClient	Digunakan untuk koneksi dengan Oracle Database seperti OracleCommand dan OracleConnection
System.Data.Odbc	Mengandung class yang dapat digunakan untuk melakukan koneksi dengan ODBC driver , seperti OdbcCommand dan OdbcConnection.

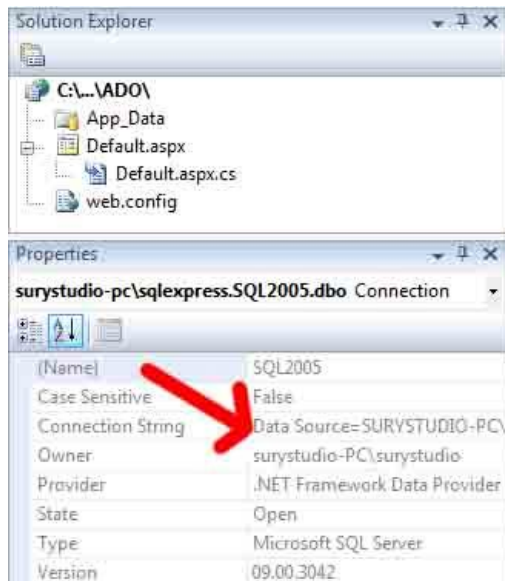
Ado.Net Data Provider Objects

	SQL Server .Net Provider	OLE DB .Net Provider	Oracle .Net Provider	ODBC .Net Provider
Connection	SqlConnection	OleDbConnection	OracleConnection	OdbcConnection
Command	Sql Command	OleDb Command	Oracle Command	Odbc Command
DataReader	SQLDataReader	OleDbDataReader	OracleDataReader	OdbcDataReader
DataAdapter	SQLDataAdapter	OleDbDataAdapter	OracleDataAdapter	OdbcDataAdapter

Sebelum anda menggunakan ADO.Net jangan lupa import ADO.NET Namespaces , contoh dalam SQL Provider , kita import 2 nameSpace pada bagian teratas dari behind code tambahkan 2 baris berikut :

```
using System.Data;
using System.Data.SqlClient;
```


setelah import namespace di atas baru kita kemudian membuat connection ke datasource , koneksi ini berisi string koneksi yang menunjukkan database server name and database name . pertanyaanya dari manakah kita mendapatkan database string ini ? kalau dari vb6 anda akan mendapatkan dari Data.UDL pastinya ^^ , tapi tadi sebelumnya kita sudah add Database melalui Server Explorer , maka sekarang anda dapat melihat properties dari database tersebut pada posisi kanan bawah database , jangan lupa klik database tersebut terlebih dahulu pada server explorer dan baru lihat kanan bawah ^^ , berikut contohnya :



Copy Connection tersebut dan lihat hasilnya :

```
Data Source=SURYSTUDIO-PC\\SQLEXPRESS;Initial Catalog=SQL2005;Integrated Security=True
```

String ini akan kita jadikan String yang akan kita letakkan di dalam SqlConnectionString seperti berikut
Letakkan String di dalam Connectionn String:

```
SqlConnection con = new SqlConnection("Data Source=SURYSTUDIO-PC\\SQLEXPRESS;Initial Catalog=SQL2005;Integrated Security=True");
```

Ketika anda compile ini akan mengalami ERROR !!!!! anda pasti bertanya kenapa ? bukankah saya sudah mengikuti prosedur yang ada , perhatikan pada "**SURYSTUDIO-PC\\SQLEXPRESS**" , string berikut memiliki Special Character yaitu single backslash (\) , kita bisa menggunakan double backslash atau tanda @ pada awal character untuk menon-aktifkan special character menjadi seperti berikut :

```
SqlConnection con = new SqlConnection(@"Data Source=SURYSTUDIO-PC\\SQLEXPRESS;Initial Catalog=SQL2005;Integrated Security=True");
```

Atau menjadi seperti berikut :

```
SqlConnection con = new SqlConnection("Data Source=SURYSTUDIO-PC\\SQLEXPRESS;Initial Catalog=SQL2005;Integrated Security=True");
```

Terserah anda ingin pakai yang mana di antara 2 cara di atas itu terserah anda ^^.

