

Part 10 – Generate Excel Report

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://www.suryodesign.asia>

www.suryodesign.wordpress.com

Lisensi Dokumen:

Copyright © 2003-2007 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

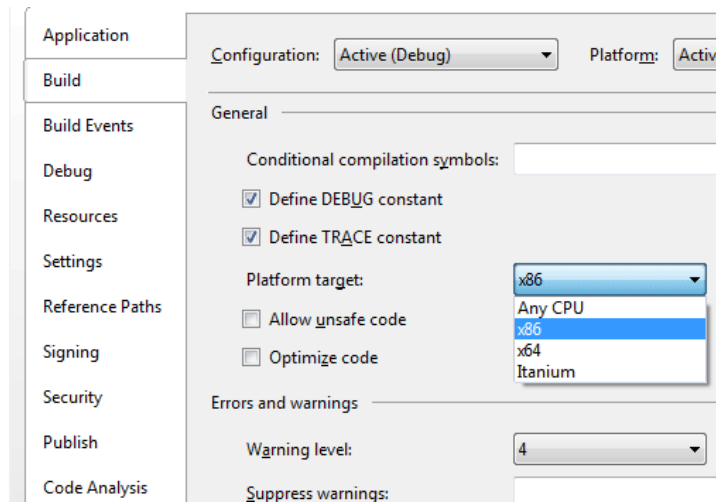
Pada pertemuan ke 10 ini saya akan membahas bagaimana membuat report ke dalam excel , banyak metode yang dapat kita gunakan untuk men-generate suatu repot berdasarkan data yang ada , kali ini saya akan membahas cara yang simple dan mudah digunakan dan dimengerti oleh para pembaca , kita akan membuat sebuah report yang hanya berisikan text dan nama column yang simple berdasarkan data yang ditampilkan, next time bila saya ada waktu luang mungkin saya akan mencoba membuat suatu report yang lebih mendalam lengkap dan sedikit extreme bagi para pembaca.untuk dapat dijadikan bahan referensi dalam membuat report yang banyak digunakan pada perusahaan-perusahaan saat ini untuk membuat laporan yang lebih kompleks.

Penulis mungkin selama beberapa minggu ke depan pada bulan Maret 2009 ini akan vakum sementara karena ada beberapa keperluan , dan penulis meminta maaf apabila ada mail / message yang tidak bisa saya balas selama bulan maret ini. Tetapi Penulis akan tetap Exist untuk terus menulis artikel pada IlmuKomputer ini pada bulan-bulan selanjutnya. Untuk artikel sendiri dapat di download di www.suryodesign.asia/freedownload

Isi

Dalam pembuatan ini ada beberapa hal yang harus di perhatikan seperti dalam penggunaan perbedaan Araay dengan ArrayList , pengetahuan OleDb Konsep dan Excel Document , dan penggunaan komponen seperti linkLabel , groupBox dan linkLabel. Hingga dalam factor Build Application yang berpengaruh terhadap PlatForm target maupun penggunaan Try Catch Finally untuk menghandle kesalahan yang terjadi.

Pertama kita akan membahas tentang PlatformTarget , maksudna disini adalah dalam menggunakan komponen OleDb bagi yang memakai Operating Vista 64 Bit dan Office 2007 mungkin akan menemukan kesalahan yang mungkin sama dihadapi oleh penulis saat itu. Untuk lebih jelasnya mungkin dapat di lihat pada link berikut ini agar pembaca dapat lebih mengerti : <http://suryodesign.wordpress.com/2009/03/05/oledbprovidernotregistered/> , tapi saya akan menjelaskan secara singkat tentang masalah ini, mungkin kita akan menemukan pesan Error seperti *"The Microsoft.Jet.OLEDB.4.0' provider is not registered on the local machine"* . , mungkin bagi yang tidak mendapatkan masalah ini dapat menghiraukan factor Operating System satu ini , bila hal ini terjadi maka yang mesti kita lakukan adalah mengganti Properties applicaton pada menu BuilTab dengan mengganti PlatFormTarget dari Any CPU menjadi X86 , untuk lebih jelasna dapat di lihat pada printscreen berikut :



Setelah masalah ini selesai , atau mungkin bagi pembaca yang tidak menemukan masalah ini

dapat langsung lanjut ke tahap selanjutnya yaitu perbedaan Array dan ArrayList.

