

# Tutorial Web Database Dengan ASP.NET dan VB.NET

**M. Choirul Amri**  
<http://www.choirulamri.or.id>

## **Lisensi Dokumen:**

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarluaskan 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**.

Dalam tulisan sebelumnya tentang “Pengantar ASP.NET” telah dipaparkan teknik-teknik dasar dalam pemrograman ASP.NET. Dalam tutorial ini dibahas pemrograman database dalam ASP.NET yang menggunakan ADO.NET dan VB.NET. Aplikasi berbasis database merupakan salah satu aplikasi web populer dan sering digunakan.

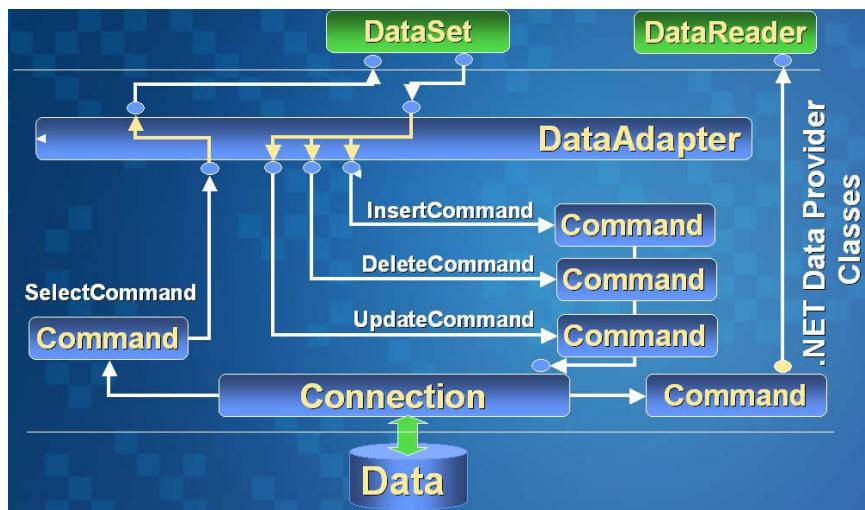
Tulisan ini tidak membahas teknik-teknik tingkat lanjut yang lebih detil. Dalam tulisan-tulisan mendatang akan dijelaskan lebih lanjut tentang berbagai teknik ADO.NET dan web control yang digunakan.

Tutorial ini ditujukan bagi mereka yang telah mengenal dasar ASP.NET dan pemrograman lain. Jika Anda masih pemula, silakan membaca tutorial dasar ASP.NET di <http://ilmukomputer.com/umum/choirul-dasar-asp.net.php>. Dalam tulisan ini digunakan IDE Visual Studio 2003 dan database SQL Server 2000.

## **1. Mengenal ADO.NET**

ADO.NET adalah teknik baru dalam mengakses database yang memanfaatkan berbagai library dalam .NET Framework. Dalam Visual Studio 6 telah dikenal ADO versi 2.7 yang dapat digunakan untuk mengakses berbagai database dari VB, C++, maupun Visual Interdev dan ASP. Sedangkan ADO.NET memberikan keleluasaan lebih karena namespace yang disediakan sangat konsisten antara satu bahasa dengan bahasa pemrograman lainnya.

Gambar berikut memberikan penjelasan arsitektur ADO.NET:



Pembahasan lebih dalam tentang obyek-obyek dalam ADO.NET tidak ditampilkan dalam tutorial ini. Anda dapat membacanya dalam tutorial Berjudul “Tutorial Pemrograman ADO.NET” yang akan segera terbit di IlmuKomputer.Com.

## Akses Data dengan ADO.NET

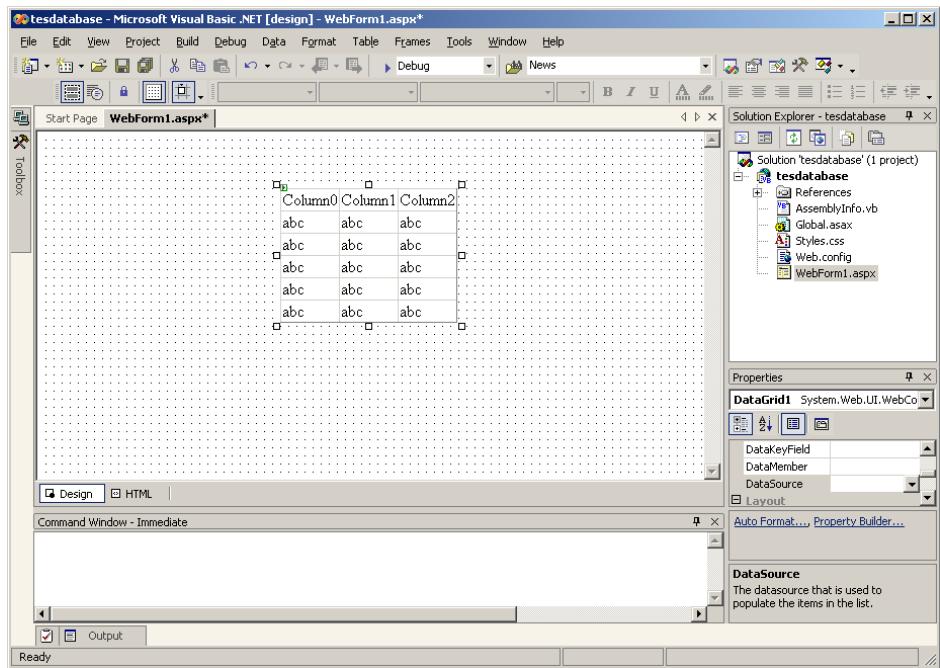
Untuk menampilkan data dari database menggunakan ADO.NET dilakukan urut-urutan sebagai berikut:

1. Membuka koneksi ke database dengan obyek Connection.
2. Melancarkan perintah SQL dengan obyek Command.
3. Tersedia 2 pilihan untuk menampilkan data, yaitu dengan DataSet atau DataReader. Obyek DataSet digunakan apabila Anda akan melakukan proses update dan manipulasi data di client. Sedangkan jika data hanya ditampilkan (read only) atau eksekusi perintah tanpa menampilkan data maka digunakan DataReader.
4. Mengaitkan data DataReader atau DataSet ke web control, misalnya DataGrid.
5. Menutup koneksi yang sudah tidak terpakai.

Tulisan ini hanya membahas ADO.NET yang menggunakan namespace System.SqlClient, yaitu untuk koneksi ke SQL Server secara native. Untuk penggunaan namespace lain seperti System.OleDb akan dijelaskan lebih lanjut dalam tulisan “Tutorial Pemrograman ADO.NET”.

## 2. Menampilkan Data ke DataGrid

Web Control DataGrid merupakan salah satu obyek yang paling banyak digunakan dalam aplikasi database. Baiklah sekarang kita menggunakan DataGrid untuk menampilkan data Customer di database NorthWind. Buka project baru dari VS 2003 dan berikan nama “tesdatabase” kemudian ganti nama file WebForm1.aspx menjadi customer.aspx. Drag-drop obyek kontrol DataGrid dari koleksi Web Forms pada toolbox sehingga dihasilkan halaman web dengan DataGrid sebagai berikut:



Sebelum menggunakan berbagai namespace yang berhubungan dengan SqlClient, jangan lupa untuk mengimportnya sebelum Class dideklarasikan. Dengan demikian penulisan akan menjadi lebih singkat dan tidak perlu mengulang namespace SqlClient.

```
Imports System.Data.SqlClient
```

Selanjutnya tuliskan kode berikut pada event Page\_Load, jangan lupa untuk menyesuaikan connection string yang digunakan sesuai setup database Anda yaitu menyangkut user, password, dan nama server.