Bila anda perhatikan connection string tersebut memiliki beberapa bagian informasi seperti : DataSource , InitialCatalog , IntegratedSecurity , dan ada 1 lagi yaitu ConnectionTimeout yang tidak terdapat di atas.

DataSource : mengindikasikan nama server dimana datasource di lokasikan , bila kita menggunakan SQL Server 2005 Express Edition , kita akan menggunakan datasource namaserver\SQLEXPRESS , karena instance name dari SQL kita adalah SQLEXPRESS

Initial catalog : adalah nama database yang akan kita akses , kita dapat mengubah initial catalog ini dengan menggunakan Connection.ChangeDatabase() Method , nanti akan saya bahas ini.

Integrated Security : mengindikasikan kita ingin melakukan koneksi dengan SQL Server menggunakan windows useraccount atau banyak disebut dengan Windows Authentication , dan ada juga SQL Server Authentication yang menggunakan ID dan Password pada database.

Connection Timeout : mengindikasikan berapa lama kita akan menunggu database kita di koneksikan hingga terhubung , Timeout ini dalam satuan second , defaultnya adalah 15 detik , jadi bila dalam 15 detik kita tidak bisa melakukan koneksi dengan sukses maka koneksi kita akan di putus dari server.

Setelah membahas Connection sekarang Saya akan membahas type-type authentication yang ada :

1. SQL Server Authentication : SQL Server menyediakan user account information pada database , ini digunakan untuk menentukan user mana saja yang diizinkan menggunakan database tersebut.

2. Windows Authentication : merupakan Authentication yang dimana anda telah berhasil login ke dalam windows maka anda akan dapat menggunakan database tanpa memerlukan User ID dan Password.



Dan ada sedikit tips dan trik dari saya untuk menyimpan Connection string di dalam Web.Config , karena saya lebih suka menyimpan string koneksi disini ? anda bertanya kenapa alasannya ? bila saya suatu saat berpindah server atau program sudah di deploy , dan di tempat kita tidak memiliki Visual Studio 2008

untuk membuka Code yang kita miliki , kita dapat membuka Web.Config dengan notepad dan menggantinya tanpa perlu merombak program utama yang kita miliki.

Berikut Web.Config pada bagian Element connectionString :

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

Dan bagaimana cara kita mengambil Connection string dari web Config ini ? Berikut caranya menggunakan ConfigurationManager :

```
string SuryoCon =
ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
```

Kita akan membuat latihan dasar menampilkan data seperti berikut :

```
Server Version : 09.00.3042
State Connection : Open
Timeout : 15
Database : SQL2005
DataSource : SURYSTUDIO-PC\SQLEXPRESS
Statistic Enabled : False
Workstation ID : SURYSTUDIO-PC
```

Berikut adalah BehindCode dengan C# nya :

```
string SuryoCon =
ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
SqlConnection con = new SqlConnection(SuryoCon);
try
{
con.Open();
Label1.Text = "<b>Server Version : </b>" + con.ServerVersion + "<br>";
Label1.Text += "<b>State Connection : </b>" + con.State.ToString() + "<br>";
Label1.Text += "<b>Timeout : </b>" + con.ConnectionTimeout.ToString() +
"<br>";
Label1.Text += "<b>Database : </b>" + con.Database.ToString() + "<br>";
Label1.Text += "<b>DataSource : </b>" + con.DataSource.ToString() + "<br>";
Label1.Text += "<b>Statistic Enabled : </b>" +
con.StatisticsEnabled.ToString() + "<br>";
Label1.Text += "<b>Workstation ID : </b>" + con.WorkstationId.ToString() +
"<br>";
}
catch(Exception kesalahan)
{
Label1.Text = "ADA MASALAH PADA DATABASE : ";
Label1.Text += kesalahan.Message;
}
```

```
finally  
{  
con.Close();  
con.Dispose();  
}
```

Apapun yang anda lakukan pegang konsepnya dari semua nya , konsep adalah intinya , percuma bila anda berhasil membuat tapi tidak mengetahui asal – usul atau aktivitas yang ada dan hanya sekedar ingin tampil saja.