Array :

Array biasa saat di definisikan memiliki nilai atau batas array tersebut ataupun dimensi awal yang mesti dimiliki oleh array tersebut seperti berikut :

```
// Buat dan inisialisasi sebagai integer array dan object Array
int[] myIntArray = new int[5] { 1, 2, 3, 4, 5 };
Object[] myObjArray = new Object[5] { 26, 27, 28, 29, 30 };

// Cetak nilai dari kedua array di atas
Console.WriteLine( "Initially," );
Console.Write( "integer array:" );
PrintValues( myIntArray );
Console.Write( "Object array: " );
PrintValues( myObjArray );

// Copy nilai dari 2 element array pertama ke dalam object Array
Array.Copy( myIntArray, myObjArray, 2 );
```

ArrayList :

Kemudian untuk arrayList sendiri memiliki perbedaan dimana kita tidak perlu untuk mendefinisikan batas atas atau pun bawah dan tidak perlu setting di posisi Array beberapa kita mesti insert , dan arrayList dapat menampung berbagai macam tipe dan object yang ada ,atau biasa nya orang menyebut arrayList dengan sebutan Dynamic Array , dan untuk menggunakan arrayList jangan lupa untuk menambahkan syntax berikut :

```
using System.Collections;
```

kemudian untuk membuat arrayList itu sendiri anda dapat mendefinisikan seperti berikut ini :

```
ArrayList myArrayList = new ArrayList();
myArrayList.Add(56);
myArrayList.Add("String");
myArrayList.Add(new Form());
```

dan untuk membaca object yang ada dapat menggunakan foreach untuk membaca setiap elemen yang merupakan bagian dari arrayList itu sendiri seperti berikut :

```
foreach (object objInsert in myArrayList)
{
```

}

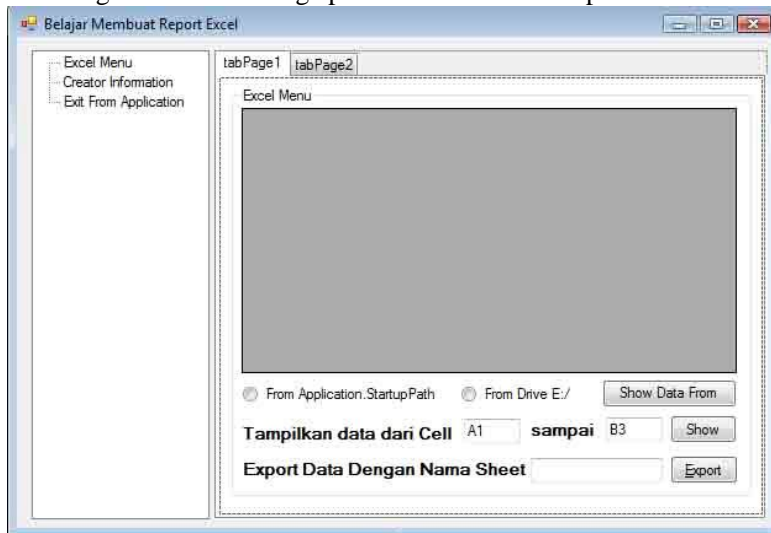
```
        MessageBox.Show(Convert.ToString(myArrayList));  
    }
```