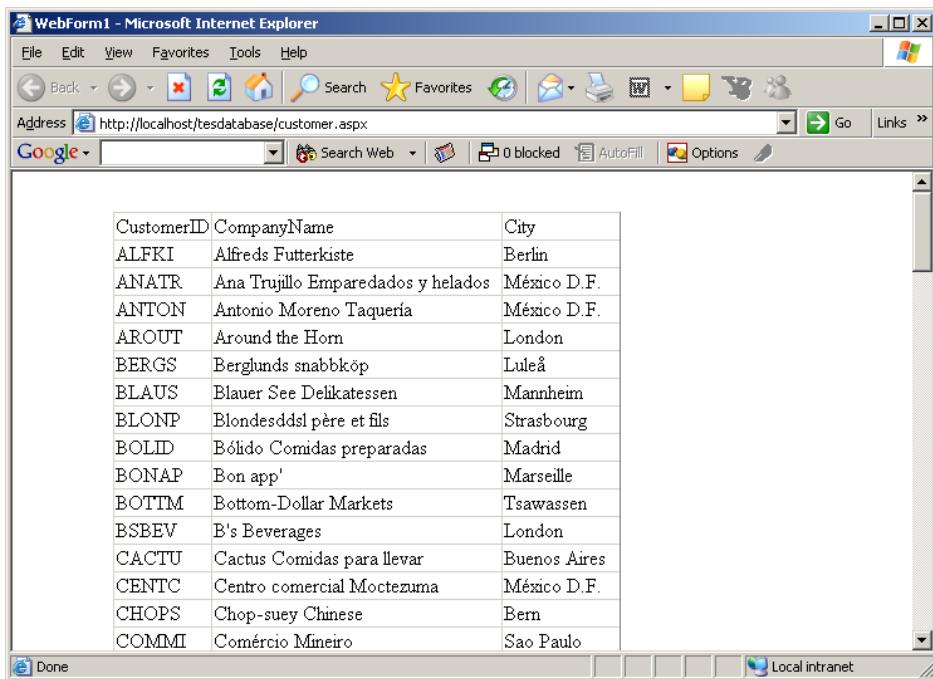
```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    'definisi string koneksi dan buka koneksi
    Dim strCn As String = "server=matrix; user=asp; password=asp; database=northwind"
    Dim cn As SqlConnection = New SqlConnection(strCn)

    'definisi string SQL dan buka data adapter
    Dim strSelect = "SELECT CustomerID, CompanyName, City FROM Customers"
    Dim adpSQL As SqlDataAdapter = New SqlDataAdapter(strSelect, strCn)

    'deklarasi dataset dan isi dataset dengan data
    Dim dsCustomers As New DataSet
    adpSQL.Fill(dsCustomers)

    'kaitkan data di dataset ke datagrid
    DataGrid1.DataSource = dsCustomers
    DataGrid1.DataBind()
    'tutup koneksi
    cn.Close()
End Sub
```

Jalankan customer.aspx maka datagrid tampil dengan 3 kolom data berikut:

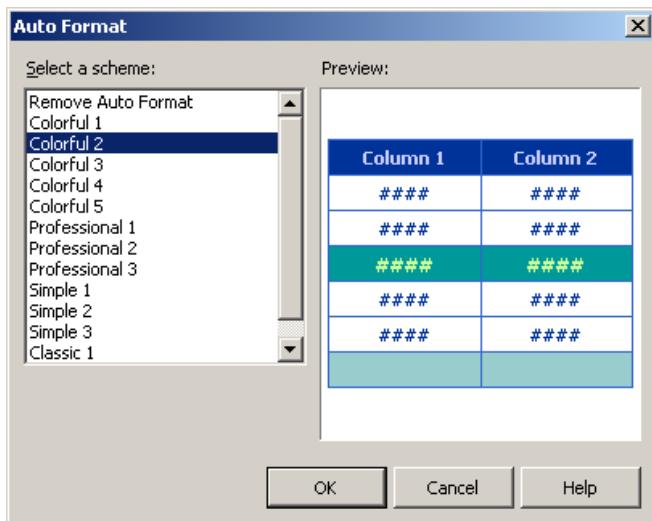


The screenshot shows a Microsoft Internet Explorer window titled "WebForm1 - Microsoft Internet Explorer". The address bar shows the URL "http://localhost/testdatabase/customer.aspx". The main content area displays a DataGrid control with three columns: CustomerID, CompanyName, and City. The data is as follows:

CustomerID	CompanyName	City
ALFKI	Alfreds Futterkiste	Berlin
ANATR	Ana Trujillo Emparedados y helados	México D.F.
ANTON	Antonio Moreno Taquería	México D.F.
AROUT	Around the Horn	London
BERGS	Berglunds snabbköp	Luleå
BLAUS	Blauer See Delikatessen	Mannheim
BLONP	Blondesddsl père et fils	Strasbourg
BOLID	Bólido Comidas preparadas	Madrid
BONAP	Bon app'	Marseille
BOTTM	Bottom-Dollar Markets	Tsawassen
BSBEV	B's Beverages	London
CACTU	Cactus Comidas para llevar	Buenos Aires
CENTC	Centro comercial Moctezuma	México D.F.
CHOPS	Chop-suey Chinese	Bern
COMMI	Comércio Mineiro	Sao Paulo

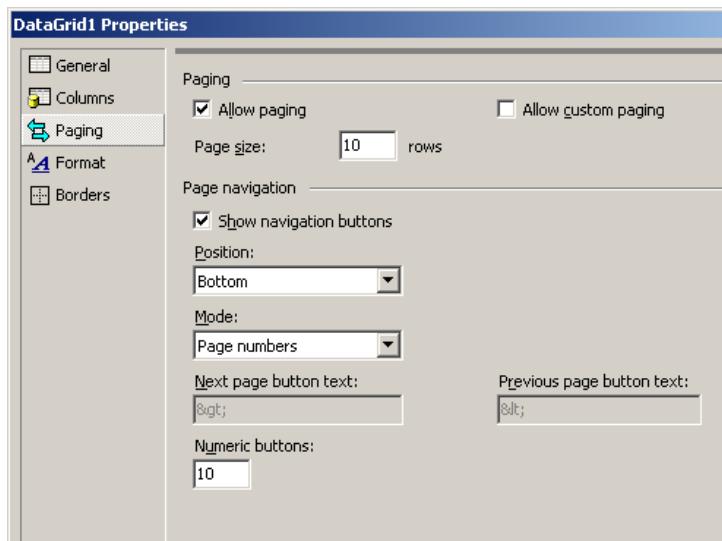
## 2.1 Merubah Tampilan Warna DataGrid

Jika Anda ingin memperindah tampilan DataGrid, maka telah tersedia berbagai template yang dapat dipilih. Klik kanan DataGrid dari mode design dan masuk ke menu Auto Format. Maka tersedia berbagai pilihan tampilan DataGrid.



