

C# Part 1 – Pengenalan Logika Basic

M.Suryo Pranoto

survodesign@yahoo.co.id

<http://suryostudio.net>

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

Pertama kali mendengar kata “C#” kita akan bertanya “Bahasa pemrograman apakah ini” ? “Apa Tujuannya kita mempelajarinya” ? “Bagaimana cara mempelajarinya” ? “Apa bedanya dengan bahasa .Net lainnya yg terdapat pada Visual Studio” ?

Saya Akan menjawab pertanyaan ini masing-masing agar dapat membuka pikiran anda lebih luas lagi. C# adalah salah satu bahasa pemrograman intermediate yg digunakan oleh programmer untuk membuat executable programs / program yg dapat di eksekusi. C# dapat memisahkan jarak antara aplikasi yang powerfull tapi membingungkan seperti C++ dan sangat mudah sekali untuk digunakan , C# memiliki extension .CS .

Tujuan kita mempelajari C# adalah untuk membuat suatu program, apa itu program ? wah menjadi pertanyaan lagi ☺ , program adalah suatu executable file atau dengan kata lain adalah file yg dapat di eksekusi dengan double klik suatu icon ,Sebagai contoh adalah Microsoft word yg sering kita gunakan adalah sebuah program, file ini memiliki extension .exe yang merupakan executable files.

Sebelum menjalankan aplikasi yg dibuat dengan C# , kita memerlukan Common Language Runtime(CLR) untuk meng-eksekusi program yg di generate dengan C#. Ini kita dapat dari .Net Framework yg dapat kita download langsung dari situs Microsoft secara free, atau otomatis terinstall ke dalam komputer ketika menginstall Visual Studio sebagai suatu syarat standard pada procedure penginstalan.

Berikut ini beberapa kelebihan C# :

- **Flexible:** C# program dapat di eksekusi di mesin computer sendiri atau di transmikan melalu web dan di eksekusi di computer lainnya
- **Powerful:** C# memiliki sekumpulan perintah yang sama dengan C++ yang kaya akan fitur yang lengkap tetapi dengan gaya bahasa yang lebih diperhalus sehingga memudahkan penggunaanya
- **Easier to use:** C# memodifikasi perintah yang sepenuhnya sama dengan C++ dan

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2007 IlmuKomputer.Com

memberitahu dimana letak kesalahan kita bila ada kesalahan dalam aplikasi , hal ini dapat mengurangi waktu kita dalam mencari error

- **Visually oriented:** The .NET library code yang digunakan oleh C# menyediakan bantuan yang dibutuhkan untuk membuat tampilan yang complicated dengan frames, dropdown , tabbed windows, group button , scroll bar , background image , dan lainnya

- **Secure:** semua bahasa pemrograman yg digunakan untuk kebutuhan internet mesti memiliki security yg benar-benar aman untuk menghindari aksi kejahatan dari pihak lain seperti hacker , C# memiliki segudang fitur untuk menanganinya

Isi

Pertama – tama , mari kita mempelajari struktur dasar dari logika basic pemrograman. kita mesti mengenal penggunaan macam-macam perulangan yang kita gunakan untuk meng-handle memproses data yang telah diinput menggunakan looping seperti for / while ataupun sesuatu nilai inputan yang sudah pasti di definisikan nilai yang akan di proses menggunakan case , kita mesti mengetahui kapan kita mesti menggunakannya. berikut adalah contoh untuk menampilkan output huruf 1-6 dengan berbagai model perulangan :

1.For

```
int[] array1 = {0, 1, 2, 3, 4, 5};  
  
for (int i=0; i<6; i++)  
{  
    System.Console.WriteLine(array1[i].ToString());  
}
```

2.For Each

```
int[] array1 = {0, 1, 2, 3, 4, 5};  
  
foreach (int n in array1)  
{  
    System.Console.WriteLine(n.ToString());  
}
```

3.While

```
int[] array1 = {0, 1, 2, 3, 4, 5};  
int x = 0;  
  
while (x < 6)  
{  
    System.Console.WriteLine(array1[x].ToString());  
    x++;  
}
```

4. Do While

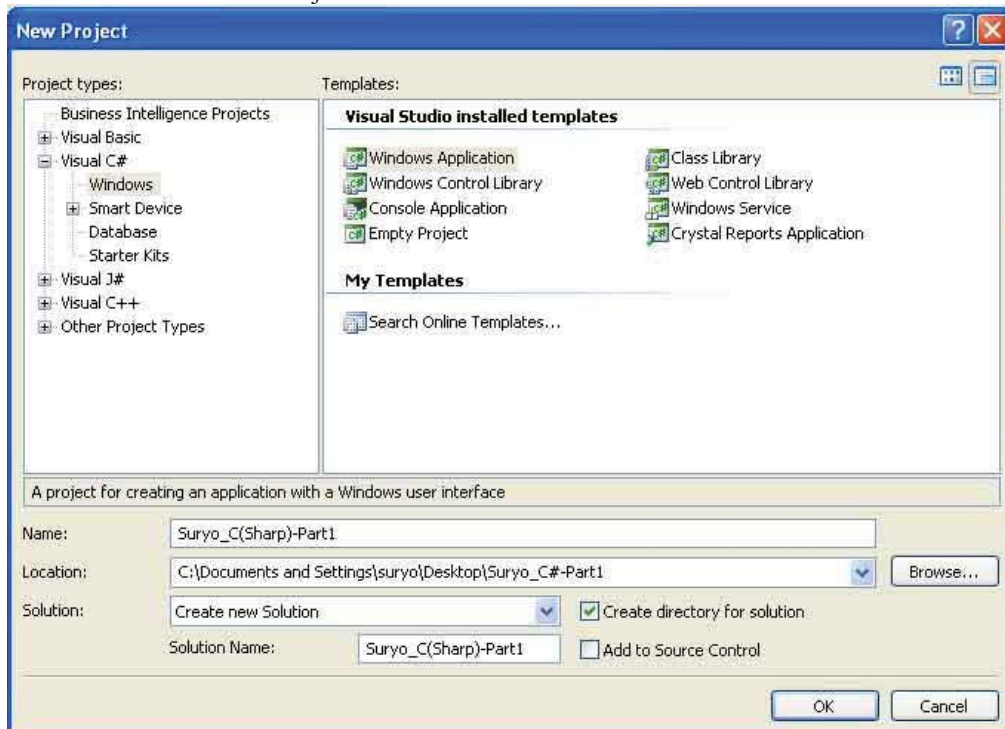
```
int[] array1 = {0, 1, 2, 3, 4, 5};  
int x = 0;  
  
do  
{  
    System.Console.WriteLine(array1[x].ToString());  
    x++;  
} while(x < 6);
```

Bila anda sudah memahami basic looping dari perulangan di atas , kita dapat langsung memulai membuat Project Baru , berikut langkahnya Step By Step :

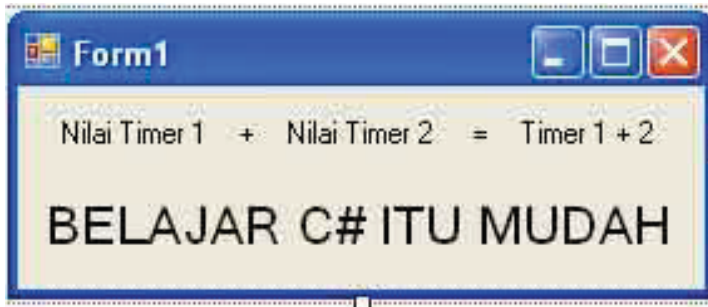
1. Pertama – tama Buka Visual Studio anda dari Start > All Program > Microsoft Visual Studio 2005 > Microsoft Visual Studio 2005 atau dengan membuka run dan ketik “devenv”



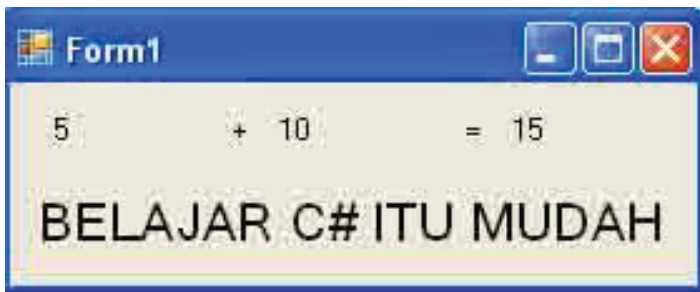
2. Kemudian Pilih New Project



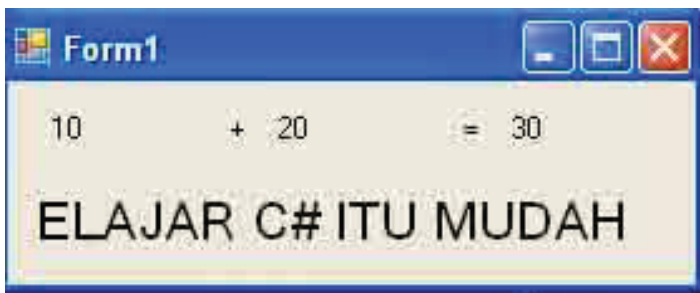
3. Drag komponen pada menu toolbox di kiri, drag komponen label sebanyak 6 dan 1 Timer, kemudian design form sehingga menjadi seperti berikut :



4. Kita menginginkan aplikasi kita setiap detik untuk nilai label 1 selalu ditambah 5 dan nilai label 2 selalu ditambah 10, dan label ketiga merupakan penjumlahan dari label 2 dan 3, kemudian kita menginginkan kalimat “BELAJAR C# ITU MUDAH” untuk bergerak dari kanan ke kiri, berikut tampilan Form dijalankan maka akan tampil seperti berikut pada interval detik pertama :



Kemudian pada interval timer ke 2 angka penjumlahan akan terus bertambah dan huruf “BELAJAR C# ITU MUDAH” akan bergerak dari kanan hingga kiri.



5. Berikut ini adalah contoh pemrograman beserta penjelasannya, pertama-tama deklarasi variable di dalam form utama seperti berikut

```
//variabel nilai awal untuk lbl_timer1 dan lbl_timer2 sebelum dijumlah
int a = 0;
int b = 0;
//membuat variabel bertipe string dan menampung kalimat ke dalamnya
string kata = "BELAJAR C# ITU MUDAH";
//variabel panjang dan nilai awal mula-mula
```

```
int panjang = 0;  
int nilai_awal = 0;
```

Kemudian agar setiap detik nilai berubah dan kalimat pun bergerak , maka kita perlu timer, double klik timer yang telah kita drag ke dalam form , dan kemudian kita isikan baris perintah berikut :

```
//Mencari Panjang dari kalimat "BELAJAR C# ITU MUDAH"  
//kemudian meng-konversi nilai ke integer  
panjang = Convert.ToInt16(kata.Length);  
//variabel a ditambah 5 , dan b ditambah 10  
a = a + 5;  
b = b + 10;  
//ganti nilai dari masing2 Label sesuai variabel dan  
//mengconvert nilai variabel yg tadinya bertipe integer ke string  
lbl_Timer1.Text = Convert.ToString(a);  
lbl_Timer2.Text = Convert.ToString(b);  
lbl_Hasil.Text = Convert.ToString(a + b);  
//reset nilai menjadi 0 ketika variabel bernilai sama  
if (nilai_awal == panjang)  
{  
    nilai_awal = 0;  
}  
//ambil kalimat sesuai indexnya untuk ditampilkan ke dalam label  
lbl_belajar.Text = kata.Substring(nilai_awal, panjang - nilai_awal);  
//variabel nilai awal di tambah 1  
nilai_awal = nilai_awal + 1;
```

Selamat , anda telah berhasil membuat latihan part 1. mari kita belajar lebih lanjut untuk mengasah kemampuan kita. Simak Terus Step By Step Latihan ini.

Penutup

Mempelajari C# mungkin agak sulit dibandingkan bahasa pemrograman VB yang lebih bersahabat dan lebih mudah di mengerti , tetapi ini merupakan suatu tantangan bagi kita sebagai Praktisi IT untuk mendalaminya , pada pertemuan pertama ini kita mempelajari basic dari C# agar dapat kita terapkan pada pertemuan selanjutnya.

Referensi

- MSDN 2005
- <http://msdn.microsoft.com>
- www.gotdotnet.com

C# Part 2 – Class Dan Array

M.Suryo Pranoto
suryodesign@yahoo.co.id
<http://suryostudio.net>

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

Dalam artikel ini kita akan mempelajari konsep dari object-oriented programming dalam bahasa pemrograman C#.seperti object dan class , untuk mempelajari artikel ini setidaknya kita mesti memiliki basic dari bahasa pemrograman agar dapat mengerti.

Mungkin dengan membaca artikel ini saja belum cukup untuk mendapatkan pengertian dari Object dan Class , jadi saya memberikan contoh aplikasi dari artikel ini agar dapat dimengerti lebih jelas dan dapat di pelajari lebih lanjut , dan diharapkan bagi para pembaca juga untuk tidak tergantung kepada artikel ini dan membaca artikel lainnya agar dapat mengerti lebih jauh tentang Object dan Class ini.

Pada pertemuan ke 2 ini kita akan membahas :

- 1.Belajar membuat Class
- 2.Belajar membuat Class dalam class
- 3.Belajar menggunakan array untuk digunakan di dalam class
- 4.Menampilkan Hasil dari class tersebut ke dalam Console Application

Isi

OOP atau yg biasa kita sebut dengan Object Oriented Programming merupakan cara baru untuk membuat aplikasi program computer , sebelum OOP ada , programmer membuat aplikasi komputer menggunakan procedural programming (atau yg biasa kita buat structure programming) untuk menyelesaikan masalah yang di hadapi , tapi ketika OOP dapat menyelesaikan banyak masalah disbanding dengan procedural programming , maka kebanyakan programmer dan developer mulai menggunakan OOP languages. Dalam procedural programming kebanyakan fungsi dalam program ditulis dalam beberapa modul / dapat lebih dari 1 module tergantung dengan aplikasi, dan modul ini tergantung antara yang satu dengan yang lainnya , dan bila kita ingin mengubah modul ini maka kita mesti mengganti baris2 yang

telah kita tulis dalam modul tadi , tapi dengan OOP kita dapat menulis bagian-bagian yang independent dari program yang disebut dengan class yang menggambarkan bagian dari fungsi program tersebut dan ketika kita ingin mengubah isi dari class ini, kita dapat mengubah target dari fungsi class yang telah kita buat yang mengandung masalah tersebut karena class ini dapat kita turunkan.

Class dan Object :

Mungkin tidak mudah untuk mengerti arti dari sebuah class dan object tetap penulis akan mencoba menerangkannya , sebenarnya konsep class dan object adalah menghubungkan satu dan yang lainnya dan biasanya para pemula dalam pemrograman tidak mengerti dan tidak peduli dan hanya berpikir hal ini menghabiskan waktu dalam mempelajari C#.