Ketika anda membuka koneksi dengan **Open()** method , saat ini juga kita memiliki live onnection ke dalam database kita , salah satu prinsip yang penting bagaimana kita melakukan proses pembacaan data yang tercepat , caranya adalah dengan membuat code yang efisien dan membuat flow yang baik agar dapat mengurangi waktu yang akses, setelah anda membuka koneksi maka kemudian segeralah tutup koneksi tersebut setelah anda mengambil data yang ada di dalamnya dengan menggunakan **Close()** method agar site berjalan dengan efisien, atau ada cara lain dengan menggunakan **Dispose** untuk menghapus objek atau juga dengan menggunakan **using** , seperti berikut :

```
string SuryoCon =  
ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;  
SqlConnection con = new SqlConnection(SuryoCon);  
using(con)  
{  
con.Open();  
Label1.Text = "<b>Server Version : </b>" + con.ServerVersion + "<br>";  
Label1.Text += "<b>State Connection : </b>" + con.State.ToString() + "<br>";  
Label1.Text += "<b>Timeout : </b>" + con.ConnectionTimeout.ToString() +  
"<br>";  
Label1.Text += "<b>Database : </b>" + con.Database.ToString() + "<br>";  
Label1.Text += "<b>DataSource : </b>" + con.DataSource.ToString() + "<br>";  
Label1.Text += "<b>Statistic Enabled : </b>" +  
con.StatisticsEnabled.ToString() + "<br>";  
Label1.Text += "<b>Workstation ID : </b>" + con.WorkstationId.ToString() +  
"<br>";  
}
```

Kegunaan dari using di sini adalah untuk menghapus objek Con setelah Semua yang berada dalam cakupan **Using** selesai dijalankan.

Setelah kita mencoba untuk menampilkan data yang berasal dari server sebagai latihan maka sekarang kita akan mencoba menampilkan data yang berasal dari database, sekarang kita akan menggunakan beberapa Command baru seperti SqlCommand , SqlDataReader dan SqlDataAdapter.

Untuk membuat perintah kita dapat membuat menjadi seperti ini :

```
SqlCommand com = new SqlCommand();  
com.Connection = con;
```

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2009 IlmuKomputer.Com

```
com.CommandText = "select * from msbarang";
```

Atau Seperti berikut :

```
SqlCommand com = new SqlCommand("select * from msbarang ", con);
```

Kemudian untuk membaca data anda dapat melakukan perintah seperti berikut :

```
con.Open();  
SqlCommand com = new SqlCommand("select * from MsBarang ", con);  
SqlDataReader rd;  
rd = com.ExecuteReader();  
rd.Read();  
MessageBox.Show("Nama Barang : " + rd["NamaBarang"].ToString() + " , Harga :  
" + rd["Harga"].ToString()) ;
```

Perintah di atas kita gunakan untuk membaca data pada baris pertama , pertanyaanya bagaimana bila kita ingin membaca data pertama dan kedua ? anda tinggal majukan pointer sekali lagi untuk bergerak ke baris selanjutnya , gunakan Read() Method lagi menjadi seperti berikut :

```
con.Open();  
SqlCommand com = new SqlCommand("select * from MsBarang ", con);  
SqlDataReader rd;  
rd = com.ExecuteReader();  
rd.Read();  
MessageBox.Show("Nama Barang : " + rd["NamaBarang"].ToString() + " , Harga :  
" + rd["Harga"].ToString()) ;  
rd.Read();  
MessageBox.Show("Nama Barang : " + rd["NamaBarang"].ToString() + " , Harga :  
" + rd["Harga"].ToString()) ;
```