## 2.2 DataGrid Paging

Anda juga dapat membagi tampilan DataGrid menjadi beberapa halaman. Dengan demikian penampilan menjadi lebih efisien karena user tidak perlu menggunakan scrolling vertikal untuk melihat data yang ada. DataGrid memiliki property builder yang juga dapat diakses dengan klik kanan DataGrid. Pilih menu Paging dan isikan kolom-kolom yang sesuai. Anda dapat menentukan berapa baris yang ditampilkan dalam satu halaman, serta tampilan paging baik berupa nomor halaman maupun tombol Previous/Next.



Sekarang apabila halaman tersebut dijalankan maka tampil halaman dengan paging. Tetapi ketika nomor halaman diklik maka tidak terjadi perubahan data. Mengapa? Karena masih ada satu langkah lagi, yaitu menyisipkan kode untuk melakukan binding data ke DataGrid setiap terjadi perubahan page index. Untuk itu ditambahkan kode di event `DataGrid1_PageIndexChanged`.

```
Private Sub DataGrid1_PageIndexChanged(ByVal source As Object, ByVal e As System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles DataGrid1.PageIndexChanged
    DataGrid1.CurrentPageIndex = e.NewPageIndex
    DataGrid1.DataSource = dsCustomers
    DataGrid1.DataBind()
End Sub
```

Dari kode diatas terlihat bahwa terjadi pengulangan kode untuk mengaitkan dataset ke DataGrid, yaitu sama dengan kode sebelumnya ketika event Load. Agar kode menjadi lebih efisian maka dapat dibuat sebuah method dengan nama `BindGrid` dan berisi kode untuk binding tersebut.

```
Sub BindGrid()
    'kaitkan data di dataset ke datagrid
    DataGrid1.DataSource = dsCustomers
    DataGrid1.DataBind()
End Sub
```

Dengan demikian Anda hanya perlu memanggil method `BindGrid` tersebut setiap kali diperlukan. Untuk mendukung penambahan method tersebut, maka perlu dilakukan sedikit perubahan dalam deklarasi variabel dan obyek-obyek ADO.NET. Berikut adalah kode program yang telah dimodifikasi secara lengkap:

```
Public Class WebForm1
    Inherits System.Web.UI.Page

    'definisi string koneksi dan buka koneksi
    Dim strCn As String = "server=matrix; user=asp; password=asp; database=northwind"
    Dim cn As SqlConnection = New SqlConnection(strCn)

    'deklarasi dataset
    Dim dsCustomers As New DataSet

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
        Handles MyBase.Load
            'definisi string SQL dan buka data adapter
            Dim strSelect = "SELECT CustomerID, CompanyName, City FROM Customers"
            Dim adpSQL As SqlDataAdapter = New SqlDataAdapter(strSelect, cn)

            'isi dataset
            adpSQL.Fill(dsCustomers)

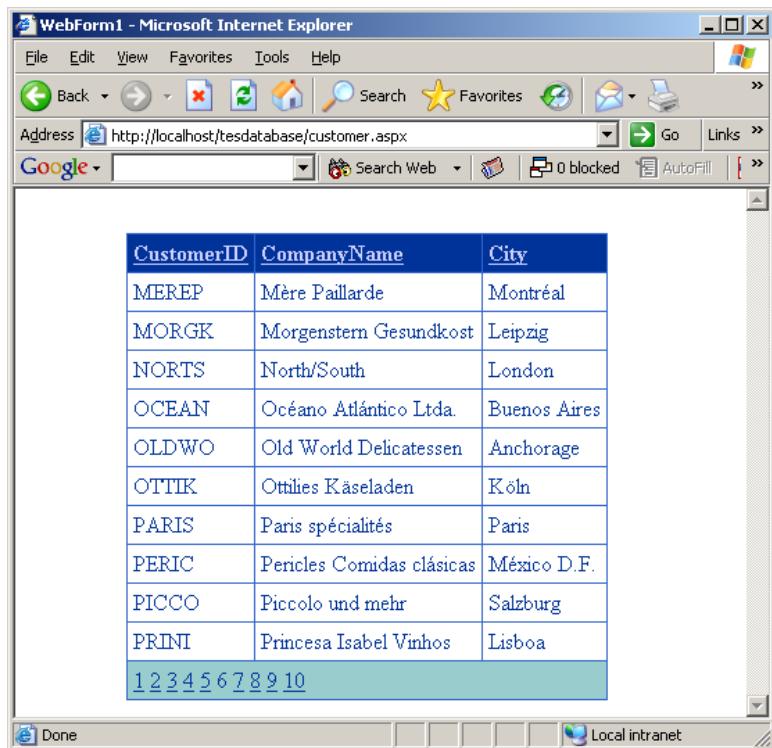
            'binding ke grid
            If Not IsPostBack Then
                BindGrid()
            End If

            'tutup koneksi
            cn.Close()
    End Sub

    'method untuk mengaitkan datagrid
    Sub BindGrid()
        'kaitkan data di dataset ke datagrid
        DataGrid1.DataSource = dsCustomers
        DataGrid1.DataBind()
    End Sub

    Private Sub DataGrid1_PageIndexChanged(ByVal source As Object, ByVal e As System.Web.UI.WebControls.DataGridPageChangedEventArgs) Handles DataGrid1.PageIndexChanged
        'mengaitkan dataset ke datagrid setiap terjadi perubahan paging
        DataGrid1.CurrentPageIndex = e.NewPageIndex
        BindGrid()
    End Sub

End Class
```



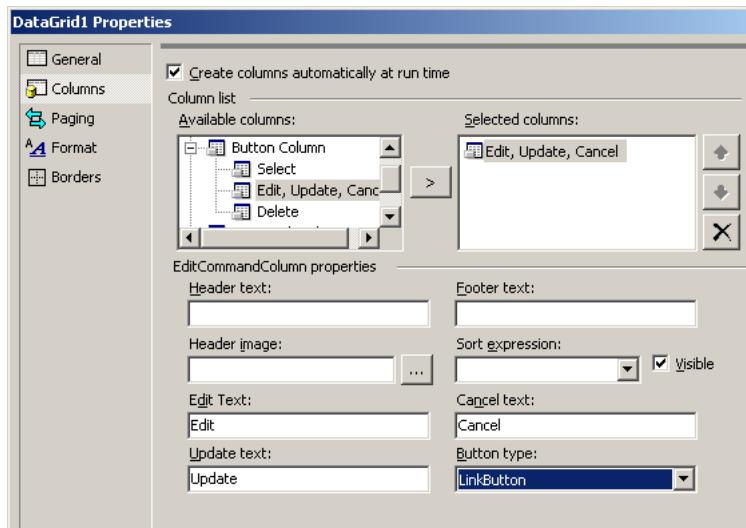
## 2.3 Edit dan Update Data dengan DataGrid

Kita telah membahas cara menampilkan data dengan DataGrid namun belum dapat mengedit maupun menghapus data di dalamnya. Operasi edit, hapus, dan simpan data dapat dilakukan menggunakan berbagai perintah ADO.NET yang digabungkan dengan berbagai event yang ada di dalam DataGrid tersebut.

### Mengedit Data di DataGrid.

Langkah paling awal adalah menetapkan nama-nama kolom dalam DataGrid. Dalam pembahasan sebelumnya nama-nama kolom terjadi secara otomatis berdasarkan data yang diquery dari database. Penentuan nama kolom dalam DataGrid lebih memastikan pengendalian jumlah kolom, serta lebih leluasa memodifikasi berbagai property maupun event di kolom tersebut.