Try Catch Finally

Try Catch Finally kurang lebih Saya jelaskan seperti berikut , Try merupakan sisi dimana kita mencoba / meletakkan coding yang telah kita buat , kemudian Catch untuk menangkap kesalahan yang ada , kemudian Finally akan dijalankan apabila dalam Try tidak ditemukan kesalahan , berikut adalah contoh Simple yang dapat kita lihat dari MSDN Microsoft

```
try  
{  
    result = top / bottom;  
}  
catch (System.Exception ex)  
{  
    System.Console.WriteLine("{0} exception caught here.", ex.GetType().ToString());  
    System.Console.WriteLine(ex.Message);  
}  
finally  
{  
    System.Console.WriteLine("Clean-up code executes here...");  
}  
System.Console.WriteLine("Program execution continues here...");
```

Sekarang mari kita rancang aplikasi User Inteface seperti berikut ini :



Komponen yang digunakan :

- Button
- DataGridView
- GroupBox
- Label
- LinkLabel
- Panel
- RadioButton
- SplitContainer
- TextBox
- TreeView

Dalam konsep design , Gunakan SplitContainer untuk membagi menjadi 2 kolom , kolom kiri untuk navigasi dan kolom kanan sebagai ViewData, dan Group semua komponen di kanan menggunakan GroupBox agar lebih rapih dan terorganisi.

Setelah Anda Mendesign mari kita pertama – tama untuk membaca File excel kemudian menampilkannya pada DataGridView berdasarkan Radio Button :

```
//buat string koneksi dan simpan namafile Excel yang akan di load
string connectionString = null;
string namaFile = "barang.xls";
//setting koneksi berdasarkan drive atau application.StartupPath
if (radioButton1.Checked)
{
    connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=E:\barang.xls;Extended Properties=""Excel 8.0;HDR=YES;""";
}
else
{
    connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" +
Application.StartupPath + "\\ " + namaFile + ";Extended Properties='Excel
8.0;HDR=YES' ";
}
//buat koneksi dan eksekusi
OleDbConnection cn = new OleDbConnection(connectionString);
OleDbDataAdapter cmd = new OleDbDataAdapter("select * from [Sheet1$]",
cn);
//buat dataset ds3 kemudian fill dataset dengan query dari excel pada Sheet1
```

5

```
DataSet ds3 = new DataSet();  
cmd.Fill(ds3, "abc");  
//tampilkan query ke dalam datagrid kemudian dispose dataset  
dataGridView1.DataSource = ds3;  
dataGridView1.DataMember = "abc";  
ds3.Dispose();
```

Cara yang telah kita buat adalah menampilkan semua data yang berasal dari Excel secara keseluruhan, tapi bila kita ingin menampilkan berdasarkan pilihan kolom / cell yang kita miliki kita dapat memvalidasi seperti berikut :

```
//gunakan try dan catch untuk meng-handle kesalahan yang ada  
try  
{  
    //cek validasi txtCell1 dan txtCell2 tidak boleh kosong  
    if (txtCell1.Text == "" || txtCell2.Text == "")  
    {  
        //exit dari aplikasi  
        return;  
    }  
  
    //buat koneksi dan kemudian eksekusi berdasarkan query dan cell inputan  
    string connectionString = @"Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=E:\barang.xls;Extended Properties=""Excel 8.0;HDR=YES;""";  
    OleDbConnection cn = new OleDbConnection(connectionString);  
    OleDbDataAdapter cmd = new OleDbDataAdapter("select * from [Sheet1$" +  
txtCell1.Text + ":" + txtCell2.Text + "]", cn);  
    //buat dataset dan kemudian dataset tersebut di fill berdasarkan query  
    //dan di simpan dalam table DataTable pada dataSetBarang  
    DataSet DataSetBarang = new DataSet();  
    cmd.Fill(DataSetBarang, "DataTable1");  
    //tampilkan data pada datagrid berdasarkan dataset, kemudian dispose  
    dataset  
    dataGridView1.DataSource = DataSetBarang;  
    dataGridView1.DataMember = "DataTable1";  
    DataSetBarang.Dispose();  
}  
catch (Exception salah)  
{  
    //stop proses dan memberitahu kesalahan yang ada  
    throw (salah);  
}
```