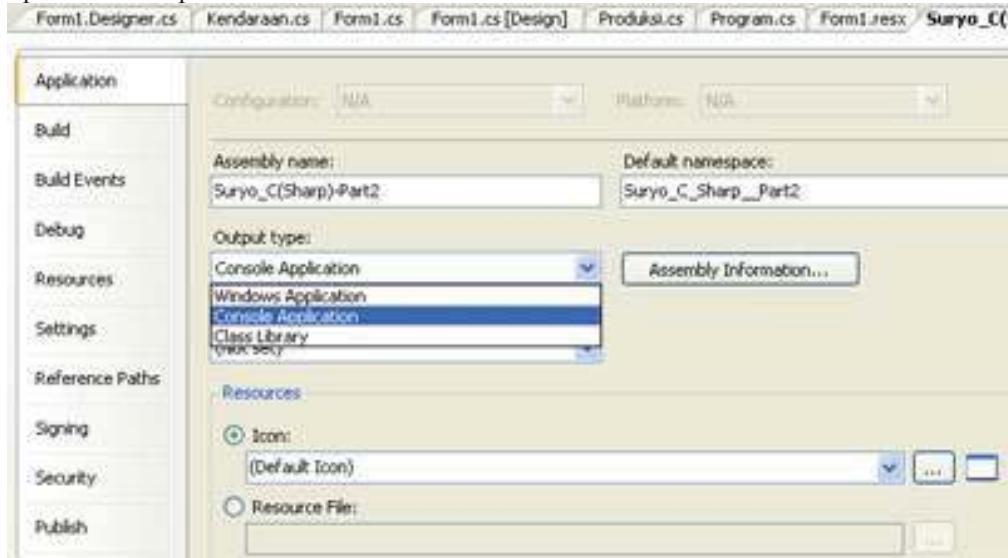
Sebagai contoh dalam Bumi ini . kita memiliki class dan object yang merupakan dari bumi ini yang kita gambarkan dalam sebuah makhluk hidup dan aktivitasnya. Semua yang ada di bumi ini disebut dengan object seperti manusi adalah objek , dan binatang juga objek. Mudah dimengerti bukan ? lalu apa itu class ?

Dalam dunia ini kita memiliki perbedaan antara objek yang satu dengan objek yang lainnnya , sebagai contoh adalah pengelompokkan klasifikasi (seperti nama kita) , yaitu saya adalah object, david juga objek , maria juga object , maka kami adalah **People Class** , kemudian saya memiliki kucing bernama andi , david memiliki kucing bernama puppy dan maria memiliki kucing bernama anie maka mereka termasuk ke dalam **Cat Class** , jadi dengan class ini kita dapat mendeskripsikan properties dan fungsi dari setiap group dari objek yang ada.

C# Class tergantung penggunaannya pada blok2 bahasa pemrograman. Yang saya maksud blok2 disini adalah setiap kita bekerja dengan C# dan membuat sebuah class pada program , maka class dari template yang telah kita buat tadi memiliki fungsi dan properti atau behavior yang sama dengan class sebelumnya tetapi setiap kita buat class ini kita dapat mengubah nilai atau properties yang ada. Sebagai Contoh kita membuat **People Class** , manusia ini memiliki **nama** , **tinggi** , dan **berat** . maka kita dapat membuat mahasiswa yang memiliki nama , tinggi , dan berat yang berbeda-beda juga.

Mari kita langsung saja membuat nya ke dalam Form C# kita dan mari kita buka Visual Studio 2005 kita dan mengikut Step By Step berikut :

1. add Form baru seperti biasa dan jangan lupa untuk mengganti start-up propertiesnya dengan tipe **Console** seperti berikut ini :



2. Kemudian anda akan bertanya bagaimana caranya membuat suatu class dalam C(Sharp) ? Pertama klik kanan pada Solution Explore dan add class seperti berikut :



Kemudian double klik class tersebut dan tulis isi class seperti berikut :Public class Kendaraan

```
class Kendaraan
{
    public string PlatNomor ;
    public string Model ;
    public string Manufaktur ;
    public int SpeedMax ;
}
```


Kemudian setelah kita add class yang telah berisi variabel2 tersebut bagaimana cara menerapkannya di dalam form yang akan kita gunakan ? kita mesti mendefinisikan suatu object yang menjadi bagian dari class kendaraan tersebut

```
Kendaraan mobilPertamaku;  
mobilPertamaku = new Kendaraan();
```

jangan bingung melihat cara pendeklarasian tersebut , saya akan menjelaskannya , baris pertama akan mendeklarasikan variabel mobilPertamaku sebagai type dari kendaraan , kemudian baris ke 2 untuk "new Kendaraan" membuat suatu perintah yang di spesifikasikan sebagai object baru dari type kendaraan dan menyimpan nya ke dalam lokasi dari variabel mobilPertamaku, operator baru ini akan menyimpan semua properti yang sama yang di miliki oleh class kendaraan, kurang lebih bisa kita bilang mobilPertamaku adalah object baru dari kendaraan yang akan kita buat ketika variabel di panggil atau dengan mudah kita bilang "mobilPertamaku adalah salah satu contoh kendaraan yang baru dan memiliki properti yang sama seperti model , manufaktur , maxSpeed , PlatNomor yang dimiliki oleh kendaraan"

kemudian kita akan mencoba mendefinisikan variabel secara langsung seperti berikut untuk mencoba apakah object yg telah kita buat benar2 memiliki properti seperti kendaraan.

```
Kendaraan mobilPertamaku;  
mobilPertamaku = new Kendaraan();
```

```
mobilPertamaku.PlatNomor = "BE 71 AA" ;  
mobilPertamaku.Model = "Suzuki New Baleno" ;  
mobilPertamaku.SpeedMax = 160 ;  
mobilPertamaku.Manufaktur = "Suzuki";
```

```
Console.WriteLine ("Mobilku : " + mobilPertamaku.Model + ", Buatan : " +  
mobilPertamaku.Manufaktur + "/n" + "Memiliki kecepatan : " + mobilPertamaku.SpeedMax +  
"/n" + "Dan memiliki Plat Nomor : " + mobilPertamaku.PlatNomor );
```

```
mobilPertamaku.PlatNomor = "BE 71 AA" ;  
//merupakan cara pengisian variabel secara langsung ke dalam suatu object yang telah di buat  
dan bukan melalui inputan, melalui latihan step by step kita akan mencoba ke tahap lebih lanjut  
untuk mempelajari ini
```

Kemudian setelah ini kita akan belajar bagaimana cara membuat alias dari object dan cara mengosongkan object. Tambahkan baris berikut ini setelah yang anda ketik tadi

```
Kendaraan MobilCoklat = mobilPertamaku ;  
MobilCoklat.Model = "Suzuki APV" ;  
Console.WriteLine("\n");  
Console.WriteLine("Mobil pertamaku adalah : " + mobilPertamaku.Model);  
  
mobilPertamaku = null;  
MobilCoklat.Model = "Suzuki APV";  
Console.WriteLine("Mobil Coklat pertamaku adalah : " + MobilCoklat.Model);
```

Disini kita akan membuat “alias” untuk nama object yang telah kita buat tadi dengan object mobilpertamaku akan kita buat aliasnya menjadi MobilCoklat , jadi disini kita bisa mengisi object melalui *MobilCoklat.Model* atau dengan melalui *mobilpertamaku.model*

Dan bila kita mengosongkan salah satu object tersebut dengan *mobilPertamaku = null*

Maka object tersebut tidak dapat di akses tetapi ingat alias dari object tersebut masih dapat diakses , karena itu bila ingin menghapus semuanya maka hapuslah nama object berikut alias dari object tersebut agar object tersebut benar2 terhapus.

Sekarang kita akan mencoba untuk membuat Class yang berada dalam class itu, jangan pernah berpikir bahwa ini sulit sebelum mencoba ,

Pertama add suatu class dan beri nama class produksi. Dan deklarasikan variabel tahun produksi seperti berikut di dalamnya :

```
class Produksi  
{  
    public int tahunProduksi;  
}
```

Kemudian di dalam class kendaraan yang telah kita buat , kita akan mencoba untuk memasukkan class produksi tadi ke dalam class kendaraan sehingga menjadi seperti berikut :

```
class Kendaraan  
{  
    public string PlatNomor ;  
    public string Model ;  
    public string Manufaktur ;  
    public int SpeedMax ;  
    public Produksi keteranganMobil ;  
}
```

Setelah berhasil membuat class dalam class , sekarang kita akan menciptakan object di atas dengan menggunakan array yang bersifat dinamis yang bisa kita buat sesuai dengan kebutuhan yang kita inginkan , seperti pembeli contohnya, kita tidak akan pernah meramal atau tau berapa banyak pembeli yang akan datang ke toko kita yang akan berbelanja di toko kita , tetapi pembeli datang bila mereka berbelanja , maka pertama sebelum membeli mereka akan menanyakan produk kita yaitu “barang” dan berapa “harga” dari barang itu sendiri , kita dapat

menganggap manusia disini sebagai object kita dan barang maupun uang sebagai variabel yg akan selalu kita gunakan.

mari kita buat class lokal di dalam form tersebut yang nantinya akan kita gunakan dengan memanfaatkan array.

```
public class pembeli
{
    public string nama ;
    public double harga ;
}
```

Kemudian kita akan menambahkan dalam form kita di bawah baris-baris perintah yang telah kita ketik sebelumnya :

```
Console.Write("Masukkan jumlah pembeli : ");
string Pembeli = Console.ReadLine();
int jumlahPembeli = Convert.ToInt32(Pembeli);
pembeli[] listPembeli = new pembeli[jumlahPembeli];

for (int i = 0; i < listPembeli.Length; i++)
{
    Console.Write("Pembeli ke " + (i + 1) + " : ");
    string NamaPembeli = Console.ReadLine();
    Console.Write("Tawaran Harga : ");
    string hargaTawaran = Console.ReadLine();
    double convertHargaTawaran = Convert.ToDouble(hargaTawaran);
    pembeli thisBuyer = new pembeli();
    thisBuyer.nama = NamaPembeli;
    thisBuyer.harga = convertHargaTawaran;
    listPembeli[i] = thisBuyer ;
}

Console.Write("\n");
foreach (pembeli i in listPembeli)
{
    System.Console.WriteLine("Pembeli dengan nama " + i.nama + " berani membeli dengan harga : " + i.harga);
}
```

Saya akan mencoba menjelaskannya beberapa syntax diatas :

console.write = kita gunakan untuk menulis

Console.read = kita gunakan untuk membaca inputan

Int jumlahPembeli = deklarasi variable bertipe int

pembeli[] listPembeli = new pembeli[jumlahPembeli] = membuat object baru dengan nama listPembeli dari pembeli dengan batas array yang telah kita tentukan

thisBuyer.nama = NamaPembeli = mengisi variabel nama dengan isi dari variabel

namaPembeli

Console.Write("/n") = kita gunakan untuk mencetak spasi

foreach (pembeli i in listPembeli) = kita gunakan untuk membaca setiap object yg ada dalam array tersebut , akan terus looping dari awal hingga akhir array.

Berikut Saya akan Paste total dari semua baris perintah / code pada form maupun Class .

Form1 :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Suryo_C_Sharp__Part2
{
    public partial class Form1 : Form
    {
        public class pembeli
        {
            public string nama ;
            public double harga ;
        }

        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            Kendaraan mobilPertamaku;
            mobilPertamaku = new Kendaraan();

            mobilPertamaku.PlatNomor = "BE 71 AA" ;
            mobilPertamaku.Model = "Suzuki New Baleno" ;
            mobilPertamaku.SpeedMax = 160 ;
            mobilPertamaku.Manufaktur = "Suzuki";

            Console.WriteLine("Mobilku : " + mobilPertamaku.Model + ", Buatan : " +
            mobilPertamaku.Manufaktur +
            "/n" + "Memiliki kecepatan : " + mobilPertamaku.SpeedMax + "/n" + "Dan
            memiliki Plat Nomor : " + mobilPertamaku.PlatNomor );

            Kendaraan MobilCoklat = mobilPertamaku ;
            MobilCoklat.Model = "Suzuki APV" ;
            Console.Write("/n");
        }
    }
}
```

```
Console.WriteLine("Mobil pertamaku adalah : " + mobilPertamaku.Model);

mobilPertamaku = null;
MobilCoklat.Model = "Suzuki APV";
Console.WriteLine("Mobil Coklat pertamaku adalah : " + MobilCoklat.Model);

Produksi YearProduct = new Produksi();
YearProduct.tahunProduksi = 2004;
MobilCoklat.keteranganMobil = YearProduct;
Console.WriteLine("Tahun          Produksi          :          "          +
Convert.ToString(MobilCoklat.keteranganMobil.tahunProduksi + "\n"));

Console.Write("Masukkan jumlah pembeli : ");
string Pembeli = Console.ReadLine();
int jumlahPembeli = Convert.ToInt32(Pembeli);
pembeli[] listPembeli = new pembeli[jumlahPembeli];

for (int i = 0; i < listPembeli.Length; i++)
{
    Console.Write("Pembeli ke " + (i + 1) + " : ");
    string NamaPembeli = Console.ReadLine();
    Console.Write("Tawaran Harga : ");
    string hargaTawaran = Console.ReadLine();
    double convertHargaTawaran = Convert.ToDouble(hargaTawaran);
    pembeli thisBuyer = new pembeli();
    thisBuyer.nama = NamaPembeli;
    thisBuyer.harga = convertHargaTawaran;
    listPembeli[i] = thisBuyer ;

}
Console.Write("\n");

foreach (pembeli i in listPembeli)
{
    System.Console.WriteLine("Pembeli dengan nama " + i.nama + " berani
membeli dengan harga : " + i.harga);
}
}
}
```

Class Kendaraan

```
using System;
using System.Collections.Generic;
using System.Text;

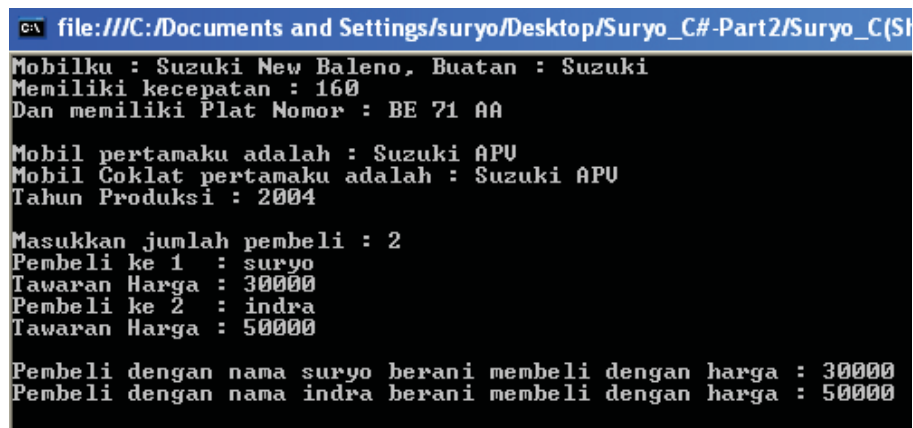
namespace Suryo_C_Sharp__Part2
{
    class Kendaraan
    {
        public string PlatNomor ;
        public string Model ;
        public string Manufaktur ;
        public int SpeedMax ;
        public Produksi keteranganMobil ;
    }
}
```

Class Produksi

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Suryo_C_Sharp__Part2
{
    class Produksi
    {
        public int tahunProduksi;
    }
}
```

Tampilan dari Aplikasi ketika dijalankan :



```
file:///C:/Documents and Settings/suryo/Desktop/Suryo_C#-Part2/Suryo_C(SI
Mobilku : Suzuki New Baleno, Buatan : Suzuki
Memiliki kecepatan : 160
Dan memiliki Plat Nomor : BE 71 AA
Mobil pertamaku adalah : Suzuki APU
Mobil Coklat pertamaku adalah : Suzuki APU
Tahun Produksi : 2004
Masukkan jumlah pembeli : 2
Pembeli ke 1 : suryo
Tawaran Harga : 30000
Pembeli ke 2 : indra
Tawaran Harga : 50000
Pembeli dengan nama suryo berani membeli dengan harga : 30000
Pembeli dengan nama indra berani membeli dengan harga : 50000
```

C# Part 3 – Static Keyword & Function

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://suryostudio.net>

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

Static Keyword dan Function mungkin sudah tidak asing lagi di mata para programmer , Static Keyword dan Function ini sudah banyak kita temui tetapi mungkin hanya ada beberapa artikel yang dapat menjelaskan tentang penggunaan Static Dan Function ini secara mudah dan jelas agar dapat langsung mudah di praktekan.