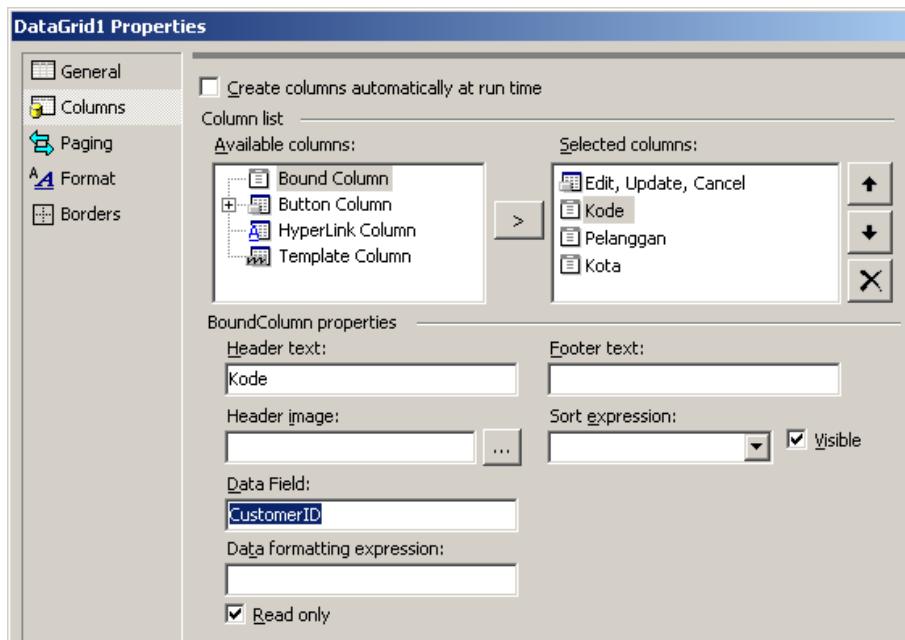
Klik kanan DataGrid, selanjutnya masuk ke Property Builder > tab Column. Hilangkan check di "Create Columns Automatically at Run Time". Kemudian pilih Edit, Update, Cancel column untuk menampilkan hyperlink pada DataGrid.



Di bagian Button type dipilih LinkButton sehingga yang ditampilkan adalah link Edit, Update, dan Cancel.

Kemudian tambahkan pula 3 buah Bound column untuk membatasi jumlah kolom DataGrid hanya 3 buah saja. Isikan properti kolom sebagai berikut:

Header Text	Data Field	Read Only
Kode	CustomerID	YES
Pelanggan	CompanyName	NO
Kota	City	NO



Langkah berikutnya adalah menambahkan kode di method BindGrid agar DataKeyField di DataGrid dikenali dan berhubungan dengan field dari database.

```
Sub BindGrid()
    'kaitkan data di dataset ke datagrid
    DataGrid1.DataSource = dsCustomers
    DataGrid1.DataKeyField = "CustomerID"
    DataGrid1.DataBind()
End Sub
```

Apabila halaman customer.aspx yang telah dimodifikasi tersebut dijalankan maka tampak sebagai berikut:

	Kode	Pelanggan	Kota
<a href="#">Edit</a>	ALFKI	Alfreds Futterkiste	Berlin
<a href="#">Edit</a>	ANATR	Ana Trujillo Emparedados y helados	México D.F.
<a href="#">Edit</a>	ANTON	Antonio Moreno Taquería	México D.F.
<a href="#">Edit</a>	AROUT	Around the Horn	London
<a href="#">Edit</a>	BERGS	Berglunds snabbköp	Luleå
<a href="#">Edit</a>	BLAUS	Blauer See Delikatessen	Mannheim
<a href="#">Edit</a>	BLONP	Blondesddsl père et fils	Strasbourg
<a href="#">Edit</a>	BOLID	Bólido Comidas preparadas	Madrid
<a href="#">Edit</a>	BONAP	Bon app'	Marseille
<a href="#">Edit</a>	BOTTM	Bottom-Dollar Markets	Tsawassen
1 2 3 4 5 6 7 8 9 10			

Tetapi jika link Edit diklik tidak terjadi apa-apa, karena masih ada beberapa langkah lagi yang harus ditambahkan. Anda harus menambahkan kode handler di event EditCommand agar DataGrid dapat melakukan operasi Edit dan Cancel seperti yang diharapkan.

```
Private Sub DataGrid1>EditCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventEventArgs) Handles DataGrid1.EditCommand
    DataGrid1.EditItemIndex = e.Item.ItemIndex
    BindGrid()
End Sub
```

Sekarang apabila halaman tersebut dirun dan diklik link Edit maka DataGrid menampilkan baris data yang siap untuk diedit.

	Kode	Pelanggan	Kota
<a href="#">Edit</a>	ALFKI	Alfreds Futterkiste	Berlin
<a href="#">Update Cancel</a>	ANATR	Ana Trujillo Emparedados	México D.F.
<a href="#">Edit</a>	ANTON	Antonio Moreno Taquería	México D.F.

Agar DataGrid merespon jika link Cancel diklik, maka ditambahkan kode berikut di event CancelCommand:

```
Private Sub DataGrid1>CancelCommand(ByVal source As Object, ByVal e As
System.Web.UI.WebControls.DataGridCommandEventEventArgs) Handles DataGrid1.CancelCommand
    DataGrid1.EditItemIndex = -1
    BindGrid()
End Sub
```

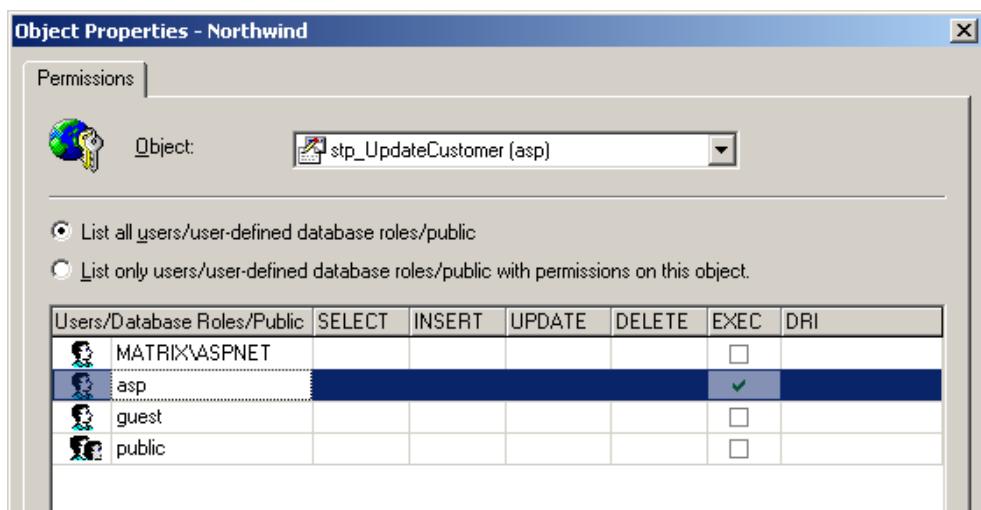
Sekarang jika link Cancel diklik maka baris yang diedit akan kembali ke posisi semula.

Langkah terakhir adalah menambahkan kode untuk link Update. Bagian ini merupakan segmen terpanjang dari kode-kode sebelumnya, karena harus ditentukan parameter dan dibuat syntax T-SQL untuk mengupdate data ke database.