Kemudian setelah kita berhasil untuk membaca data , saatnya kita untuk Generate data yang telah kita tampilkan ke dalam Excel Seperti berikut :

```
//cek validasi txtNamaShet
if (txtNamaSheet.Text == "")
{
    //stop / break code
    return;
}
//buat array list dan variabel boolean
ArrayList tampungQueryInsert = new ArrayList();
bool flagColumn = false ;

//buat string penampung untuk CreateColumn dan Insert data
string createColumn = "CREATE TABLE " + txtNamaSheet.Text + "(";
string insertData = "insert into " + txtNamaSheet.Text + " values(";

//lakukan perulangan berdasarkan jumlah Rows
for (int i = 0; i < dataGridView1.Rows.Count; i++)
{
    //lakukan perulangan berdasarkan jumlah Column
    for (int j = 0; j < dataGridView1.ColumnCount; j++)
    {
        //setelah membaca data maka ambil data pada cell untuk nanti diinsert
        insertData += "'" + dataGridView1.Rows[i].Cells[j].Value + "',";
        //bila flagColumn bernilai false maka isi dalam if dijalankan
        if (!flagColumn )
        {
            //tampung Syntax untuk Create Column
            createColumn += Convert.ToString( dataGridView1.Columns[j].Name) + "
            char(255)" + ",";
        }
    }
    //potong huruf terakhir ',' pada String agar valid untuk diinsert
    insertData = insertData.Substring(0, insertData.Length - 1) + ")";
    //tambah ke dalam ArrayList
    tampungQueryInsert.Add(insertData);
    //set Default isi string insertData
    insertData = "insert into " + txtNamaSheet.Text + " values(";
    //bila flagColumn bernilai false maka isi dalam if dijalankan
    if (!flagColumn)
```

```
{
//hapus huruf terakhir ',' pada string CreateColumn agar valid untuk
diinsert
createColumn = createColumn.Substring(0, createColumn.Length - 1) + ")";
}
//set flagColumn bernilai true
flagColumn = true;
}
//buat koneksi
System.Data.OleDb.OleDbConnection connExcel = new
System.Data.OleDb.OleDbConnection(@"Provider=Microsoft.Jet.OLEDB.4.0;D
ata Source=E:\coba.xls;Extended Properties=""Excel 8.0;HDR=YES;""");
// Buka Koneksi
connExcel.Open();
// Buat Command
System.Data.OleDb.OleDbCommand cmdExcel = new
System.Data.OleDb.OleDbCommand();

//set koneksi dan command yang akan digunakan pada command
cmdExcel.Connection = connExcel;
cmdExcel.CommandText = Convert.ToString(createColumn);
//eksekusi command
cmdExcel.ExecuteNonQuery();

//baca setiap data yang ada pada ArrayList
//kemudian data tersebut kita insert
foreach (object objInsert in tampungQueryInsert)
{
cmdExcel.CommandText = Convert.ToString(objInsert);
cmdExcel.ExecuteNonQuery();
}
//tutupKoneksi
connExcel.Close();
```

Setelah menu Excel selesai saatnya kita mengatur Navigasi pada Treeview , double klik TreeView pada left container dan isi dengan baris perintah berikut :

```
//cek Treeview yang di klik
//indeks pada Treeview di mulai dari indeks 0 bukan 1
if (treeView1.SelectedNode.Index == 0)
{
// fanggil function hiddenTabPage , kemudian add tabPage1
```



```
//kemudian select TabPage pertama
hideTabPage();
tabControl1.TabPages.Add(tabPage1);
tabControl1.SelectedIndex = 0;
}
else if (treeView1.SelectedNode.Index == 1)
{
// fanggil function hiddenTabPage , kemudian add tabPage2
//kemudian select TabPage pertama
hideTabPage();
tabControl1.TabPages.Add(tabPage2);
tabControl1.SelectedIndex = 0;
}
else if (treeView1.SelectedNode.Index == 2)
{
//buat messageBox untuk exit dengan option Yes / No untuk exit dari aplikasi
if (MessageBox.Show("Anda Ingin exit dari aplikasi ?", "Exit Menu",
MessageBoxButtons.YesNo) == DialogResult.Yes)
{
//memberi message
MessageBox.Show("Terima Kasih telah membaca artikel ini");
//exit Dari Aplikasi
Environment.Exit(1);
}
}
```

