

C# Part 2 – Class Dan Array

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

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#.

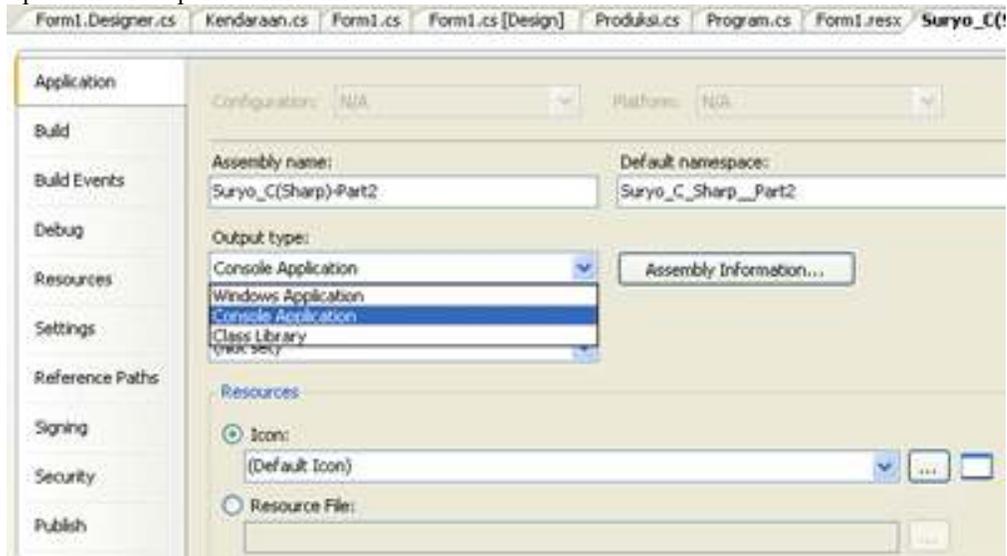
Sebagai contoh dalam Bumi ini . kita memiliki class dan object yang merupakan dari bumi ini yang kita gambarkan dalam sebuah makhluk hidup dan aktivitasnya. Semua yang ada di bumi ini disebut dengan object seperti manusi adalah objek , dan binatang juga objek. Mudah dimengerti bukan ? lalu apa itu class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

Dalam dunia ini kita memiliki perbedaan antara objek yang satu dengan objek yang lainnya , sebagai contoh adalah pengelompokan 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.

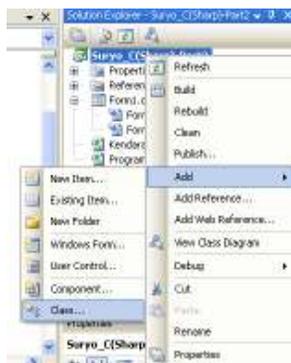
C# Class tergantung penggunaannya pada blok2 bahasa pemrograman. Yang saya maksud blok2 disini adalah setiap kita bekerja dengan C# dan membuat sebuah class pada program , maka class dari template yang telah kita buat tadi memiliki fungsi dan properti atau behavior yang sama dengan class sebelumnya tetapi 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 menyimpannya ke dalam lokasi dari variable 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 variable 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 variable 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 melalu *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.WriteLine("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.WriteLine("Pembeli ke " + (i + 1) + " :");
    string NamaPembeli = Console.ReadLine();
    Console.WriteLine("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.WriteLine("\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;

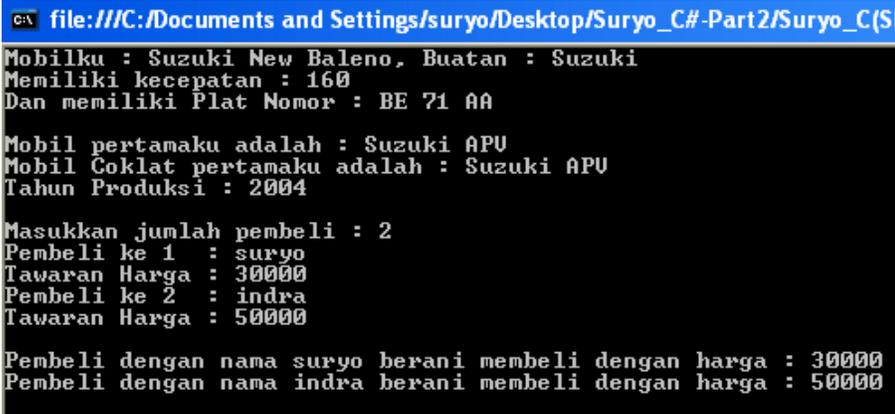
namespace Suryo_C_Sharp_Part2
{
    class Kendaraan
    {
        public string PlatNomor ;
        public string Model ;
        public string Manufaktur ;
        public int SpeedMax ;
        public Produksi keteranganMobil ;
    }
}
```

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
```

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