Saya akan mencoba menjelaskannya step by step agar lebih mudah di mengerti , dan juga memberikan contoh agar di ketahui penggunaan dan penerapannya.

Isi

Di Pertemuan ke 3 ini , kita akan belajar mengenai penggunaan Static Keyword dan bagaimana menggunakan function pada C#.

Pertama kita mesti mengetahui apa itu Static ?

Static disini maksudnya bisa berupa variable / function yang kita deklarasikan dengan menggunakan keyword Static di depan variable / function tersebut seperti berikut ini

```
public static double nStaticNilai;  
public static void mahasiswaClassFunction() // static  
{  
    Console.WriteLine("mahasiswa class function");  
}
```


Static hanya dapat diubah melalui dimana tempat itu di deklarasikan , contohnya apa bila variable tersebut kita deklarasikan dalam class

```
class Mahasiswa
{
    public static double nStaticNilai;
    public static void mahasiswaClassFunction() // static
    {
        Console.WriteLine("mahasiswa class function");
    }
}
```

variable static seperti berikut dapat langsung kita panggil dengan perintah berikut :

```
Mahasiswa.nStaticNilai = 3;
```

Hasilnya tentu saja akan berbeda apabila kita membuat suatu member/anggota baru dari class tersebut seperti berikut :

```
Mahasiswa mhs = new Mahasiswa();
mhs.nStaticNilai <<<<< baris perintah berikut ini tidak akan dapat di
jalankan karena variabel ini di akses via member / object yang di buat dan
bukan melalui class itu sendiri
```

saya harap contoh di atas dapat menjelaskan mengenai Static Keyword dan sekarang kita akan mencoba membahas function.

Function itu sendiri adalah Suatu kumpulan perintah yg telah kita siapkan dan dapat kita panggil sewaktu-waktu ingin kita gunakan , function ini bisa menerima nilai yang kita lempar, dalam passing parameternya biasanya menggunakan kata “ref” di depan variable yang akan dilempar , dan function itu bisa mengembalikan suatu nilai yang telah kita hitung di dalamnya., berikut contoh dari penggunaan function :

```
//membuat object mahasiswa baru
Mahasiswa mhs2 = new Mahasiswa();
//memanggil function yang telah kita gunakan
mhs2.hitungNilai(ref mhs2.nilaiUjian1, ref mhs2.nilaiUjian2 , ref no);

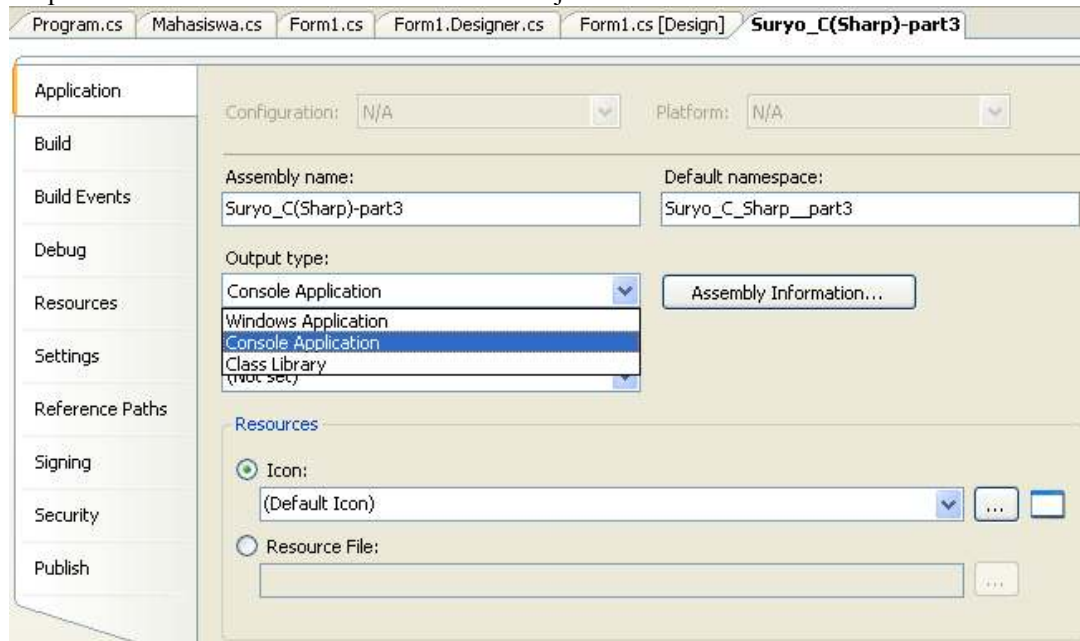
//function hitungNilai yang telah kita definisikan
public void hitungNilai(ref int a, ref int b , ref int no)
{
    nNilai++;
    Console.WriteLine("%n");
    nStaticNilai = Convert.ToDouble(a) / Convert.ToDouble(b);
    Console.WriteLine("Nilai rata2 mahasiswa ke " + no + " : "
+ nStaticNilai);
}
```

Setelah kita mengerti apa itu Static Keyword dan function dan telah mengetahui bagaimana cara

penggunaannya dalam pemrograman. maka alangkah baiknya kita langsung mencoba membuatnya dalam **aplikasi console** :

saya akan mencoba membuat nya dalam step by step agar lebih mudah untuk di ikuti :

1. Buka Visual Studio 2005 atau ketik “devenv” pada run menu.
2. Beri nama Solution dan pilih directori Save di tempat yang anda inginkan , dan Ganti Startup Properties menjadi Console



3.Add Suatu Class dan beri nama class “mahasiswa”

4.Kemudian buat variable dan function pada Class hingga menjadi seperti berikut ini :

```
using System;
using System.Collections.Generic;
using System.Text;
namespace Suryo_C_Sharp__part3
{
    class Mahasiswa
    {
        public string nama = "";
        public double nNilai ; // non-static
        public static double nStaticNilai;// static
        public static int Nomor = 0 ;
        public int nilaiUjian1 = 0 ;
        public int nilaiUjian2 = 0 ;
        public void mahasiswaMemberFunction() // non-static
        {
            Console.WriteLine("mahasiswa member function");
        }

        public static void mahasiswaClassFunction() // static
        {
            Console.WriteLine("mahasiswa class function");
        }

        public void hitungNilai(ref int a, ref int b , ref int no)
            // non-static
        {
            nNilai++;
            Console.WriteLine("¥n");
            nStaticNilai = Convert.ToDouble(a) / Convert.ToDouble(b);
            Console.WriteLine("Nilai rata2 mahasiswa ke " + no + " :
"
                + nStaticNilai);
        }
    }
}
```

5.Pada Form1.cs buat variable deklarasi dan fungsi seperti berikut ini

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;

namespace Suryo_C_Sharp__part3
{
    public partial class Form1 : Form
    {
        public Form1 ()
        {

```

```
InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
    Mahasiswa mhs = new Mahasiswa(); // membuat object mhs
    Mahasiswa mhs2 = new Mahasiswa(); // membuat object mhs2
    Mahasiswa mhs3 = new Mahasiswa(); // membuat object mhs3

    mhs.mahasiswaMemberFunction();
    //memanggil function dari mhs, function ini dapat
    kita panggil
    //karena function ini tidak memiliki static
    //karena itu mhs sebagai member(object yg dibuat
    dari Mahasiswa)
    //dapat memanggil function tersebut

    //cobalah untuk memanggil mhs.mahasiswaClassFunction
    //lines ini tidak akan dapat dipanggil / dicompile
    karena memiliki
    //keyword "Static" di depan functionnya dan kita
    hanya dapat
    //memanggilnya melewati class itu sendiri dan tidak
    dapat mengakses
    //melewati object / member yang telah dibuat
    //jadi untuk memanggilnya kita langsung melewati classnya
    //seperti berikut ini :

    Mahasiswa.mahasiswaClassFunction();
    //via class dapat mengakses fungsi mahasiswa static

    //mhs.mahasiswaClassFunction
    //via object tidak dapat mengakses

    mhs.nNilai = 5 ; // via member
    Mahasiswa.nStaticNilai = 100; // via class

    Console.WriteLine("Masukkan nama mahasiswa pertama : ");
    string mhs2Nama = Console.ReadLine();

    string mhs2Nilai1 ;
    do
    {
        Console.WriteLine("Masukkan nilai pertama (max 100) : ");
        mhs2Nilai1 = Console.ReadLine();
    } while (Convert.ToInt16(mhs2Nilai1) > 100);

    string mhs2Nilai2;
    do
    {
        Console.WriteLine("Masukkan nilai kedua (max 100) : ");
        mhs2Nilai2 = Console.ReadLine();
    } while (Convert.ToInt16(mhs2Nilai2) > 100);
}
```

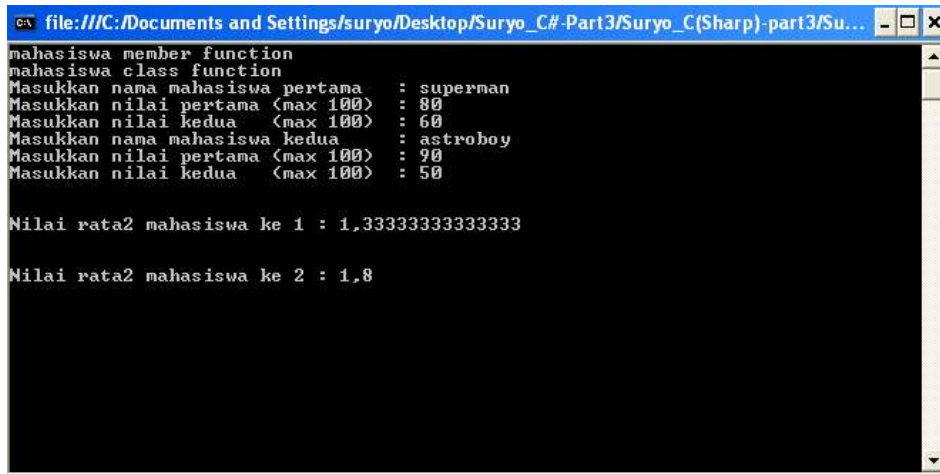
```
mhs2.nama = mhs2Nama;
mhs2.nilaiUjian1 = Convert.ToInt16(mhs2Nilai1) ;
mhs2.nilaiUjian2 = Convert.ToInt16(mhs2Nilai2) ;

Console.Write("Masukkan nama mahasiswa kedua : ");
string mhs3Nama = Console.ReadLine();
string mhs3Nilai1;
do
{
    Console.Write("Masukkan nilai pertama (max 100) : ");
    mhs3Nilai1 = Console.ReadLine();
} while (Convert.ToInt16(mhs3Nilai1) > 100);

string mhs3Nilai2;
do
{
    Console.Write("Masukkan nilai kedua (max 100) : ");
    mhs3Nilai2 = Console.ReadLine();
} while (Convert.ToInt16(mhs3Nilai2) > 100);

mhs3.nama = mhs3Nama;
mhs3.nilaiUjian1 = Convert.ToInt16(mhs3Nilai1);
mhs3.nilaiUjian2 = Convert.ToInt16(mhs3Nilai2);
int no = 1;
mhs2.hitungNilai(ref mhs2.nilaiUjian1, ref
mhs2.nilaiUjian2, ref no);
no++;
mhs3.hitungNilai(ref mhs3.nilaiUjian1, ref
mhs3.nilaiUjian2, ref no);
    }
}
```

Berikut adalah hasil Output dari aplikasi tersebut :



```
file:///C:/Documents and Settings/suryo/Desktop/Suryo_C#-Part3/Suryo_C(Sharp)-part3/Su...
mahasiswa member function
mahasiswa class function
Masukkan nama mahasiswa pertama : superman
Masukkan nilai pertama <max 100> : 80
Masukkan nilai kedua <max 100> : 60
Masukkan nama mahasiswa kedua : astroboy
Masukkan nilai pertama <max 100> : 90
Masukkan nilai kedua <max 100> : 50