Agar proses update ke database dapat berjalan lebih cepat dan efisien, maka dibuat stored procedure untuk mengupdate data tersebut. Jalankan perintah berikut dari Query Analyzer:

```
CREATE PROCEDURE stp_UpdateCustomer
    @CustomerID char(5),
    @CompanyName nvarchar(40),
    @City nvarchar(15)
AS
UPDATE Customers
SET CompanyName = @CompanyName,
    City = @City
WHERE CustomerID = @CustomerID
```

Kemudian jangan lupa untuk memberikan permission yang sesuai bagi user terhadap stored procedure tersebut.



Untuk mengeksekusi stp\_UpdateCustomer maka digunakan perintah-perintah ADO.NET di event UpdateCommand dari DataGrid. Perhatikan pula bahwa sebelum perintah ADO.NET digunakan maka terlebih dahulu mengambil kontrol texbox dari DataGrid yang terdapat di dalam cell.

Stored procedure dieksekusi menggunakan obyek Command, sehingga perlu dideklarasikan obyek tersebut di bagian general declaration Class WebForm.

```
Dim cmd As New SqlCommand
```

Kemudian masuk di event DataGrid1\_UpdateCommand dan mendeklarasikan variabel untuk mengambil nilai cell yang terdapat di DataGrid1.

```
Dim txtPelanggan As TextBox
Dim txtKota As TextBox
txtPelanggan = CType(e.Item.Cells(2).Controls(0), TextBox)
txtKota = CType(e.Item.Cells(3).Controls(0), TextBox)
```

Langkah penting berikutnya adalah mendeklarasikan parameter serta memasukkan nilai dari parameter tersebut. Nilai parameter diambil dari cell-cell di DataGrid1.

```
'assign parameter
Dim prCompany As SqlParameter = New SqlParameter("@CompanyName", SqlDbType.NVarChar, 40)
cmd.Parameters.Add(prCompany)
prCompany.Value = txtPelanggan.Text

Dim prCity As SqlParameter = New SqlParameter("@City", SqlDbType.NVarChar, 15)
cmd.Parameters.Add(prCity)
prCity.Value = txtKota.Text

Dim prCustomerID As SqlParameter = New SqlParameter("@CustomerID", SqlDbType.Char, 5)
cmd.Parameters.Add(prCustomerID)
prCustomerID.Value = DataGrid1.DataKeys(e.Item.ItemIndex)
```

Setelah parameter dimasukkan nilainya maka dibuka koneksi database dan dilancarkan perintah menggunakan obyek Command.

```
'eksekusi perintah
cn.Open()
With cmd
    .CommandText = "stp_UpdateCustomer"
    . CommandType = CommandType.StoredProcedure
    .Connection = cn
    .ExecuteNonQuery()
End With
```

Langkah selanjutnya adalah mengembalikan posisi DataGrid ke mode non edit, dan mengaitkan data dari database sehingga tampil di DataGrid. Jangan lupa pula untuk menutup koneksi yang sudah tidak digunakan.

```
DataGrid1.EditRowIndex = -1
BindGrid()
cn.Close()
```

Coba Anda jalankan kembali aplikasi tersebut, dan klik link Edit pada salah satu baris. Rubahlah data yang ada dan simpan perubahan tersebut dengan mengklik “Update”. Data tersimpan di database tetapi hasil perubahan belum dapat dilihat di DataGrid. Mengapa? Karena kita menggunakan DataSet sebagai data source, dan tidak melakukan refresh DataSet sebelum data ditampilkan. Dengan demikian data yang ditampilkan adalah DataSet lama, sedangkan perubahan data sudah disimpan langsung di database. Cobalah merefresh kembali browser agar data hasil perubahan ditampilkan.

Jika ingin menampilkan data tanpa merefresh browser maka dapat dilakukan dengan menarik DataSet dari database sebelum data tersebut ditampilkan kembali. Resikonya adalah proses penampilan data dapat menjadi lebih lama. Tambahkan kode tersebut sebelum method BindGrid() pada baris akhir, sehingga menjadi:

```
DataGrid1.EditRowIndex = -1

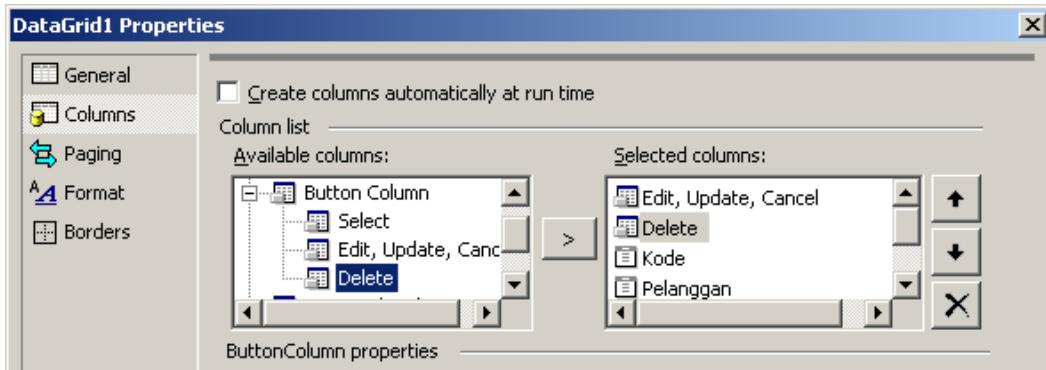
'clear dataset dan ambil dataset baru
dsCustomers.Clear()
adpSQL.Fill(dsCustomers)

'tampilkan di datagrid
BindGrid()
cn.Close()
```

Jalankan kembali aplikasi dan lakukan perubahan data dan simpan, maka data hasil penyimpanan akan langsung ditampilkan di DataGrid.

## 2.4 Menambahkan Fungsi Delete

Selain Edit dan Update dapat pula ditambahkan link “Delete” untuk menghapus data. Seperti cara sebelumnya, Anda tinggal masuk ke Property Builder dari DataGrid dan menambahkan link Delete pada tab Column.



		Kode	Pelanggan	Kota
Edit	Delete	Databound	Databound	Databound
Edit	Delete	Databound	Databound	Databound
Edit	Delete	Databound	Databound	Databound
Edit	Delete	Databound	Databound	Databound

Kita akan membuat operasi penghapusan data menggunakan stored procedure di SQL Server sehingga perlu dibuat dahulu procedure tersebut. Ketik dan jalankan kode berikut dari Query Analyzer terhadap database NorthWind.

```
CREATE PROC stp_hapusKaryawan
    (@CustomerID nchar(5))
AS
DELETE Customers
WHERE CustomerID = @CustomerID
```

Sebagaimana telah dilakukan sebelumnya, jangan lupa untuk memberikan hak eksekusi stored procedure stp\_hapusKaryawan kepada user terkait di SQL Server.

Selanjutnya digunakan ADO.NET untuk mengeksekusi perintah tersebut. Masukkan kode berikut di dalam event DataGrid1\_DeleteCommand.