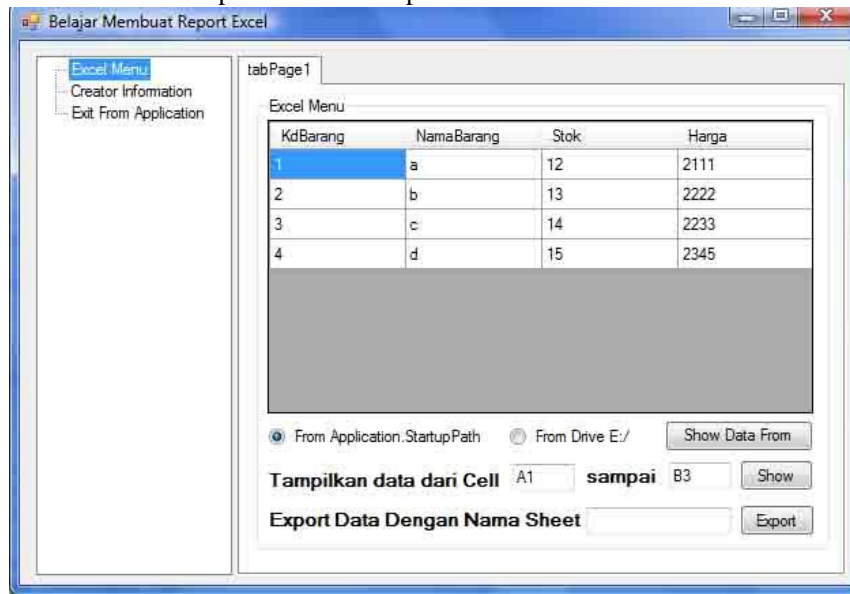
Setelah menu navigasi dan TabPage1 untuk menampilkan data jadi , saatnya kita mengatur LinkLabel pada tabPage2 , berikut tampilan tabPage2 :



Dapat kita lihat disana ada 3 linkLabel , double klik masing-masing link label dan isikan perintah seperti berikut :

```
//membuka Link dengan sumber link dari namaLinkLabel itu sendiri  
System.Diagnostics.Process.Start(linkLabel1.Text);
```

Berikut adalah tampilan akhir dari aplikasi :



Fiuhh... akhirnya setelah melewati berbagai latihan , semoga anda dapat menjadi lebih terbiasa dengan komponen dan gaya penulisan yang ada pada C# dan membuat logika anda terlatih ^^

Penutup

Jangan Pernah menyerah apapun masalah yang anda hadapi , ingat pesan Thomas Edison yang berbunyi “Saya hanya menemukan 100000 cara yang tidak bekerja , dan bukan berarti saya gagal dalam menyelesaikannya” , terus berusaha dan keep improve dan jangan lupa sumbangkan ilmu anda pada komunitas yang anda , kaitkanlah ilmu selalu dengan membaca dan menuangkannya pada komunitas yang ada.

Referensi

- MSDN 2005
- <http://msdn.microsoft.com>
- www.gotdotnet.com
- <http://social.msdn.microsoft.com/>
- www.vbdotnetforum.com
- <http://www.c-sharpcorner.com>
- <http://www.dotnetspider.com>

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 & overclocking computer.

Ym : suryolovetyka

Email : suryodesign@yahoo.co.id

Website : <http://www.suryodesign.asia>

Blog : www.suryodesign.wordpress.com