Nilai rata2 mahasiswa ke 1 : 1,33333333333333
Nilai rata2 mahasiswa ke 2 : 1,8
```

Penutup

Mempelajari C# mungkin agak sulit dibandingkan bahasa pemrograman VB yang lebih bersahabat dan lebih mudah di mengerti , tetapi ini merupakan suatu tantangan bagi kita sebagai Praktisi IT untuk mendalaminya , pada pertemuan pertama ini kita mempelajari basic dari C# agar dapat kita terapkan pada pertemuan selanjutnya.

Referensi

- MSDN 2005
- <http://msdn.microsoft.com>
- www.gotdotnet.com

Biografi Penulis



M.Suryo Pranoto – 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.suryostudio.net , Blog : www.suryodesign.wordpress.com

C# Part 4 – M.Access Dengan OLE DB Provider

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://suryostudio.net>

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

Teknologi merupakan hal yang sangat pesat berkembang saat ini , dan sudah merupakan kebutuhan kita , semua data yang kita pakai saat ini tersimpan ke dalam suatu database, Kita dapat mengakses database tersebut , dalam hal programming untuk mengakses suatu database ini kita memerlukan Provider , bisa dengan OLE DB , ODBC , SQL Client ataupun Oracle Connection dan banyak lagi jenisnya , kita mesti mengetahui yang mana yang akan kita gunakan sesuai dengan kebutuhan kita. Setiap Provider mendukung jenis database yang berbeda dan juga memiliki kelebihan masing-masing.

Dalam artikel ini kita akan coba membahas OLE DB menggunakan database Microsoft Access. Kita akan coba membuat suatu Transaction Insert Update Delete yang simple terlebih dahulu atau Hard Code (Code yang tertera secara jelas dalam setiap blok-blok tanpa menggunakan suatu modul atau kelas yang di panggil berulang – ulang , tujuannya adalah agar latihan ini dianggap sebagai pemanasan dan dapat membuka logika pikir kita bagi yang baru mempelajari OLE DB dengan Microsoft Access ini , pada pertemuan selanjutnya baru kita akan masuk ke SQL Client dan saya akan membahas tentang fitur2 yang belum kita bahas pada page ini.

Isi

1. Buat database dengan Microsoft Access., design database dengan nama database **data** dan design form database seperti berikut dan jangan lupa untuk mengisi data.

MsBarang : Table		
	Field Name	Data Type
	KodeBarang	AutoNumber
	NamaBarang	Text
	HargaBarang	Number
	JumlahBarang	Number

2. Buat Project C# baru, dalam aplikasi saya , saya menamakannya dengan folder market dan kemudian save di directori Yang anda kehendaki , kemudian pindahkan database anda ke dalam directori aplikasi anda dengan alamat

NamaFile > NamaFile > Bin > Debug > “PASTE DATABASE DI DALAM SINI”

Mungkin muncul pertanyaan kenapa mesti kita mesti memindahkan database di dalam folder Debug ini ? Karena di dalam folder debug itu dimana ada file output dari aplikasi kita dimana aplikasi di jalankan. Dan nantinya kita akan membuat string koneksi database untuk meload data dimana database itu diambil dengan alamat direktori yang beralamat 1 folder dengan alamat dimana aplikasi jalankan , ini alas an kita menaruh database di dalam sini , kita bias saja menaruhnya di sembarang tempat sesuai kemauan tapi itu tergantung dengan alamat Connection String yang kita gunakan bila bersifat statis dan tidak dinamis.

3. Kemudian dalam aplikasi , double klik form anda dan tambah kan baris perintah berikut :

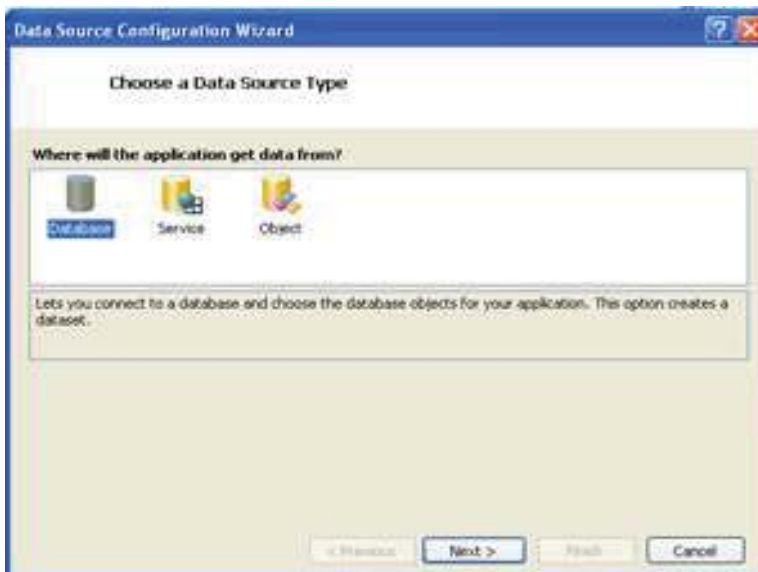
```
using System.Data.OleDb;
```

Tujuan Perintah ini Akan kita gunakan untuk memanggil beberapa perintah seperti berikut, dan masih banyak juga perintah lainnya selain perintah di bawah ini :

```
OleDbCommand , OleDbCommandBuilder , OleDbConnection ,  
OleDbConnectionStringBuilder , OleDbDataAdapter ,  
OleDbDataReader , OleDbEnumerator , OleDbError
```

4. Kemudian Hal Pertama adalah mencari Koneksi yang perlu kita lakukan.,
Lakukan Perintah berikut ini :

> **Klik Data , Add New Data Source , Pilih Database kemudian Next**



> Kemudian New Connection > Pilih Data Source dengan “Microsoft OleDb” Dan Kemudian Browse Database **data**. Yang telah kita buat tadi. Dan Test Connection , Kemudian Setelah anda Oke ambil String Connection yang ada , string ini nanti akan kita gunakan untuk membuat OLEDBCONNECTION.



5. Kita akan Memulai Pengertian tentang beberapa Command yang akan kita pakai

Yaitu Pertama kita mesti mengetahui Command Type , berikut *jenis2 command type*

CommandType	Description
Text	Perintah SQL yang digunakan untuk mendefinisikan data yang akan dieksekusi pada data source
StoredProcedure	Nama Dari store procedure yang telah kita buat , kita dapat menggunakan property dari parameter untuk memberikan perintah pada input atau output parameter dan dapat mengembalikan nilai tergantung dengan metode execute yang dipanggil, kecuali dalam Execute Reader tidak akan mengembalikan nilai / values
TableDirect	Untuk Memanggil Nama Dari Table

Berikut adalah hasil Nilai yang dikembalikan Dari Command Type

Command	Return Value
---------	--------------

ExecuteReader	mengembalikan DataReader object.
ExecuteScalar	Mengembalikan satu nilai / value.
ExecuteNonQuery	Eksekusi perintah yang tidak mengembalikan rows.
ExecuteXMLReader	Mengembalikan XmlReader. Hanya ada pada SqlCommand object saja.

Berikut ini adalah Jenis Eksekusi Dari **OleDbCommand**

Item	Description
<u>ExecuteReader</u>	Mengembalikan rows , dan tidak akan bekerja bila anda mengeksekusi perintah seperti insert, update , delete dan hanya dapat read saja / one way
<u>ExecuteNonQuery</u>	Eksekusi Command SQL INSERT, DELETE, UPDATE, dan SET statement.
<u>ExecuteScalar</u>	Mengembalik Satu Nilai Saja Contoh Aggregate nilai dari database seperti Count , Max , Min dan sejenisnya.

Berikut saya akan membuat masing2 Snippet dari Cara Eksekusi. , tetapi jangan lupa untuk membuat string koneksi pada bagian atas Form

```
public string strCon = "Provider=Microsoft.Jet.OLEDB.4.0;Data  
Source=|DataDirectory|/data.mdb" ;
```

1.EXECUTENONQUERY(EX : Insert , Update , Delete)

```
// Buat Koneksi
OleDbConnection connection = new OleDbConnection(strCon);
// Buat Command Yang nantinya akan kita gunakan untuk perintah
OleDbCommand command = new OleDbCommand();
// Try Catch kita gunakan untuk handle kesalahan.
try
{
    //Buka koneksi
    connection.Open();
    command.Connection = connection;
    //Pilih jenis perintah yang digunakan
    command.CommandType = CommandType.Text;
    //Eksekusi Perintah dari Command
    command.ExecuteNonQuery();
    //Tutup Koneksi
    connection.Close();
}
//Buka koneksi
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

2.Execute Reader

```
// Buat Koneksi
OleDbConnection connection = new OleDbConnection(strCon);
// Buat Command Yang nantinya akan kita gunakan untuk perintah
OleDbCommand command = new OleDbCommand();
// Try Catch kita gunakan untuk handle kesalahan.
try
{
    connection.Open(); //Buka koneksi
    command.Connection = connection;
    //Pilih jenis perintah yang digunakan
    command.CommandType = CommandType.Text;
    //Buat Object DataReader Untuk Membaca data
    //kemudian Eksekusi Perintah dari Command Tersebut
    OleDbDataReader dr = command.ExecuteReader();
    while (dr.Read())
    {
        MessageBox.Show(dr[0].ToString() + " " + dr[1].ToString());
    }
    connection.Close(); //Tutup Koneksi
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

3. Execute Scalar

```
// Buat Koneksi
OleDbConnection connection = new OleDbConnection(strCon);
// Buat Command Yang nantinya akan kita gunakan untuk perintah
OleDbCommand command = new OleDbCommand();
// Try Catch kita gunakan untuk handle kesalahan.
try
{
    int jumlahData ;
    //Buka koneksi
    connection.Open();
    command.Connection = connection;
    //Pilih jenis perintah yang digunakan
    command.CommandType = CommandType.Text;
    //Buat Object DataReader Untuk Membaca data
    command.CommandText = "select count(*) from msbarang"
    //kemudian Eksekusi Perintah dari Command Tersebut
    jumlahData = command.ExecuteScalar();
    MessageBox.Show((string)jumlahData);
    connection.Close(); //Tutup Koneksi
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

4. TableDirect dengan Fill

```
//buat koneksi
OleDbConnection connection = new OleDbConnection(strCon);
//buat command dan isi dengan NamaTable , bukan perintah SQL
OleDbCommand command = new OleDbCommand("MsBarang");
try
{
    command.Connection = connection;
    command.CommandType = CommandType.TableDirect;
    //buat dataset dengan nama ds1
    DataSet ds1 = new DataSet();
    OleDbDataAdapter dal = new OleDbDataAdapter(command);
    //isi query dari MSBARANG ke dalam ds1 dengan nama table "barang"
    dal.Fill(ds1, "barang");
    //tampilkan hasil ke dalam data grid
    dataGridView1.DataSource = ds1;
    dataGridView1.DataMember = "barang";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```


5.Fill dengan Search dan TableDirect

```
//buat koneksi dan isi koneksi
OleDbConnection connection = new
OleDbConnection("Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=|DataDirectory|/data.mdb");

//buat paramater1
OleDbParameter parameter1 = new OleDbParameter();
//buat command dan isi dengan perintah
OleDbCommand command = new OleDbCommand("select * from MsBarang where
[NamaBarang] = ?");
//add parameter1 ke untuk mengisi ? pada perintah select
command.Parameters.Add(parameter1);
//isi parameter1 dengan variabel
parameter1.Value = txtKeyword2.Text ;
try
{
    //buka koneksi dan gunakan type jenis text
    command.Connection = connection;
    command.CommandType = CommandType.Text;
    //buat dataset baru dan isi data kedalam dataset tersebut
    DataSet ds1 = new DataSet();
    OleDbDataAdapter dal = new OleDbDataAdapter(command);
    dal.Fill(ds1, "barang");
    //tampilkan data dari dataset ke dalam datagrid
    dataGridView1.DataSource = ds1;
    dataGridView1.DataMember = "barang";
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

Berikut adalah tampilan Output dari aplikasi, anda dapat mendownload aplikasi berikut dari alamat <http://www.suryodesign.asia>

The screenshot shows a Windows application window titled 'Form1' with a menu bar containing 'Menu Master', 'Exit', and 'View'. Inside the window is a 'MainMenu' dialog box. The dialog box features a table with the following data:

KodeBarang	NamaBarang	HargaBarang	JumlahBarang
1	Sendok1	25001	1001
2	Garup	2000	125
3	Peing	3000	150
4	Gelas	3500	250
5	bbb	0	0

Below the table, there are tabs for 'Insert', 'Search', and 'Delete / Update'. The 'Delete / Update' tab is active, showing a form with the following fields and buttons:

- ID Barang: 3 (dropdown menu)
- Nama Barang: Peing (text box)
- Jumlah Barang: 150 (text box)
- Harga Barang: 3000 (text box)
- Buttons: Refresh, Delete, Update

Penutup

Mempelajari C# mungkin agak sulit dibandingkan bahasa pemrograman VB yang lebih bersahabat dan lebih mudah di mengerti , tetapi ini merupakan suatu tantangan bagi kita sebagai Praktisi IT untuk mendalaminya , pada pertemuan Kee-mpat ini kita mempelajari basic Database dengan OLEDB , ini masih merupakan langkah awal dari langkah kita menuju programmer , masih banyak langkah yang mesti kita lewati

Referensi

- MSDN 2005
- <http://msdn.microsoft.com>
- www.gotdotnet.com

Part 5 –SQL 2005

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://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 bahasan kali ini saya akan menjelaskan mengenai C# dihubungkan dengan database SQL2005 , harap mempelajari C# Pert 4 terlebih dahulu agar lebih mengerti sebelum mempelajari bagian ini.

Saya akan mencoba menjelaskannya step by step agar lebih mudah di mengerti , dan juga memberikan contoh agar di ketahui penggunaan dan penerapannya.

Isi

Di Pertemuan ke 5 ini kita akan mempelajari cara membangun database, database merupakan sekumpulan table atau data yang terorganisi dan memiliki hubungan (relasi) di antara semua elemen , database disini masih merupakan sekumpulan entity yang berasal dari perancangan , dan kemudian kita menggunakan aplikasi antar muka dengan bahasa pemrograman C# untuk dapat memanagemen , mengelola , mengubah , dan memonitoring isi dari database file system dengan SQL 2005 kita tadi.

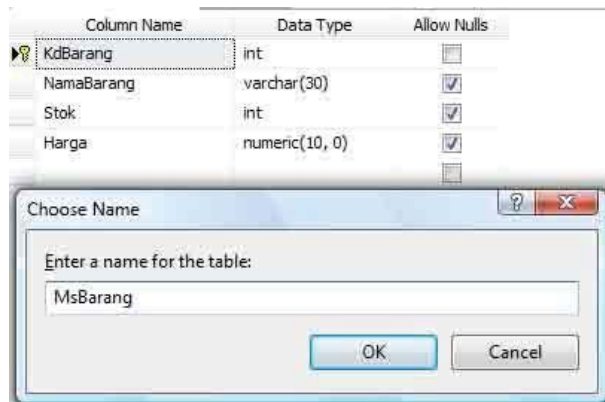
Perkembangan teknologi semakin pesat dari tahun ke tahun , begitu juga dengan penggunaanya yang makin bertambah banyak , sehingga menuntut database dengan struktur yang lebih baik

Komunitas eLearning IlmuKomputer.Com
Copyright © 2003-2007 IlmuKomputer.Com

dan juga aplikasi dengan flow yang baik agar dapat berjalan dengan cepat dan dapat menghindari atau meminimalis kesalahan yang terjadi.

Untuk Microsoft SQL Server 2005 ini sendiri saya menggunakan SQL 2005 Express yang merupakan versi gratis dari SQL Server dan dapat anda download di www.microsoft.com dengan nama file SQLSVR32.exe dan jangan lupa untuk mendownload SQL Management Studio dengan nama file SqlServer2005_SSMSEE.msi , file ini gratis dari Microsoft.