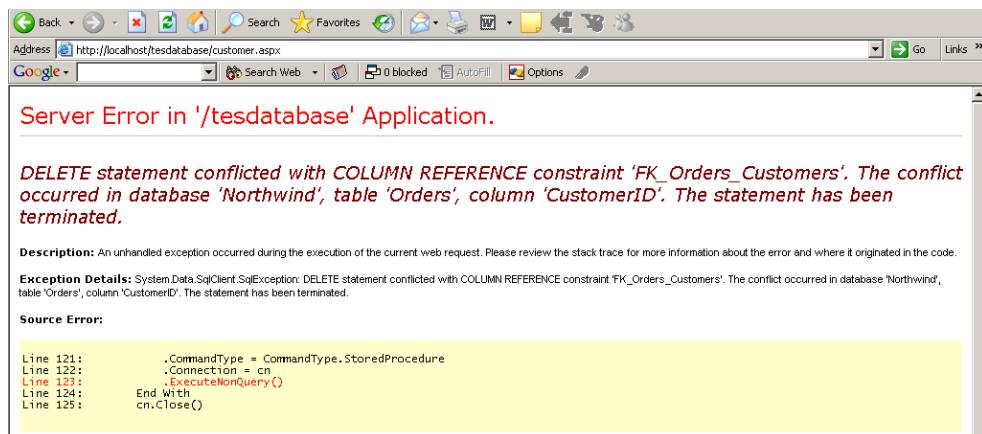
```
Private Sub DataGrid1_DeleteCommand(ByVal source As Object, ByVal e As System.Web.UI.WebControls.DataGridCommandEventEventArgs) Handles DataGrid1.DeleteCommand
    Dim prCustomerID As SqlParameter = New SqlParameter("@CustomerID",
    SqlDbType.NChar, 8)
    cmd.Parameters.Add(prCustomerID)
    prCustomerID.Value = DataGrid1.DataKeys(e.Item.ItemIndex)

    cn.Open()
    With cmd
        .CommandText = "stp_hapusKaryawan"
        . CommandType = CommandType.StoredProcedure
        . Connection = cn
        . ExecuteNonQuery()
    End With
    cn.Close()

    dsCustomers.Clear()
    adpSQL.Fill(dsCustomers)
    BindGrid()
End Sub
```

Jalankan aplikasi dan klik link Delete pada salah satu baris, maka data di baris tersebut akan terhapus.

Kemungkinan bisa saja menerima pesan error berikut:



Penyebabnya adalah tabel Customers memiliki hubungan referential integrity dengan tabel Orders. Anda dapat menghilangkan hubungan tersebut, sehingga data Customer dapat dihapus meskipun terdapat data di tabel Orders. Alternatif lainnya adalah dengan mengeset Cascading delete, sehingga data di tabel Orders dan Order Details akan ikut terhapus. Cara melakukan 2 hal ini dapat Anda baca di “Pengantar Administrasi SQL Server 2000” di IlmuKomputer.Com.

### 3. Membuat Guestbook Sederhana

Bagian yang juga paling populer dari aplikasi database adalah form untuk mengentri dan menampilkan data di database. Aplikasi yang sering ditemui antara lain formulir registrasi membership, pengisian guestbook, ataupun mengisi pesanan order.

Dalam tutorial ini dipraktekkan teknik entry data dan menampilkan hasil entry tersebut dalam bentuk aplikasi guestbook atau buku tamu. Aplikasi ini sangat banyak dijumpai dan tentu saja sudah tidak asing lagi bagi para pengguna internet.

Dalam tutorial ini hanya akan ditampilkan penggunaan guestbook dari sisi end user yang mengisi dan melihat hasil pengisian guestbook. Sedangkan berbagai interface untuk admin tidak dibahas, silakan Anda sendiri mengembangkan aplikasi tersebut ☺.

#### 3.1 Membuat Database

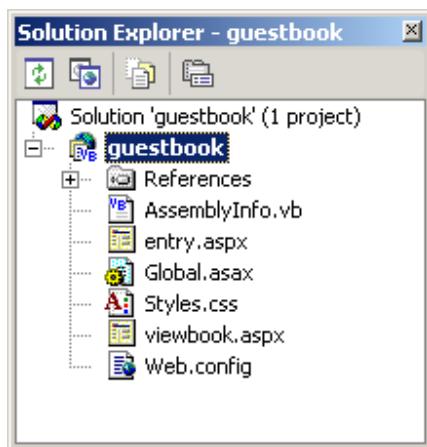
Buatlah database bernama BukuTamu di SQL Server, dan buatlah 1 tabel bernama tblEntryData dengan struktur tabel sebagai berikut:

	Column Name	Data Type	Length	Allow Nulls
1	entryid	int	4	
2	nama	nvarchar	25	
3	email	nvarchar	30	
4	url	nvarchar	30	✓
5	komentar	nvarchar	200	
6	tgl	smalldatetime	4	

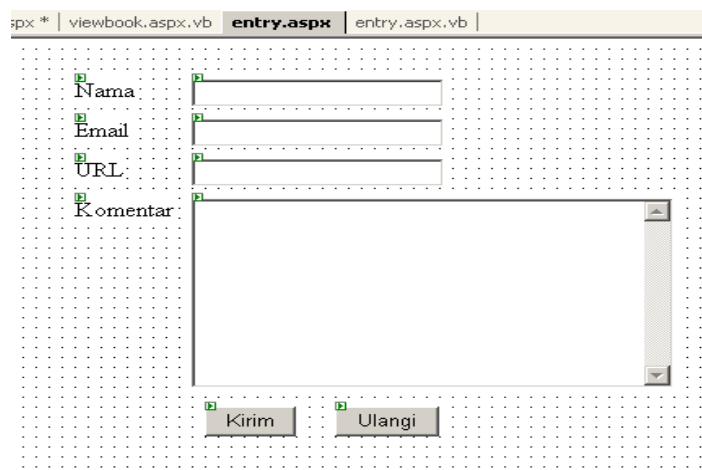
Field entryID dijadikan Primary Key dengan tipe data Integer dan autonumber. Field lainnya disetting Allow Nulls = False, sehingga harus selalu diisi.

#### 3.2 Membuat Project

Aplikasi ini terdiri dari 2 form yaitu entry.aspx sebagai halaman entry data dan viewbook.aspx untuk menampilkan entry komentar pada buku tamu. Buatlah project baru dengan nama guestbook dan sekaligus membuat 2 form tersebut.



Kemudian letakkan beberapa kontrol textbox, label, dan button pada halaman entry.aspx dengan tata letak seperti berikut:



Nilai properti dari tiap obyek tersebut adalah sebagai berikut:

Obyek	Properti	Nilai Properti
TextBox1	ID	txtnama
TextBox2	ID	txtEmail
TextBox3	ID	txtURL
TextBox4	ID	txtKomentar
Label1	Text	Nama
Label2	Text	Email
Label3	Text	URL
Label4	Text	Komentar
Button1	Text	Kirim
Button2	Text	Ulangi

### 3.3 Menambahkan validasi data

Agar input data dari user selalu terjamin kebenarannya, maka ditambahkan fasilitas untuk mengecek data yang dimasukkan di form. Fungsi validation control tersebut adalah mengecek validitas data yang dientri sehingga kemungkinan terdapat data sampah atau tidak sesuai dengan setup database bisa dihindari.

Ketika user menekan tombol “Kirim” maka validation control langsung mengecek entri apakah memenuhi syarat atau tidak.

The screenshot shows a form with four text input fields and two buttons. The fields are labeled 'Nama', 'Email', 'URL', and 'Komentar'. The 'Nama' field has an error message 'Nama harus diisi'. The 'Email' field has an error message 'Format Email tidak benar'. The 'URL' field has an error message 'URL tidak valid'. The 'Komentar' field has an error message 'Komentar harus diisi'. Below the fields are two buttons: 'Kirim' and 'Ulangi'.