Dan sekarang bagaimana bila kita ingin membaca semua data yang ada ? anda dapat memodifikasinya sedikit dengan perintah while untuk melakukan perulangan hingga data terakhir seperti berikut :

```
con.Open();  
SqlCommand com = new SqlCommand("select * from MsBarang ", con);  
SqlDataReader rd;  
rd = com.ExecuteReader();  
while (rd.Read())  
{  
    MessageBox.Show("Nama Barang : " + rd["NamaBarang"].ToString() + " , Harga :  
" + rd["Harga"].ToString());  
}
```

Dan untuk membaca data pada kolom tertentu anda tidak hanya dapat menggunakan berdasarkan nama kolom tapi dapat juga menggunakan index menjadi seperti berikut :

```
MessageBox.Show("Nama Barang : " + rd[1].ToString());
```

Anda Tadi menggunakan perintah Execute Reader , dan anda mesti mengetahui ada Jenis Execute lainnya seperti berikut :

Item	Description
ExecuteReader	Untuk membaca semua data yang ada hingga akhir , Mengembalikan Rows , dan hanya untuk membaca saja (read) , dan bersifat OneWay(untuk membaca) dan tidak akan dapat melakukan eksekusi seperti insert , update , dan delete
ExecuteNonQuery	Digunakan untuk melakukan eksekusi command seperti Insert, Delete , Update , dan Set statement
ExecuteScalar	Mengembalikan satu nilai saja contoh seperti Menghitung jumlah data yang ada , menghitung banyaknya yang ada , menghitung data tertinggi dan data terendah .

Dan juga ada jenis2 Command Type yang kita gunakan pada Command seperti pada SqlCommand seperti berikut :

CommandType	Description
Text	Perintah Query SQL yang berupa Text
StoredProcedure	Nama Dari StoreProcedure yang telah kita buat sebelumnya pada SQL2005 dan dapat kita panggil dengan menggunakan nama StoreProcedure tersebut
TableDirect	Memanggil Data dengan menggunakan nama table

Oke sekarang pada halaman Default4.aspx kita akan coba menampilkan data dari database dan menambahkan ke dalam **RadioButtonList** dan kemudian menampilkan keterangan dari Data tersebut , pertama design form seperti berikut , jangan lupa untuk mengaktifkan **Enable AutoPostBack** dari **DropDownList** , dan letakkan 1 Label pada sisi paling bawah yang akan kita pakai untuk menampilkan keterangan :



RadioButton dan RadioButtonList memiliki perbedaan , dimana RadioButtonList bisa kita tambahkan datanya secara Dynamic sesuai keinginan kita , berikut caranya :

```
com.CommandText = "Select NamaPembeli , KdPembeli from msbuyer";
rd = com.ExecuteReader();
while (rd.Read())
{
RadioButtonList1.Items.Add(rd["NamaPembeli"].ToString());
}
rd.Close();
rd.Dispose();
```

Kemudian setelah menambahkan data secara dynamic ke dalam Radio Button , sekarang saatnya untuk mencari data yang berasal dari DropDownList yang dipilih (DropDownList1_SelectedIndexChanged). double klik DropDownList1 dan Berikut adalah perintah di dalamnya :