Pertama buat database dengan nama table MsBarang dengan nama kolom dan tipe databerikut :



Kita menggunakan SQLCLIENT untuk dapat menghubungkan aplikasi kita dengan database SQL2005 , pertama kita import System data.SQLClient sehingga menjadi :

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;
```

Kemudian kita rancang Form seperti berikut :

Kemudian cari koneksi SQL2005 dan definisikan dalam SqlConnection seperti berikut :

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

Kemudian Buat Syntax untuk menampilkan data ke dalam DataGridView seperti berikut pada BtnShowData :

```
SqlCommand scom = new SqlCommand();
//membuat SQLCOMMAND dengan alias(nama pengganti) dari SQLCOMMAND
scom.Connection = con;
//include koneksi ke dalam perintah
scom.CommandType = CommandType.Text;
//menentukan jenis perintah berupa text(perintah SQL biasa)
scom.CommandText = "select * from msbarang";
//perintah SQL

DataSet ds1 = new DataSet();
//membuat dataset object
SqlDataAdapter sda = new SqlDataAdapter(scom);
//membuat data adapter untuk meng-eksekusi perintah dari SQLCOMMAND(scom)
```

```
sda.Fill(ds1, "barang");  
//isi dari select from ms barang tadi kita tampung ke dalam dataset  
dataGridBarang.DataSource = ds1;  
//set datasource untuk menampilkan data yang berasal dari ds1  
dataGridBarang.DataMember = "barang";  
//set datagrid untuk menampilkan member dari dataset ds1 yang telah kita  
pilih pada datasource sebelumnya
```

Kemudian selanjutnya kita membuat insert data , berikut coding pada BTNINSERT :

```
if (txtNama.Text == "" | txtStok.Text == "" )  
//gunakan OR untuk argumen membandingkan  
{  
    MessageBox.Show("data mesti diisi");  
    //Environment.Exit(-1);  
    goto berhenti;  
    //Pindah Baris ke Berhenti dan skip baris selanjutnya  
}  
  
int num;  
//buat variabel num  
bool isNum = int.TryParse(txtHarga.Text.ToString() , out num);  
//membuat variabel is Num dan kemudian isi dari variabel isNum itu sendiri  
//mengubah type data dan menyimpan hasilna pada variabel num  
if(!isNum)  
//mengecek nilai isNum apakah false  
//( karena bukan number melainkan alfabet )  
{  
    MessageBox.Show("Harga mesti number");  
    //bila false maka tampilkan pesan harga mesti number  
    goto berhenti;  
    //Pindah Baris ke Berhenti dan skip baris selanjutnya  
}  
isNum = int.TryParse(txtStok.Text.ToString(), out num);  
//mengisi dari variabel isNum itu sendiri dengan  
//mengubah type data dari txtstok dan menyimpan hasilna pada variabel num  
if (!isNum)  
//mengecek nilai isNum apakah false  
//( karena bukan number melainkan alfabet )  
{  
    MessageBox.Show("stok mesti number");  
    //bila false maka tampilkan pesan harga mesti number  
    goto berhenti;  
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
```

```
}  
  
con.Open();  
//membuka koneksi  
SqlCommand scom = new SqlCommand();  
//membuat SQLCOMMAND dengan alias scom  
scom.Connection = con;  
//include koneksi ke dalam sqlcommand  
scom.CommandType = CommandType.Text;  
//menentukan jenis command , defaultnya adalah commandtype.text  
scom.CommandText = " insert into MsBarang(NamaBarang,Stok,Harga) values(' "  
+ txtNama.Text + "',' " + txtStok.Text + "',' " + txtHarga.Text + ")";  
//mengisi perintah SQL dengan insert data ke dalam table barang  
scom.ExecuteNonQuery();  
//execute ( jalankan perintah )  
con.Close();  
//akhiri koneksi ( tutup koneksi )  
  
btnShow.PerformClick();  
//jalankan btnshow untuk refresh data  
  
berhenti:  
Console.WriteLine("STOP");
```

Setelah kita insert data , sekarang kita akan reset textboxt menjadi empty kembali , berikut contoh btnreset :

```
txtNama.Text = "";  
txtStok.Text = "";  
txtHarga.Text = "";  
//mengosongkan field nama , stok , dan harga
```

Kemudian sekarang kita akan mencoba mencari data yang berasal dari data set yang kita tampung tadi dan mencari berdasarkan kolom primary key yaitu kodebarang

```
if (txtKode.Text == "")  
{  
    MessageBox.Show("Isi dengan Kode Barang");  
    goto selesai;  
    //kode barang yg akan dicari tidak boleh kosong ,  
    //kemudian bila kosong tampilkan message box  
    //dan lompat Code ke bagian Selesai
```



```
}  
  
SqlCommand scom = new SqlCommand();  
scom.Connection = con;  
scom.CommandType = CommandType.Text;  
scom.CommandText = "select * from msbarang";  
//buat sql command , include koneksi  
//tentukan jenis perintah , dan isi dengan select  
  
DataSet ds1 = new DataSet();  
SqlDataAdapter sda = new SqlDataAdapter(scom);  
sda.Fill(ds1, "barang");  
DataTable dt1 = ds1.Tables["barang"];  
DataColumn[] coll = new DataColumn[1];  
coll[0] = dt1.Columns["KdBarang"];  
dt1.PrimaryKey = coll;  
DataRow rw1 = dt1.Rows.Find(txtKode.Text);  
//cari data pada dataset dengan table barang pada dataset  
//kemudian tentukan primary key pada table tersebut  
//dan cari berdasarkan primary key tersebut  
  
txtNama.Text = (string)rw1["NamaBarang"];  
txtStok.Text = Convert.ToString(rw1["Stok"]);  
txtHarga.Text = Convert.ToString(rw1["harga"]);  
//tampilkan data yang didapat ke dalam textbox  
  
selesai:  
;
```

Kemudian sekarang kita akan mencoba mencari dengan metoda ke dua yaitu dengan search menggunakan like , agak sedikit berbeda dengan search yang tadi :

```
SqlCommand scom = new SqlCommand();  
scom.Connection = con;  
scom.CommandType = CommandType.Text;  
scom.CommandText = "select * from msbarang where NamaBarang like '%" +  
txtCari.Text + "%'";  
//buat sql command , include koneksi  
//tentukan jenis perintah , dan isi dengan select  
  
DataSet ds1 = new DataSet();  
SqlDataAdapter sda = new SqlDataAdapter(scom);
```

```
sda.Fill(ds1, "barang");  
dataGridBarang.DataSource = ds1;  
dataGridBarang.DataMember = "barang";  
//buat dataset, kemudian eksekusi perintah dengan SqlDataAdapter  
//kemudian fill data hasil eksekusi ke dalam dataset dengan member barang  
//kemudian set source dan member dari datagrid
```

Kemudian sekarang kita akan mencoba untuk menghapus data , berikut coding pada TxtDelete :

```
if (txtDelete.Text == " " )  
{  
    MessageBox.Show("isi data dengan kode barang");  
    goto berhenti;  
}  
  
int num;  
bool isNum = int.TryParse(txtDelete.Text.ToString(), out num);  
if (!isNum)  
{  
    MessageBox.Show("Data mesti number");  
    goto berhenti;  
}  
  
con.Open();  
SqlCommand scom = new SqlCommand();  
scom.Connection = con;  
scom.CommandType = CommandType.Text;  
scom.CommandText = " delete from MsBarang where KdBarang = '" +  
txtDelete.Text + "'";  
scom.ExecuteNonQuery();  
con.Close();  
btnShow.PerformClick();  
  
berhenti:  
Console.WriteLine("STOP");
```

Berikut Hasil Akhir Dari Aplikasi :

C# Dengan SQL 2005

Show / Insert / Delete Data Credit

	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

Nama Barang: Gelas

Stok: 200

Harga Barang: 5000

Show Data

Find Data By Kode Barang: 1

Search Data By Nama Barang: gelas

Delete Data: 2

Insert Data Reset Data

Part 6 –SQL 2005 Stored Procedure

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://suryodesign.asia>

www.suryodesign.wordpress.com

Lisensi Dokumen:

Copyright © 2003-2007 IlmuKomputer.Com

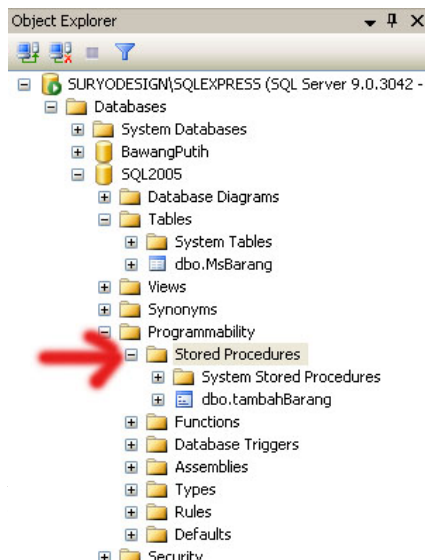
Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Pendahuluan

Pada pertemuan sebelumnya saya sudah menjelaskan bagaimana menghubungkan database SQL2005 dengan bahasa pemrograman C# , baik dari search , insert , update , dan delete dan kali ini saya akan mencoba untuk membahas salah satu kelebihan SQL yaitu STORED PROCEDURE, apa itu Store Procedure ? Stored procedure merupakan bagian SQL yang telah disiapkan oleh SQL Server atau bisa kita bilang adalah Suatu statement yang sebelumnya telah kita buat pada SQL Server, sebelumnya kita definisikan pada SQL sesuai kebutuhan kita , kemudian kita tinggal memanggil nama Stored Procedure itu pada bahasa pemrograman kita.

Saya akan mencoba untuk mengulas secara singkat dan jelas mengenai cara pembuatan Stored Procedure pada SQL 2005 dan cara penerapannya dalam aplikasi bagaimana melempar parameter itu sendiri ke dalam Stored Procedure dalam SQL 2005

Isi



Pertama Buka SQL 2005 Anda dan mari kita sama2 menuju ke direktori store procedure dan kemudian klik kanan pada directory Stored Procedure pada SQL 2005 dan pilih **Create New Stored Procedure**

Kemudian anda buat Stored Procedure Untuk menambah barang pada database , tetapi bila pada database terdapat nama barang yang sama dengan nama database yang kita input maka kita akan melakukan proses update berdasarkan barang yang ada dengan data inputan yang baru.

```
create procedure tambahBarang
@kode int ,@nama varchar(30) , @stok int ,@harga numeric(10)
as
if exists(select * from msbarang where NamaBarang = @nama )
update msbarang
set NamaBarang = @nama , Stok = @Stok , Harga = @Harga
where NamaBarang = @nama
else
insert into msbarang(NamaBarang,Stok,Harga) values(@nama,@stok,@harga)
```

Bagaimana menurut anda syntax di atas ? syntax di atas mudah sekali untuk dipahami karena ini hanya gabungan dari beberapa Syntax SQL dasar saja untuk insert , update , dan cek data.,

Kemudian setelah membuat STORED PROCEDURE pada SQL 2005 kita akan membuat program untuk insert data melalui C# ,
Berikut Contoh Stored Procedure pada C# yang saya terapkan pada tombol insert data

```
con.Open();
//membuka koneksi
SqlCommand scom = new SqlCommand();
//membuat SQLCOMMAND dengan alias scom
scom.Connection = con;
//include koneksi ke dalam sqlcommand
scom.CommandText = "tambahBarang";
scom.CommandType = CommandType.StoredProcedure;
//menentukan jenis command , defaultnya adalah
commandtype.text

SqlParameter nama = new SqlParameter("@nama",
SqlDbType.VarChar);
SqlParameter stok = new SqlParameter("@stok",
SqlDbType.Int);
SqlParameter harga = new SqlParameter("@harga",
SqlDbType.Int);
//buat sql parameter dengan nama variabel yg sama pada SQL
Stored Procedure yang telah kita buat
nama.Value = txtNama.Text;
stok.Value = txtStok.Text;
harga.Value = txtHarga.Text;
//isi variabel sql parameter tadi dengan nilai yang telah kita
input dari textboxt
scom.Parameters.Add(nama);
scom.Parameters.Add(stok);
scom.Parameters.Add(harga);
//kemudian add parameter yang telah kita buat tadi kedalam
command yang ada
scom.ExecuteNonQuery();
//jalankan storedProcedure
con.Close();
```

```

//akhiri koneksi ( tutup koneksi )
btnShow.PerformClick();
//jalankan btnshow untuk refresh data
berhenti:
//cari data pada dataset dengan table barang pada dataset
//kemudian tentukan primary key pada table tersebut
//dan cari berdasarkan primary key tersebut

txtNama.Text = (string)rw1["NamaBarang"];
txtStok.Text = Convert.ToString(rw1["Stok"]);
txtHarga.Text = Convert.ToString(rw1["harga"]);
//tampilkan data yang didapat ke dalam textbox

selesai:
;
}

```

Berikut Hasil Akhir Dari Aplikasi :

The screenshot shows a Windows application window titled "C# Dengan SQL 2005". At the top, there are tabs for "Show / Insert / Delete Data" and "Credit". Below the tabs is a table with the following data:

	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

Below the table, there are input fields and buttons:

- Input fields for "Nama Barang" (containing "Gelas"), "Stok" (containing "200"), and "Harga Barang" (containing "5000").
- Buttons: "Show Data", "Find Data By Kode Barang", "Search Data By Nama Barang", "Delete Data", "Insert Data", and "Reset Data".
- On the right side, there are three input fields corresponding to the buttons: "1" for "Find Data By Kode Barang", "gelas" for "Search Data By Nama Barang", and "2" for "Delete Data".

REFERENSI :

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

Part 7 – SQL 2005 Trigger

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://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 7 ini saya akan membahas tentang Trigger pada SQL 2005 , masi banyak orang yang awam akan feature-feature yang dimiliki oleh SQL sendiri , kebanyakan orang menganggap bahwa database hanya sebagai tempat menyimpan data , dan mereka hanya menganggap database dengan 3 gambaran simple dalam penggunaanya seperti, create table , retrieve data (mengambil / menampilkan data) dan manipulasi data (insert , update delete) , padahal banyak sekali fitur yang perlu kita oleh pada SQL 2005 ini seperti pengaturan Constraint yang ada sesuai kebutuhan system , Function , Stored Procedure , View , Index , Statistic , Trigger , Locking Transaction dan banyak lainnya.

Untuk menjadi seorang DBA (Database Administrator) ini merupakan hal dasar yang harus dipelajari , setelah memahami konsep ini , kemudian baru kita menganalisis kebutuhan database kita dan menghitung estimasi dan baru mengambil keputusan database apa yang mesti digunakan di pertimbangkan dengan factor seperti size , cost , fitur , license , dan system yang digunakan untuk menjadi server database itu sendiri.

Isi

Ayo Teriaklah “SEMANGAT” sebelum anda memulai membaca isi dari database ini :) , Oke sekarang langsung saja , apa itu Trigger ? Trigger artinya adalah **pemicu** , atau bahasa kasarnya dapat kita bilang sebagai syntax SQL yang dijalankan ketika ada proses yang di picu , proses disini adalah proses pada Query database , baik untuk retrieve data (show data) atau manipulation data(insert , update , delete) atau kebanyakan orang IT menyebutnya sebagai proses yang di lakukan ketika ada DDL (Data Definition Language dan DML (Data Manipulation Language).

1

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2007 IlmuKomputer.Com

Trigger ini terbagi menjadi 3 yaitu **Before** , **After** , dan **Instead Of**.

Kebanyakan yang umum di pakai adalah Before dan After , maksudnya Trigger Before adalah Trigger yang di lakukan sebelum proses insert/update/delete dilakukan, sedangkan After Trigger adalah Trigger yang di lakukan setelah proses insert/update/delete dilakukan.

Perlu diketahui bahwa dalam sebuah tabel dapat mempunyai beberapa trigger.

Trigger sangat berguna karena dapat secara otomatis dilaksanakan dalam server tanpa perlu kita ketik kembali sehingga dapat menyederhanakan Query yang kita miliki, sekaligus dapat menjaga konsistensi informasi dalam database . Trigger ini pun sudah disupport pada beberapa DBMS yang ada seperti pada SQL Server, MySQL, PostgreSQL, ataupun Oracle ,

Berikut adalah beberapa keuntungan trigger :

- Konsistensi Database , maksudna trigger digunakan terhadap table yang menjadi reference , dimana semua user / admin yang melakukan proses pada table tersebut akan menghasilkan proses yang sama pada database dan dapat menghindari kesalahan yang terjadi karena perbedaan hasil yang ada.
- Mudah Untuk Di Update , maksudnya Trigger yang telah kita buat pada database yang ada dapat kita ubah sewaktu – waktu tanpa mengubah coding dalam aplikasi , dan dengan mengubah trigger ini , maka semua user / client yang ada akan menggunakan trigger yang ada
- Kecepatan. Tentu saja karena trigger yang tersimpan dalam sisi database dan bukan pada sisi client dapat mengurangi bandwitch atau lalu lintas yang ada pada jaringan dan hal ini menyebabkan system kita lebih efisien dalam digunakan.
- Mengurangi Pekerjaan Seorang programmer yang membuat aplikasi dalam mengatur proses yang ada karena proses ini sudah di miliki oleh trigger

Berikut Merupakan Syntax Trigger pada [MSDN](#) :