Pengaturan properti untuk tiap validator tersebut adalah sebagai berikut:

Obyek	Properti	Nilai Properti
RequiredFieldValidator1	ErrorMessage	Nama harus diisi
	ControlToValidate	txtNama
RegularExpressionValidator1	ErrorMessage	Format Email tidak benar
	ValidationExpression	\w+([-.\w+]*)@\w+([-.\w+]*)\.\w+([-.\w+]*)
	ControlToValidate	txtEmail
RegularExpressionValidator2	ErrorMessage	URL tidak valid
	ValidationExpression	http://([\\w-]+\\.\\w-)+([\\w-]+(\\/[\\w- ./?%&=]*))?
	ControlToValidate	TxtURL
RequiredFieldValidator2	ErrorMessage	Komentar harus diisi
	ControlToValidate	txtKomentar

### 3.4 Membuat Stored Procedure

Stored procedure dibuat untuk menambahkan data hasil entri user di buku tamu. Kode berikut memasukkan data di 4 field yang terdapat di halaman entry.aspx. Ketikkan kode berikut di Query Analyzer:

```
CREATE PROCEDURE stp_isibukutamu
    (@nama      [nvarchar] (30),
     @email     [nvarchar] (30),
     @url       [nvarchar] (30),
     @komentar   [nvarchar] (200),
     @tgl       [smalldatetime])

AS INSERT INTO [bukutamu].[dbo].[tblEntryBook]
    ( [nama],
     [email],
     [url],
     [komentar],
     [tgl])

VALUES
    (@nama,
     @email,
     @url,
     @komentar,
     @tgl)
```

Sebgaimana telah dilakukan sebelumnya, Anda harus memberikan hak akses kepada user untuk mengeksekusi stored procedure tersebut. Langkah selanjutnya adalah mengeksekusi stored procedure tersebut dari halaman entry.aspx.

### 3.5 Eksekusi Stored Procedure dan Memasukkan Data

Kode untuk menambahkan komentar user ke database disisipkan di dalam event Button1\_Click sehingga dieksekusi ketika user menekan tombol "Kirim". Untuk keperluan ini kita memerlukan obyek Command yang digunakan untuk mengeksekusi stored procedure dan mendefinisikan parameter dari 4 field di form entry.aspx.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    'Deklarasi variabel
    Dim sNama As String = txtNama.Text
    Dim sEmail As String = txtEmail.Text
    Dim sURL As String = txtURL.Text
    Dim sKomentar As String = txtKomentar.Text
    Dim dTgl As DateTime = DateAndTime.Now.ToString

    'definisikan connection string
    Dim strCn As String = "server=matrix; user=asp; password=asp;database=bukutamu"
    Dim Cn As SqlConnection = New SqlConnection(strCn)

    'deklarasi Command dan memasukkan parameter
    Dim cmd As New SqlCommand
    Dim prNama As SqlParameter = New SqlParameter("@Nama", SqlDbType.NVarChar, 30)
    cmd.Parameters.Add(prNama)
    prNama.Value = sNama

    Dim prEmail As SqlParameter = New SqlParameter("@email", SqlDbType.NVarChar, 30)
    cmd.Parameters.Add(prEmail)
    prEmail.Value = sEmail
```

```
Dim prURL As SqlParameter = New SqlParameter("@URL", SqlDbType.NVarChar, 30)
cmd.Parameters.Add(prURL)
prURL.Value = sURL

Dim prKoment As SqlParameter = New SqlParameter("@komentar", _
    SqlDbType.NVarChar, 200)
cmd.Parameters.Add(prKomentar)
prKomentar.Value = sKomentar

Dim prTgl As SqlParameter = New SqlParameter("@tgl", SqlDbType.SmallDateTime)
cmd.Parameters.Add(prTgl)
prTgl.Value = dTgl

'buka koneksi
Cn.Open()
'masukkan parameter command (nama stored procedure dan koneksi)
With cmd
    .CommandText = "stp_isibukutamu"
    . CommandType = CommandType.StoredProcedure
    .Connection = Cn
    .ExecuteNonQuery()  'eksekusi stored procedure
End With
Cn.Close() 'tutup koneksi

'menuju halaman viewbook.aspx untuk menampilkan hasil entry
Response.Redirect("viewbook.aspx")
End Sub
```

### 3.6 Mengulangi Pengisian

Kadangkala user berubah pikiran dan ingin mengulang pengisian data sebelum dikirim. Untuk keperluan ini dibuat tombol “Ulangi” yang mengosongkan keempat field tersebut dan siap diisi ulang. Tuliskan kode berikut di event Click Button2:

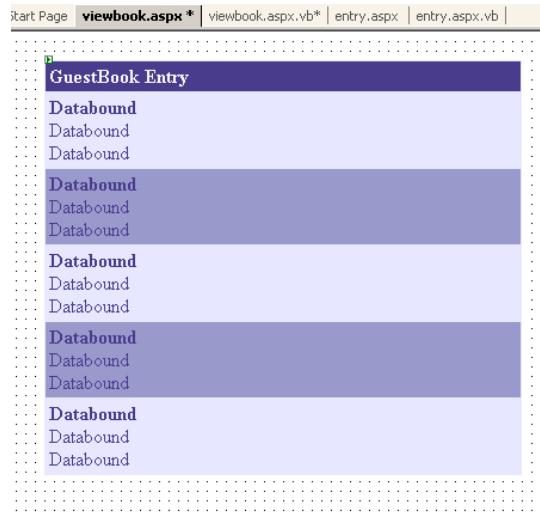
```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    txtNama.Text = ""
    txtEmail.Text = ""
    txtURL.Text = ""
    txtKomentar.Text = ""
End Sub
```

### 3.7 Menampilkan Hasil Entri Data

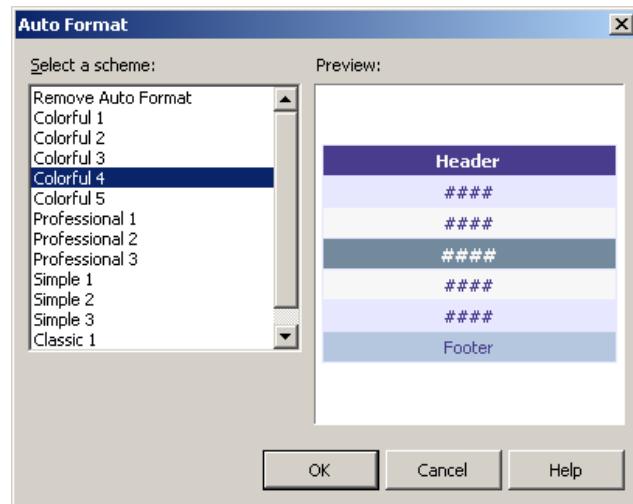
Setelah user menekan tombol “Kirim” maka dipanggil halaman viewbook.aspx yang menampilkan tabel komentar pengunjung di buku tamu. Kita menggunakan DataList untuk menampilkan komentar-komentar tersebut.

Agar akses data ke database berlangsung lebih cepat maka digunakan DataReader dalam ADO.NET dan tidak menggunakan DataSet sebagaimana pada latihan Datagrid sebelumnya. DataReader sangat sesuai untuk menampilkan data secara cepat dan tidak memerlukan editing di sisi klien.

Halaman viewbook.aspx terdiri dari 1 buah DataList sebagai berikut:



Anda dapat merubah tampilan warna DataList menggunakan Auto Format dan memilih berbagai scheme yang tersedia.



Untuk merubah lebar Datalist, pindahkan ke mode HTML and setup Width = 400px.