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
string SuryoCon =
ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
SqlConnection con = new SqlConnection(SuryoCon);
try
{
con.Open();
SqlCommand com = new SqlCommand("select * from MsBarang where KdBarang =' " +
DropDownList1.SelectedItem.Value + "' , con);
SqlDataReader rd;
rd = com.ExecuteReader();
while (rd.Read())
{
Label1.Text = "<b>Keterangan Barang : </b> <br/>";
Label1.Text += "Kode Barang = " + rd["KdBarang"].ToString() + "<br/>";
Label1.Text += "Nama Barang = " + rd["NamaBarang"].ToString() + "<br/>";
Label1.Text += "Harga Barang = " + rd["Harga"].ToString() + "<br/>";
Label1.Text += "Stok Barang = <i>" + rd["Stok"].ToString() + "</i><br/>";
}
if (rd != null)
{
rd.Close();
rd.Dispose();
}
}
catch (Exception kesalahan)
{
MessageBox.Show(kesalahan.ToString());
}
finally
{
con.Close();
con.Dispose();
}
}
```


Dan kemudian buatlah 1 function yang dapat di panggil sewaktu – waktu, Dan pastikan fungsi ini memiliki keyword **Private** di sisi depannya agar hanya dapat di panggil di form itu saja :

```
private void isiData()
{
    string SuryoCon =
    ConfigurationManager.ConnectionStrings["SuryoConnection"].ConnectionString;
    SqlConnection con = new SqlConnection(SuryoCon);
    try
    {
        con.Open();
        SqlCommand com = new SqlCommand("select namaBarang , Stok , KdBarang from
        MsBarang ", con);
        SqlDataReader rd;
        rd = com.ExecuteReader();
        while (rd.Read())
        {
            ListItem DaftarItem = new ListItem();
            DaftarItem.Text = rd["NamaBarang"] + " ( " + rd["Stok"] + " ) ";
            DaftarItem.Value = rd["KdBarang"].ToString();
            DropDownList1.Items.Add(DaftarItem);
        }
        rd.Close();
        com.CommandText = "Select NamaPembeli , KdPembeli from msbuyer";
        rd = com.ExecuteReader();
        while (rd.Read())
        {
            RadioButtonList1.Items.Add(rd["NamaPembeli"].ToString());
        }
        rd.Close();
        rd.Dispose();
    }
    catch (Exception kesalahan)
    {
        MessageBox.Show(kesalahan.ToString());
    }
    finally
    {
        con.Close();
        con.Dispose();
    }
}
```

Setelah fungsi ini di buat lalu kapan kita panggil fungsi ini ??? fungsi ini akan kita panggil ketika form di load dan bukan dalam kondisi IsPostBack (Kondisi dimana komponen yang memiliki fitur Enable AutoPostBack mengirim datanya kembali karena adanya perubahan pilihan pada komponen itu). berikut kita panggil dalam kondisi form di load :

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

```
else
{
if (RadioButtonList1.SelectedItem == null)
{
MessageBox.Show("Nama Pembeli Tidak Boleh Kosong");
return;
}
MessageBox.Show(DropDownList1.SelectedItem.Text);
MessageBox.Show(DropDownList1.SelectedItem.Value);
}
}
```

Dan anda mesti mengetahui cara untuk break dari Code untuk menghentikan program yang berjalan dengan menggunakan `return;` agar kode berhenti dan baris perintah di bawahnya tidak dijalankan. Dan di atas terdapat pilihan `SelectedItem.Text` dan `SelectedItem.Value`, apa perbedaannya? bila `Text` kita mendapatkan text yang ditampilkan tapi bila `Value` mendapatkan nilai di dalam komponen tersebut. `Text` dan `value` ini ditentukan pada saat kita menambah / add item ke dalam komponen tersebut.

Berikut adalah tampilan dari hasil data pada halaman `Default4.aspx` :

	<input type="radio"/> Didi
Nama Pembeli :	<input checked="" type="radio"/> Nunu
	<input type="radio"/> Lili
List Barang :	Piring (500)

Keterangan Barang :

Kode Barang = 2

Nama Barang = Piring

Harga Barang = 7500

Stok Barang = 500

Keterangan dari Masing-masing file :

`Default.aspx` : Koneksi dan keterangan Server

`Default2.aspx` : menggunakan Using

`Default3.aspx` : mengambil data dari database

`Default4.aspx` : mengambil dan menampilkan serta menggunakan fitur `AutoPostBack` dan `SelectedIndexChanged`

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/>