```
CREATE TRIGGER trigger_name
ON { table | view }
[ WITH ENCRYPTION ]
{
    { { FOR | AFTER | INSTEAD OF } { [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
      [ WITH APPEND ]
      [ NOT FOR REPLICATION ]
      AS
      [ { IF UPDATE ( column )
        [ { AND | OR } UPDATE ( column ) ]
        [ ...n ]
      | IF ( COLUMNS_UPDATED ( ) { bitwise_operator } updated_bitmask )
        { comparison_operator } column_bitmask [ ...n ]
      } ]
      sql_statement [ ...n ]
    }
}
```

Pertama Mari Kita membuat Table **MsBarang** dan **TrPenerimaan** Seperti berikut pada SQL2005 :

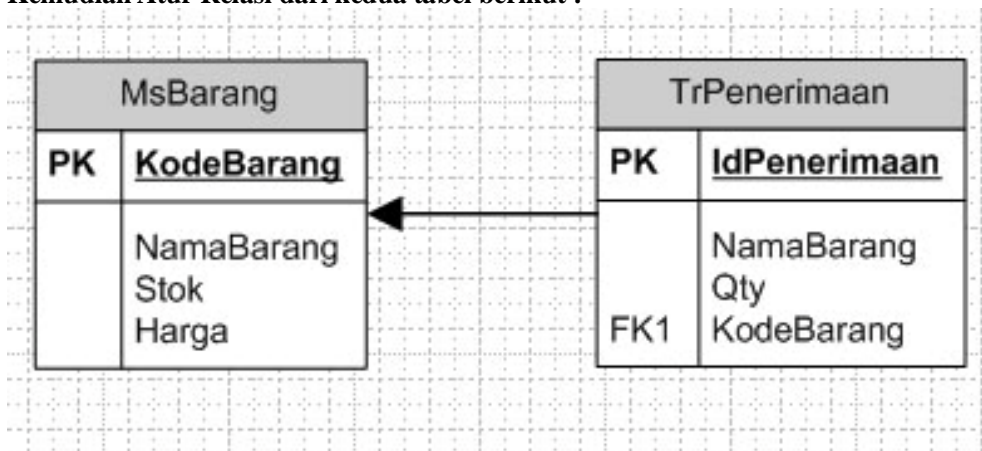
MSBARANG :

SURYODESIGN-PC\...bo.TrPenerimaan SURYODESIGN-PC			
Column Name	Data Type	Allow Nulls	
KdBarang	int	<input type="checkbox"/>	
NamaBarang	varchar(30)	<input checked="" type="checkbox"/>	
Stok	int	<input checked="" type="checkbox"/>	
Harga	numeric(10, 0)	<input checked="" type="checkbox"/>	

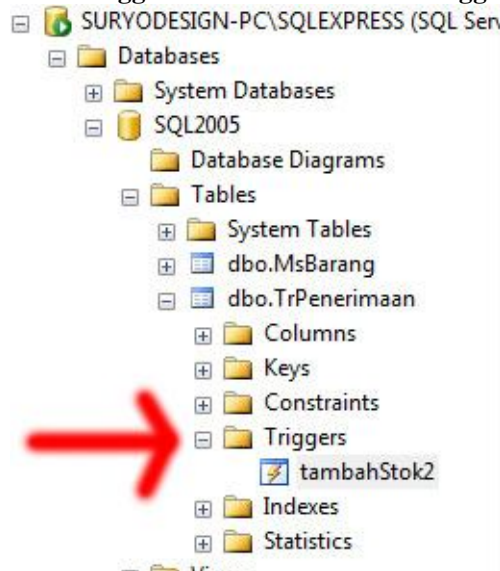
TRPENERIMAAN :

SURYODESIGN-PC\...bo.TrPenerimaan SURYODESIGN-PC			
Column Name	Data Type	Allow Nulls	
IdPenerimaan	int	<input type="checkbox"/>	
NamaBarang	varchar(30)	<input checked="" type="checkbox"/>	
Qty	int	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Kemudian Atur Relasi dari kedua tabel berikut :



Oke Sekarang Saatnya kita membuat Trigger pada SQL 2005 , Pilih Tabel **MsBarang** dan Cari Menu **Trigger** Dan Klik Kanan **New Trigger**



Kemudian Kita Buat Proses Trigger Seperti Berikut :

```
USE [SQL2005]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create trigger [dbo].[tambahStok2]
on [dbo].[TrPenerimaan]
for INSERT
as
begin
SET NOCOUNT ON
declare @NamaBarang varchar(30) , @Qty int
select @NamaBarang = NamaBarang , @Qty = Qty from TrPenerimaan
update MsBarang set Stok = (STok + @Qty) where NamaBarang = @Namabarang
end
```

Saya Akan menjelaskan secara singkat pada syntax di atas

1. Gunakan Database SQL 2005
2. Create Trigger dengan Instance [dbo] dan nama trigger [tambahStok2]
3. Trigger dilakukan Ketika ada Proses insert
4. Deklarasikan variable dan type yang digunakan
5. Isi Variabel yang kita gunakan sesuai dengan variable yang kita input
6. Lakukan Proses Trigger Update pada table MsBarang untuk menambah jumlah Stok.

Setelah Proses Trigger Selesai maka kita mesti pastikan proses trigger kita jalan sebelum kita implementasikan pada aplikasi. Coba tambahkan Penerimaan dengan proses penerimaan barang dengan nama yang ada pada table MsBarang dan pastikan proses tersebut success untuk diinsert dan kemudian lihat table msBarang

```
insert into TrPenerimaan(NamaBarang,Qty) values('Gelas','500000')
```

Pastikan bahwa Stok pada MsBarang bertambah sesuai dengan penerimaan

```
select * from TrPenerimaan  
select * from MsBarang
```

Setelah Semua Proses Jalan maka sekarang kita akan mencoba meng-implementasikannya pada C#.Net , aplikasi ini merupakan aplikasi lanjutan dari Pertemuan 6 , bila anda belum memiliki aplikasi pada pertemuan 6 anda dapat mendownloadnya disini [http://www.suryodesign.asia/freedownload/C_Sharp_Part6%20-%20SQL2005%20Stored%20Procedure%20\(2\).zip](http://www.suryodesign.asia/freedownload/C_Sharp_Part6%20-%20SQL2005%20Stored%20Procedure%20(2).zip)

sekarang mari kita design aplikasi dengan Form Berikut , gunakan komponen DataGridView, ComboBox , TextBox , Button , Label , dan SplitContainer

Setelah Proses Design Aplikasi Selesai Sekarang kita akan mencoba untuk meload Nama Barang Ke Dalam Combobox , disini saya mempraktikkan 2 Cara untuk meload ke dalam Combo Box , ada banyak cara seperti binding , gunakan datasource , wizard dan manual additem. Tapi disini saya menerapkan 2 Cara yang simple saja , anda dapat berkreasi sendiri sesuai dengan logika anda untuk mengasah kemampuan anda sendiri :

Cara 1 Load Data ke dalam ComboBox

```
SqlCommand scom = new SqlCommand();
scom.Connection = con;
scom.CommandType = CommandType.Text;
scom.CommandText = "select NamaBarang from msbarang";

DataSet ds1 = new DataSet();
SqlDataAdapter sda = new SqlDataAdapter(scom);
sda.Fill(ds1, "barang");
cmbNamaBarang.DataSource = ds1.Tables["barang"] ;
cmbNamaBarang.DisplayMember = "NamaBarang";
```

Cara 2 Load Data ke Dalam ComboBox

```
SqlDataReader dr;
SqlCommand scom = new SqlCommand();
scom.Connection = con;
scom.CommandType = CommandType.Text;
scom.CommandText = "select * from msbarang";
con.Open();
DataSet ds1 = new DataSet();
SqlDataAdapter sda = new SqlDataAdapter(scom);
sda.Fill(ds1, "barang");
dr = scom.ExecuteReader();
while(dr.Read())
{
    cmbNamaBarang2.Items.Add(dr.GetValue(1));
}
cmbNamaBarang2.SelectedIndex = 0;
con.Close();
```

Setelah Proses Load data ke dalam Combo Box Selesai maka kita akan Load Data TrPenerimaan ke dalam DataGrid , berikut codingnya :

```
SqlCommand scom = new SqlCommand();
scom.Connection = con;
scom.CommandType = CommandType.Text;
scom.CommandText = "select * from trPenerimaan";

DataSet ds2 = new DataSet();
SqlDataAdapter sda = new SqlDataAdapter(scom);
sda.Fill(ds2, "transaksi");
dataGridTransaksi.DataSource = ds2;
dataGridTransaksi.DataMember = "transaksi";
```

Kemudian disini saya membuat dengan Split Container , dimana pada kolom kiri dan kanan ada sama2 tombol submit , anda dapat menggunakan yang mana saja untuk insert data , perbedaan pada cara insert data ke dalam combobox menyebabkan proses pengambilan data dari combobox itu sendiri dengan metode pengambilan yang berbeda.

Berikut Button Tambah data pada Left Container

```
string a = cmbNamaBarang.Text;
if (txtQty.Text == "")
//gunakan OR untuk argumen membandingkan
{
    MessageBox.Show("Qty mesti diisi");
    //Environment.Exit(-1);
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}

int num;
//buat variabel num
bool isNum = int.TryParse(txtQty.Text.ToString(), out num);
//membuat variabel is Num dan kemudian isi dari variabel isNum itu sendiri
//mengubah type data dan menyimpan hasilna pada variabel num
if (!isNum)
//mengecek nilai isNum apakah false
//( karena bukan number melainkan alfabet )
{
    MessageBox.Show("Qty mesti number");
    //bila false maka tampilkan pesan harga mesti number
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}

con.Open();
//open koneksi
SqlCommand scom = new SqlCommand();
//membuat SQLCOMMAND dengan alias scom
scom.Connection = con;
//include koneksi ke dalam sqlcommand
scom.CommandText = "insert into TrPenerimaan(NamaBarang,Qty) values('" +
a + "','" + txtQty.Text + "')";
scom.CommandType = CommandType.Text;
//menentukan jenis command , defaultnya adalah commandtype.text
scom.ExecuteNonQuery();
//jalankan storedProcedure
con.Close();
//akhiri koneksi ( tutup koneksi )
btnShowTr.PerformClick();
tabControl1.SelectedIndex = 0;
btnShow.PerformClick();
tabControl1.SelectedIndex = 1;
//jalankan btnshow untuk refresh data
berhenti:
    Console.WriteLine("STOP");
```

. Berikut Button Tambah data pada Right Container

```
string a = Convert.ToString(cmbNamaBarang2.SelectedItem);
if (txtQty2.Text == "")
//gunakan OR untuk argumen membandingkan
{
    MessageBox.Show("Qty mesti diisi");
    //Environment.Exit(-1);
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}
int num;
//buat variabel num
bool isNum = int.TryParse(txtQty2.Text.ToString(), out num);
//membuat variabel isNum dan kemudian isi dari variabel isNum itu sendiri
//mengubah type data dan menyimpan hasilna pada variabel num
if (!isNum)
//mengecek nilai isNum apakah false
//( karena bukan number melainkan alfabet )
{
    MessageBox.Show("Qty mesti number");
    //bila false maka tampilkan pesan harga mesti number
    goto berhenti;
    //Pindah Baris ke Berhenti dan skip baris selanjutnya
}
con.Open();
//open koneksi
SqlCommand scom = new SqlCommand();
//membuat SQLCOMMAND dengan alias scom
scom.Connection = con;
//include koneksi ke dalam sqlcommand
scom.CommandText = "insert into TrPenerimaan(NamaBarang,Qty) values('" +
a + "','" + txtQty2.Text + "')";
scom.CommandType = CommandType.Text;
//menentukan jenis command , defaultnya adalah commandtype.text