```
<body MS_POSITIONING="GridLayout">
<form id="Form1" method="post" runat="server">
    <asp:alist id="DataList1" style="Z-INDEX: 101; LEFT: 40px; POSITION: absolute; TOP: 10px; width: 400px; height: 150px; font-size: 10pt; font-family: Verdana, Arial, Helvetica, sans-serif; border-width: 1px; border-color: #E7E7FF; border-style: none; background-color: transparent;" HorizontalAlign="Center" Width="400px" BorderWidth="1px" GridLines="1" BackColor="Transparent" BorderStyle="None" BorderColor="#E7E7FF">
```

Kode untuk menampilkan data di halaman viewbook.aspx ditempatkan di event Page\_Load dengan cara mengklik ganda ruang kosong halaman tersebut dan berpindah ke halaman code behind viewbook.aspx.vb.

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    'deklarasi koneksi dan SQL
    Dim strCn As String = "server=matrix; user=asp; password=asp;database=bukutamu"
    Dim strSQL As String = "Select Nama, Email, URL, Komentar, Tgl FROM
                           tblEntryBook ORDER BY EntryID DESC"
    Dim Cn As SqlConnection = New SqlConnection(strCn)
    'deklarasi Command
    Dim cmd As SqlCommand = New SqlCommand(strSQL, Cn)

    'Buka koneksi
    Cn.Open()
    'Kaitkan datareader ke Datalist
    DataList1.DataSource = cmd.ExecuteReader
    DataList1.DataBind()

    'tutup koneksi
    Cn.Close()
End Sub
```

Apabila menggunakan DataList ternyata kode diatas belum cukup untuk menampilkan data. Anda harus menambahkan sedikit kode HTML di DataList. Buka kembali viewbook.aspx dalam mode HTML dan sisipkan kode untuk menampilkan isi tabel tblEntryBook.

Kode tersebut merupakan sebuah template yang disisipkan diantara tag <ItemTemplate> </ItemTemplate>. Tag ini merupakan template untuk menampilkan detil data pada DataList. Selain tag ini, DataList juga memiliki berbagai template yang tidak dibahas dalam tulisan ini.

```
<ItemTemplate>
<b>
<%# DataBinder.Eval( Container, "DataItem.Nama" )%>
</b>
<br />
<%# DataBinder.Eval( Container, "DataItem.Email" )%>
<br />
<%# DataBinder.Eval( Container, "DataItem.URL" )%>
<br />
<%# DataBinder.Eval( Container, "DataItem.Komentar" )%>
</ItemTemplate>
```

Dalam kode diatas terlihat ditampilkan 4 field: Nama, Email, URL, dan Komentar dan setiap field dipisahkan dengan tag <br> yang merupakan spasi untuk berpindah baris.

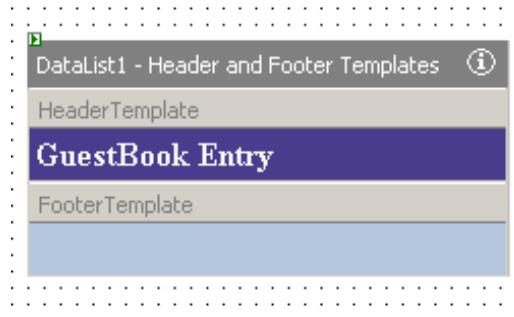
Tag tersebut disisipkan di dalam <asp:alist> <</asp:alist> yang merupakan kode server side dari web control DataList.

```
<body MS_POSITIONING="GridLayout">
    <form id="Form1" method="post" runat="server">
        <asp:datalist id="DataList1" style="Z-INDEX: 101; LEFT: 40px; POSITION: ab
            HorizontalAlign="Center" Width="400px" BorderWidth="1px" GridLines="Hc
            BackColor="Transparent" BorderStyle="None" BorderColor="#E7E7FF">
            <SelectedItemStyle Font-Bold="True" ForeColor="#F7F7F7" BackColor="#73
            <HeaderTemplate>
                GuestBook Entry
            </HeaderTemplate>
            <AlternatingItemStyle BackColor="#9999CC"></AlternatingItemStyle>
            <ItemStyle ForeColor="#4A3C8C" BackColor="#E7E7FF"></ItemStyle>
            <ItemTemplate>
                <b>
                    <%# DataBinder.Eval( Container, "DataItem.Nama" )%>
                </b>
                <br />
                <%# DataBinder.Eval( Container, "DataItem.Email" )%>
                <br />
                <%# DataBinder.Eval( Container, "DataItem.URL" )%>
                <br />
                <%# DataBinder.Eval( Container, "DataItem.Komentar" )%>
            </ItemTemplate>
            <FooterStyle ForeColor="#4A3C8C" BackColor="#B5C7DE"></FooterStyle>
            <HeaderStyle Font-Bold="True" ForeColor="#F7F7F7" BackColor="#4A3C8C">
        </asp:datalist></form>
    </body>
```

Selain mengedit HTML secara langsung, Anda juga dapat mengedit template DataList dengan mengklik kanan kontrol tersebut dan memilih template yang akan diedit. Berikut contoh untuk menambahkan judul DataList di bagian Header.



Kemudian tuliskan GuestBook Entry di bagian header tersebut:



### 3.8 Mencoba GuestBook

Tibalah saatnya Anda menguji buku tamu tersebut. Jadikan halaman entry.aspx sebagai startup page dan jalankan aplikasi. Anda dapat mencoba mengosongkan data dan menekan tombol “Kirim” untuk menguji bahwa validation control telah bekerja sempurna.

alhost/guestbook/entry.aspx

Nama: Choirul

Email: noirul@ilmukomputer.com

URL: tes URL tidak valid

Komentar:

Komentar harus diisi

Jika isian sudah benar dan tombol “Kirim” ditekan maka halaman viewbook.aspx menampilkan daftar komentar yang masuk di buku tamu.

GuestBook Entry

Choirul  
choirul@ilmukomputer.com  
http://testing.com  
testing belajar membuat buku tamu. sudah bisa belom?  
wewq;ew  
jlkewrewrew

## Penutup

Masih banyak sekali topik-topik pemrograman database dalam ASP.NET yang perlu dipelajari untuk lebih mendalami pemrograman .NET. Tulisan ini telah memberikan pengantar, silakan Anda mengembangkan lebih lanjut dengan membaca berbagai tutorial dan buku. Penulis akan membahas berbagai topik database dan ASP.NET pada berbagai kesempatan lain.

Berikut ini adalah berbagai situs internet dimana Anda dapat mempelajari lebih jauh:

<http://msdn.microsoft.com/>

<http://asp.net/>

<http://4guysfromrolla.com/>

<http://codeproject.com/>

-----0OO0-----

Apabila Anda memiliki pertanyaan, silakan posting di:

<http://forum.ilmukomputer.com>

atau gabung di mailing list dotnet dengan mengirim email kosong ke:

[dotnet-subscribe@netindonesia.net](mailto:dotnet-subscribe@netindonesia.net)