scom.ExecuteNonQuery();
//jalankan storedProcedure
con.Close();
//akhiri koneksi ( tutup koneksi )
btnShowTr.PerformClick();
tabControl1.SelectedIndex = 0;
btnShow.PerformClick();
tabControl1.SelectedIndex = 1;
//jalankan btnshow untuk refresh data
berhenti:
Console.WriteLine("STOP");
```

Untuk Aplikasi Secara lengkap anda dapat mendownloadnya di : www.suryodesign.asia/freedownload , sampai jumpat pada artikel selanjutnya dengan materi pembuatan program yang efisien dengan menggunakan class dan locking transaction untuk menghindari kesalahan yang terjadi.

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

Part 8 – Cryptography And Efficient Flow Code

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://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

Perkembangan Teknologi di Indonesia sangatlah pesat , teknologi menentukan sekali kemajuan di suatu perusahaan, baik di mulai dari proses bisnis dalam perusahaan itu sendiri atau iklan maupun advertising ataupun merupakan webprofile pada suatu halaman internet , dalam prosesnya sendiri lalu lintas informasi dalam dunia maya perlu dijaga , hal ini sangat penting , karena informasi tersebut merupakan informasi yang penting bagi suatu pihak tertentu baik dalam penerimaan maupun pengiriman proses informasi.

Dalam hal ini kita membutuhkan pengamanan , banyak hal yang di lakukan oleh programmer untuk mengamankan data yang ada mulai dari menggunakan SSL , SESSION , COOKIES , ataupun ENCRYPTION dan Kriptografi dan alternative sejenisnya. Dan bagaimana kita menerapkan nya di dalam pemrograman kita , disini saya akan mengajarkannya dan memberitahu bagaimana menerapkan inheritance dan menggunakannya di dalam class kita , sehingga berapa banyak nya form yang kita miliki kita tetap akan menggunakan 1 class untuk show data dan 1 class untuk insert apapun jenisnya ☺.

1

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2007 IlmuKomputer.Com

harap Sebelum membaca artikel ini anda telah mengerti mengenai konsep manipulation data pada SQL 2005 dan mengerti konsep dasar penggunaan Class dan inheritance yang telah saya jelaskan pada artikel sebelumnya , bagi yang belum mengerti dapat anda download di www.suryodesign.asia/freedownload , harap pelajari konsep2 pada artikel sebelumnya atau anda akan mengalami kesusahan atau bahkan tersesat pada artikel ini karena tidak mengerti konsep dan penggunaan class maupun cara passing parameter .

Isi

Disini saya akan membahas mengenai Kriptografi, Kriptografi disini berasal dari bahasa Yunani yaitu Kryptos yang berarti tersembunyi dan grafo yang berarti menulis, dan kemudian Kriptografi sendiri artinya merupakan teknik untuk menyamarkan atau membuat informasi / pesan yang telah telah diinput seolah olah tidak terlihat oleh pihak lain. Sekalipun mereka bisa mendapatkan informasi yang ada tetapi mereka tidak akan mendapatkan isi pesan yang asli dari message tersebut.

Kita menggunakan MD5 yang merupakan singkatan dari Message Digest 5 (MD-5) , disini nomor "5" merupakan versi perbaikan dari versi2 yang ada pada sebelumnya , MD5 di buat oleh Ron Rivest pada tahun 1991 , untuk menggantikan hash function sebelumnya yaitu MD4 yang dibuat pada tahun 1991 dan kemudian pada tahun 1996 kemudian ditemukan MD5 , dan kemudian pada tahun 2007 dilakukan penelitian oleh Arjen Lestra untuk membuat SSL Sertificate validate untuk menambah atau mensupport keamanan pada MD5 ini .

Apa itu MD5 ?

MD5 adalah teknik / formula yang mengubah kalimat yang ada dan kemudian di hashing menjadi satu arah yang menghasilkan hash 128 bit "finger print" atau "message digest" yang maksudnya secara langsung di konversi dan tidak meninggalkan jejak sehingga tidak dapat di konversi kembali ke dalam bentuk aslinya. MD5 merupakan salah satu cara untuk mengubah data integrity yang berguna untuk menyimpan password atau data yang sensitive atau hanya boleh digunakan oleh beberapa pihak saja.

Apakah MD5 merupakan enkripsi ?

Ada beberapa yg bilang MD5 merupakan teknik Enkripsi, tetapi ada yg bilang jg bukan , hal ini merupakan suatu perbedaan pendapat dimana satu pihak mengatakan bahwa yang namanya enkripsi mesti dapat di dekripsi untuk mendapatkan value / nilai aslinya , dan ada yg mengatakan ini hanyalah jalan satu arah untuk mengacak atau mengubah kalimat yang ada dan tidak meninggalkan kalimat aslinya dan kita tidak dapat menggunakan md5sum untuk mendapat kembali kalimat aslinya.

Okay , sekarang kita memiliki kalimat seperti password yang telah di konversi, dapatkan kita mendapatkannya isi kalimat original tersebut dengan cara seperti Brute Force ?

seperti system password yang ada , kita dapat melakukan brute force yang ada , tetapi bagaimanapun MD5 sum merupakan 128 bit space yang artinya kita membutuhkan 2^{128} kemungkinan untuk mendapatkan aslinya, bias dibutuhkan waktu yang lama sekali untuk kalimat yang diinput merupakan gabungan number dan alphabet yang dibedakan huruf besar atau kecilna dan makin panjang kalimatnya pun mempengaruhi lamanya hasil pencarian.

Okay. Tetapi bagaimana bila ada MD5 dictionary yang dibuat oleh seseorang untuk memecahkan MD5 yang kita miliki , dan mengetahui password MD5 aslinya ?

ini memungkinkan saja , tetapi ini membutuhkan waktu yang lama juga karena dictionary ini sendiri terdiri dari alphet (huruf kecil dan besar) dan juga number yang berarti ada kemungkinan 46,656,000,000 entries data dan semua terdiri dari 32 characters setiap kalimat entriesnya (jangan lupa karena MD5 memiliki panjang 32 karakter untuk hasil output nya). Ini membutuhkan sekitar 1 Terabyte data untuk menyimpan dan entah dibutuhkan waktu berapa lama untuk mencarinya.

Berikut adalah step by step MD5 Alghotritm yang saya dapat dari narasumber pada salah satu situs yang membahas tentang Cryptography , sumber ini saya biarkan dalam bentuk original agar tidak terjadi kesalahan dalam translate nya .

MD5 Algorithm Overview

MD5 algorithm is well described in RFC 1321 - The MD5 Message-Digest Algorithm, see <http://www.ietf.org/rfc/rfc1321.txt>. Below is a quick overview of the algorithm.

MD5 algorithm consists of 5 steps:

Step 1. Appending Padding Bits. The original message is "padded" (extended) so that its length (in bits) is congruent to 448, modulo 512. The padding rules are:

- The original message is always padded with one bit "1" first.
- Then zero or more bits "0" are padded to bring the length of the message up to 64 bits fewer than a multiple of 512.

Step 2. Appending Length. 64 bits are appended to the end of the padded message to indicate the length of the original message in bytes. The rules of appending length are:

- The length of the original message in bytes is converted to its binary format of 64 bits. If overflow happens, only the low-order 64 bits are used.
- Break the 64-bit length into 2 words (32 bits each).

- The low-order word is appended first and followed by the high-order word.

Step 3. Initializing MD Buffer. MD5 algorithm requires a 128-bit buffer with a specific initial value. The rules of initializing buffer are:

- The buffer is divided into 4 words (32 bits each), named as A, B, C, and D.
- Word A is initialized to: 0x67452301.
- Word B is initialized to: 0xEFCDAB89.
- Word C is initialized to: 0x98BADCFE.
- Word D is initialized to: 0x10325476.

Step 4. Processing Message in 512-bit Blocks. This is the main step of MD 5 algorithm, which loops through the padded and appended message in blocks of 512 bits each. For each input block, 4 rounds of operations are performed with 16 operations in each round. This step can be described in the following pseudo code slightly modified from the RFC 1321's version:

Input and predefined functions:

A, B, C, D: initialized buffer words

$F(X,Y,Z) = (X \text{ AND } Y) \text{ OR } (\text{NOT } X \text{ AND } Z)$

$G(X,Y,Z) = (X \text{ AND } Z) \text{ OR } (Y \text{ AND } \text{NOT } Z)$

$H(X,Y,Z) = X \text{ XOR } Y \text{ XOR } Z$

$I(X,Y,Z) = Y \text{ XOR } (X \text{ OR } \text{NOT } Z)$

T[1, 2, ..., 64]: Array of special constants (32-bit integers) as:

$T[i] = \text{int}(\text{abs}(\sin(i)) * 2^{32})$

M[1, 2, ..., N]: Blocks of the padded and appended message

R1(a,b,c,d,X,s,i): Round 1 operation defined as:

$a = b + ((a + F(b,c,d) + X + T[i]) \lll s)$

R2(a,b,c,d,X,s,i): Round 1 operation defined as:

$a = b + ((a + G(b,c,d) + X + T[i]) \lll s)$

R3(a,b,c,d,X,s,i): Round 1 operation defined as:

$a = b + ((a + H(b,c,d) + X + T[i]) \lll s)$

R4(a,b,c,d,X,s,i): Round 1 operation defined as:

$a = b + ((a + I(b,c,d) + X + T[i]) \lll s)$

Algorithm:

For k = 1 to N do the following

AA = A

BB = B

CC = C

DD = D

(X[0], X[1], ..., X[15]) = M[k] /* Divide M[k] into 16 words */

/* Round 1. Do 16 operations. */

R1(A,B,C,D,X[0], 7, 1)

R1(D,A,B,C,X[1],12, 2)

R1(C,D,A,B,X[2],17, 3)

```
R1(B,C,D,A,X[ 3],22, 4)
R1(A,B,C,D,X[ 4], 7, 5)
R1(D,A,B,C,X[ 5],12, 6)
R1(C,D,A,B,X[ 6],17, 7)
R1(B,C,D,A,X[ 7],22, 8)
R1(A,B,C,D,X[ 8], 7, 9)
R1(D,A,B,C,X[ 9],12,10)
R1(C,D,A,B,X[10],17,11)
R1(B,C,D,A,X[11],22,12)
R1(A,B,C,D,X[12], 7,13)
R1(D,A,B,C,X[13],12,14)
R1(C,D,A,B,X[14],17,15)
R1(B,C,D,A,X[15],22,16)

/* Round 2. Do 16 operations. */
R2(A,B,C,D,X[ 1], 5,17)
R2(D,A,B,C,X[ 6], 9,18)
R2(C,D,A,B,X[11],14,19)
R2(B,C,D,A,X[ 0],20,20)
R2(A,B,C,D,X[ 5], 5,21)
R2(D,A,B,C,X[10], 9,22)
R2(C,D,A,B,X[15],14,23)
R2(B,C,D,A,X[ 4],20,24)
R2(A,B,C,D,X[ 9], 5,25)
R2(D,A,B,C,X[14], 9,26)
R2(C,D,A,B,X[ 3],14,27)
R2(B,C,D,A,X[ 8],20,28)
R2(A,B,C,D,X[13], 5,29)
R2(D,A,B,C,X[ 2], 9,30)
R2(C,D,A,B,X[ 7],14,31)
R2(B,C,D,A,X[12],20,32)

/* Round 3. Do 16 operations. */
R3(A,B,C,D,X[ 5], 4,33)
R3(D,A,B,C,X[ 8],11,34)
R3(C,D,A,B,X[11],16,35)
R3(B,C,D,A,X[14],23,36)
R3(A,B,C,D,X[ 1], 4,37)
R3(D,A,B,C,X[ 4],11,38)
R3(C,D,A,B,X[ 7],16,39)
R3(B,C,D,A,X[10],23,40)
R3(A,B,C,D,X[13], 4,41)
R3(D,A,B,C,X[ 0],11,42)
R3(C,D,A,B,X[ 3],16,43)
R3(B,C,D,A,X[ 6],23,44)
R3(A,B,C,D,X[ 9], 4,45)
R3(D,A,B,C,X[12],11,46)
R3(C,D,A,B,X[15],16,47)
R3(B,C,D,A,X[ 2],23,48)

/* Round 4. Do 16 operations. */
R4(A,B,C,D,X[ 0], 6,49)
R4(D,A,B,C,X[ 7],10,50)
R4(C,D,A,B,X[14],15,51)
R4(B,C,D,A,X[ 5],21,52)
R4(A,B,C,D,X[12], 6,53)
```

```
R4(D,A,B,C,X[ 3],10,54)
R4(C,D,A,B,X[10],15,55)
R4(B,C,D,A,X[ 1],21,56)
R4(A,B,C,D,X[ 8], 6,57)
R4(D,A,B,C,X[15],10,58)
R4(C,D,A,B,X[ 6],15,59)
R4(B,C,D,A,X[13],21,60)
R4(A,B,C,D,X[ 4], 6,61)
R4(D,A,B,C,X[11],10,62)
R4(C,D,A,B,X[ 2],15,63)
R4(B,C,D,A,X[ 9],21,64)
```

```
A = A + AA
B = B + BB
C = C + CC
D = D + DD
End of for loop
```

Output:

A, B, C, D: Message digest

Step 5. Output. The contents in buffer words A, B, C, D are returned in sequence with low-order byte first.

Sekarang langsung saja masuk ke penerapan Cryptography tersebut ke dalam aplikasi. Pertama buat database MsUser pada SQL2005 dengan struktur berikut :

SURYODESIGN-PC\...05 - dbo.MsUser				SURYODESIGN-PC			
Column Name		Data Type		Allow Nulls			
Username		varchar(20)		<input type="checkbox"/>			
PassMD5		varchar(35)		<input checked="" type="checkbox"/>			
PassSHA1		varchar(35)		<input checked="" type="checkbox"/>			
				<input type="checkbox"/>			

Setelah anda membuatnya maka kemudian design form input user dan login seperti berikut , aplikasi ini merupakan aplikasi lanjutan dari pertemuan 7 , anda dapat download di web saya bila belum memilikinya di www.suryodesign.asia .

Design Form Input User seperti berikut :

Username	PassMD5	PassSHA1
adli	58c1a701fa3ac19ede3b2699b1a73a7e	/yWAS62pzUyF/JoHp/FdI3mpGk=
mizone	b2348f469a472692555470f6ade2a2ef	HV8q+0pg0x3Kuhcf/DiYBIDRmpk=
suryo	98d006db0378032df287e4bb6913a13c	wRLAldCoyqRUywuSC3CUB+JNyY=

Username

Password 1

Password 2

Kemudian design Form MenuLogin seperti berikut :

Menu Login C#

Username

Password

Encryption



Kemudian Pada Class2005.Cs tambahkan class induk 2005.cs yang akan kita inherits atau turunkan pada class anaknya , jangan lupa mengenai konsep setter dan getter yang berguna untuk menangkap nilai dan men set nilai , dan jgn lupa jg mengenai konsep static , nonstatic , serta penggunaan public atau private.

```
public class ClassInduk2005
{
    public string prefixNama = "suryo." ;
    public static int _flagLogin = 0;
    public static int flagLogin
    {
        get { return _flagLogin; }
        set { _flagLogin = value; }
    }

    public static string _Nama;
    public static string Nama
    {
        get { return _Nama; }
        set { _Nama = value; }
    }
}
```

Kemudian turunkan pada Class2005 kita dengan menambahkan titik dua (:) atau colon seperti berikut pada Class2005 yang kita miliki pada awalnya :

```
public class Class2005 : ClassInduk2005
```

Setelah ini dilakukan maka kita dapat memanggil baseclass dimana tempat variable di inheritkan / diturunkan kepada form anak.

Kemudian Deklarasi variable yang kita gunakan untuk menampilkan serta insert dan menampung data pada Class2005.cs

```
private static SqlConnection con = new SqlConnection(@"Data
Source=SURYODESIGN-PC\SQLEXPRESS;Initial Catalog=SQL2005;Integrated Security=True");
private static SqlCommand com = new SqlCommand();
private static SqlDataAdapter sda = new SqlDataAdapter();
private static SqlDataReader sdr;
private static DataSet _ds ;

public static DataSet ds
{
    get { return _ds; }
    set { _ds = value; }
}
```



```
private static string _query;
public static string query
{
    get { return _query; }
    set { _query = value; }
}

private static string _Message;
public static string Message
{
    get { return _Message; }
    set { _Message = value; }
}
```

Setelah mendeklarasikan variable diatas sekarang saatnya kita membuat setter dan getter untuk enkripsi MD5 dan SHA1 yang telah diinput,dan kita tidak langsung mengacak input yang ada , tetapi kita menambahkan sedikit ramuan dari input user digabung dengan “suryo.”, kita menambahkan ini sebagai KeyReference agar hasil enkripsi kita tidak sepenuhnya merupakan inputan user asli , tetapi sebenarnya ada KeyReference yang disisipkan disana, anda dapat lebih creative dengan cara anda sendiri , berikut contoh enkripsi MD5:

```
private static string _stringMD5 ;
public static string stringMD5
{
    get
    {
        //String kosong yang digunakan menampung enkripsi
        string ret = String.Empty;
        //deklarasikan cryptography
        MD5CryptoServiceProvider md5Hasher = new MD5CryptoServiceProvider();
        //dapatkan byte data
        byte[] data = System.Text.Encoding.ASCII.GetBytes("suryo." +
            _stringMD5);
        //Encrypt
        data = md5Hasher.ComputeHash(data);
        //convert data dari byte menjadi hex
        for (int i = 0; i < data.Length; i++)
        {
            ret += data[i].ToString("x2").ToLower();
        }
        //Return encoded
    }
}
```

```
return ret;
}
set { _stringMD5 = value ; }
}
```

Berikut Enkripsi untuk SHA1 :

```
private static string _stringSHA1;
public static string stringSHA1
{
    get{
        string rethash = "";
        System.Security.Cryptography.SHA1 hash
        = System.Security.Cryptography.SHA1.Create();
        System.Text.ASCIIEncoding encoder = new System.Text.ASCIIEncoding();
        byte[] combined = encoder.GetBytes("suryo." + _stringSHA1);
        hash.ComputeHash(combined);
        rethash = Convert.ToBase64String(hash.Hash);
        return rethash;
    }
}
set { _stringSHA1 = value; }
}
```

Setelah itu selesai , maka untuk aplikasi kita , kita akan menggunakan 1 class saja untuk semua data yang tampil dan semua proses insert yang ada seperti berikut :
Untuk Show Data :

```
public static void tampilData(ref string namaTable)
{
    com.Connection = con;
    com.CommandType = CommandType.Text;
    com.CommandText = _query;
    _ds = new DataSet();
    SqlDataAdapter sda = new SqlDataAdapter(com);
    sda.Fill(_ds, namaTable);
}
```

Setelah kita deklarasikan ini , di aplikasi kita untuk menampilkan cukup dengan

```
dataGridBarang.DataSource = Class2005.ds;          dataGridBarang.DataMember = a;
```

Kita tidak perlu Return Dataset dan sejenisnya karena kita dapat langsung meng-aksesnya melewati class2005 . banyak cara / modifikasi dari source yang dapat dilakukan , ini tergantung dari programmer itu sendiri lebih menyukai cara yang mana.

Berikut Coding untuk Insert Data :

```
public static void insertData(ref string perintahQuery )
{
    con.Open();
    com.Connection = con;
    com.CommandType = CommandType.Text;
    com.CommandText = perintahQuery;
    com.ExecuteNonQuery();
    con.Close();
}
```

Dan Untuk Memanggilnya dari form insert data, ada beberapa hal yg perlu kita lakukan yaitu definisikan class , konversi password ke dalam md5 dan SHA1 kemudian tentukan table untuk dataset dan baru kemudian insert dan refresh table yang ada seperti berikut :

```
ClassInduk2005 ClassAnak2005 = new ClassInduk2005();
string enkripMd5 = Class2005.stringMD5 = txtPass1.Text;
string enkripSha1 = Class2005.stringSHA1 = txtPass2.Text;
string query = "insert into MsUser values('" + txtUsername.Text + "','" + Class2005.stringMD5 + "','" + Class2005.stringSHA1 + "')";

Class2005.insertData(ref query);
Class2005.query = "select * from msuser";
string a = "user";
Class2005.tampilData(ref a);
dataGridUser.DataSource = Class2005.ds;
dataGridUser.DataMember = a;
```

Setelah Insert Data Sukses saat nya kita membahas mengenai Login, berikut isi classlogin pada class2005.cs pada aplikasi :

```
public static void loginData(ref string txtUsername,ref string txtPass, ref int indexNo)
{
    stringMD5 = txtPass;
    stringSHA1 = txtPass;
```

```
if (con.State != ConnectionState.Open)
{
    con.Open();
}
com.Connection = con;
com.CommandType = CommandType.Text;
if (indexNo == 0)
{
    com.CommandText = "select * from msuser where Username = " + txtUsername
    + " and passMD5 =" + stringMD5 + "";
}
else
{
    com.CommandText = "select * from msuser where Username = " + txtUsername
    + " and PassSHA1 =" + stringSHA1 + "";
}
SqlDataReader sdr = com.ExecuteReader();
if (sdr.HasRows)
{
    Class2005.flagLogin = 1;
}
else
{
    Class2005.flagLogin = 0;
}
if (con.State == ConnectionState.Open)
{
    con.Close();
}
}
```

Kemudian Cara Kita Login dari Form Login aplikasi kita seperti berikut :

```
string txtPass = txtPassLogin.Text;
int indexNo = cmbEnkripsi.SelectedIndex ;
string userLogin = txtUserLogin.Text;
Class2005.loginData(ref userLogin, ref txtPass, ref indexNo);
if (Class2005.flagLogin > 0 )
{
    Form1 form1 = new Form1();
    form1.Show();
    this.Hide();
}
}
```

Part 9 – BindingSource

M.Suryo Pranoto

suryodesign@yahoo.co.id

<http://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

Banyak Cara Yang di lakukan untuk menampilkan dan mengontrol data pada aplikasi User Interface yang ada saat ini , masing-masing programmer memiliki tehnik dan cara masing-masing untuk membuat aplikasi yang mana menurut mereka paling efektif dan terbaik untuk digunakan oleh User. Disini saya memberikan dasar cara menggunakan BindingSource untuk mengontrol dan membuat navigasi data.

harap Sebelum membaca artikel ini anda telah mengerti mengenai konsep menampilkan data pada SQL 2005 yang telah saya jelaskan pada artikel sebelumnya , bagi yang belum mengerti dapat anda download di www.suryodesign.asia/freedownload atau di <http://ilmukomputer.org/2009/01/06/suryodesign/> , harap pelajari konsep2 pada artikel sebelumnya atau anda akan mengalami kesusahan atau bahkan tersesat pada artikel ini karena tidak mengerti konsep dan yang ada.

1

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2007 IlmuKomputer.Com

Isi

BindingSource komponen biasa kita gunakan dalam navigasi data , bindingSource sendiri memiliki banyak tujuan , pertama agar memudahkan binding control dalam mengendalikan data dalam form, memberikan notifikasi / peringatan atas perubahan yang terjadi dan services lainnya antara windows form controls dan data source. Untuk menggunakan bindingSource ini kita dapat Attach Binding Source komponen kedalam data source dengan menggunakan DataSource property. Untuk Binding yang complex. Kita dapat memilih menggunakan DataMember property untuk beberapa kolom atau list pada data source, kemudian anda dapat menempelkan control tersebut pada BindingSource.

Kemudian untuk navigasi data kita dapat menggunakan MoveNext, MoveLast, MoveFirst, MovePrevious dan kita dapat sorting ataupun filter data yang ada. Kita biasa menggunakan bindingsource untuk mengontrol data. Bindingsource bukan hanya powerfull tool dan tidak hanya untuk binding tetapi dapat dengan cepat untuk memfilter dan sorting data. Kita dapat menggunakan Filter property untuk mem-filter data.

```
//filter data berdasarkan kolom namabarang  
bS.Filter = "NamaBarang like '%" + txtFilter.Text + "%'";
```

dimana bS adalah BindingSource yang telah kita buat dan NamaBarang adalah Kolom Table dan txtFilter merupakan textbox yang kita miliki.

Kita dapat juga menggunakan untuk membandingkan nilai seperti berikut

```
bS.Filter = "Qty >= 50 AND Order_ID <= 20"
```

Sekarang mari kita design form seperti berikut :

KdBarang	NamaBarang	Stok	Harga
1	Gelas	5364	5000
2	Piring	5	7500
3	Sendok	5	3000
4	Mangkok	51	10000
10	s	5	222
11	s2	5	222

<< < > >> Reload Binding

Kode Barang: 1
Nama Barang: Gelas
Stok: 5364
Harga: 5000

Filter:
Filter Data Cancel Filter

Hal pertama yang harus kita lakukan adalah deklarasi global SqlConnection , DataTable dan BindingSource seperti berikut

```
//buat sqlconnection , datatable , dan bindingsource
SqlConnection con = new SqlConnection(@"Data
Source=SURYODESIGN-PC\SQLEXPRESS;Initial
Security=True");
Catalog=SQL2005;Integrated
DataTable dt = new DataTable();
BindingSource bs = new BindingSource();
```

Kemudian selanjutnya kita isi button ReloadBinding untuk meload data ke dalam dataGrid

```
//clear data table
dt.Clear();
//clear databindings pada textbox
txtNama.DataBindings.Clear();
txtStok.DataBindings.Clear();
txtHarga.DataBindings.Clear();
txtKd.DataBindings.Clear();

//buat data adapter dan sqlbuilder
SqlDataAdapter da = new SqlDataAdapter("select * from msbarang", con);
SqlCommandBuilder scb = new SqlCommandBuilder(da);
//fill data ke dalam datatable
da.Fill(dt);
//set binding source datasource dan tampilkan ke dalam grid
bS.DataSource = dt;
dataGridView1.DataSource = bS;

//binding data ke dalam textbox yg ada
txtNama.DataBindings.Add("Text", bS, "NamaBarang");
txtKd.DataBindings.Add("Text", bS, "KdBarang");
txtStok.DataBindings.Add("Text", bS, "Stok");
txtHarga.DataBindings.Add("Text", bS, "Harga");
```

Setelah itu kemudian kita isi masing2 tombol next , prev , first , last

Button Next :

```
//Pointer Maju Selangkah
bS.MoveNext();
```

Button Prev :

```
//Pointer Mundur Selangkah
bS.MovePrevious();
```

Button Last :

```
//Pointer Data Terakhir
bS.MoveLast();
```


Button First :

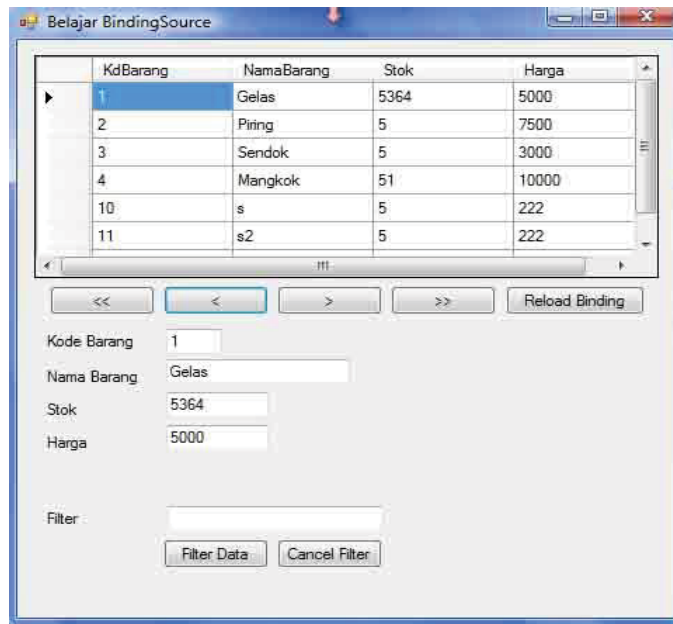
```
//Pointer Data Pertama  
bS.MoveFirst();
```

Kemudian Untuk Filter Data kita dapat gunakan seperti berikut :

```
//filter data berdasarkan kolom namabarang  
bS.Filter = "NamaBarang like '%" + txtFilter.Text + "%'";
```

Simple bukan binding source ? tapi dari titik ini masih banyak yang dapat di kembangkan lagi, ini merupakan tugas kita sebagai programmer untuk dapat meng-explore lebih jauh lagi.

Berikut hasil akhir aplikasi :



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

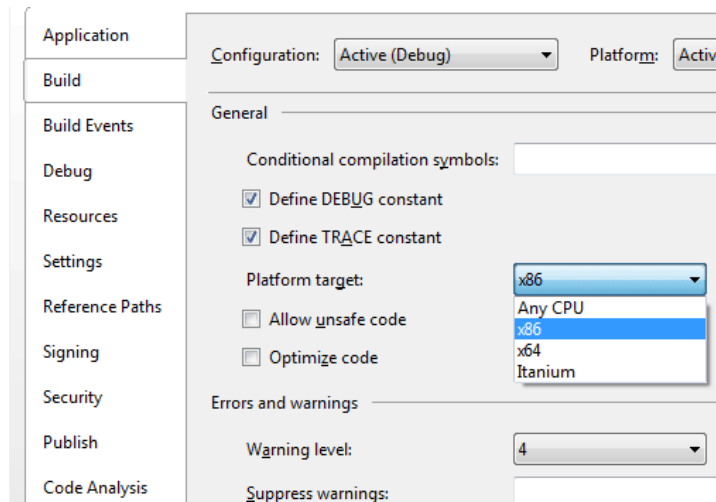
Pada pertemuan ke 10 ini saya akan membahas bagaimana membuat report ke dalam excel , banyak metode yang dapat kita gunakan untuk men-generate suatu report berdasarkan data yang ada , kali ini saya akan membahas cara yang simple dan mudah digunakan dan dimengerti oleh para pembaca , kita akan membuat 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 seberapa 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)
{
```

}

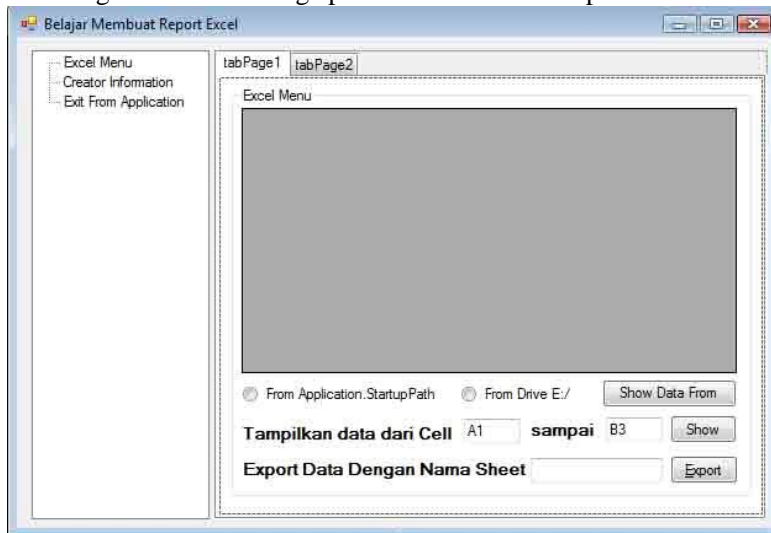
```
        MessageBox.Show(Convert.ToString(myArrayList));  
    }
```

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);
}
}
```

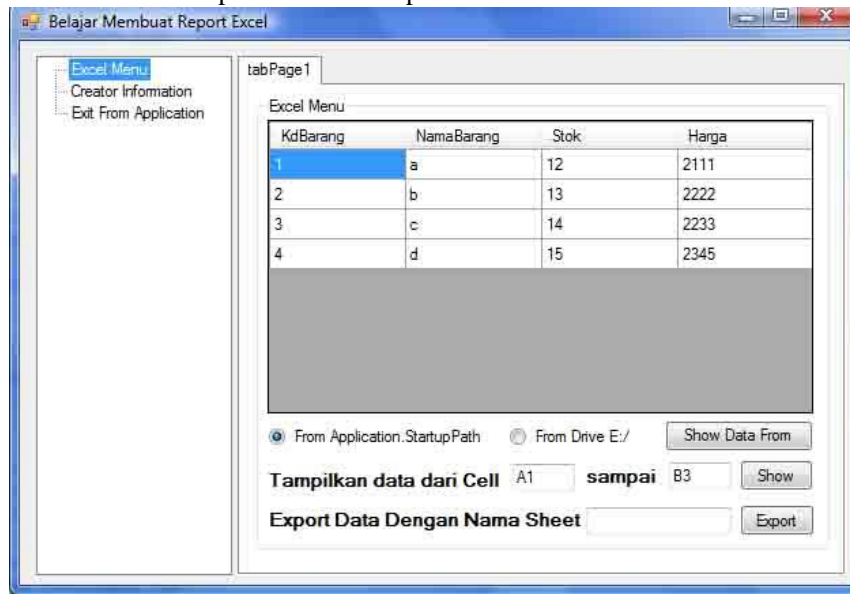
Setelah menu navigasi dan tabPage1 untuk menampilkan data jadi , saatnya kita mengatur LinkLabel pada tabPage2 , berikut tampilan tabPage2 :



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