

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

Setelah kita mengerti Konsep yang telah saya rangkum pada pertemuan 1 , kita dapat memulai basic pengetahuan dari perbedaan masing-masing dataType , cara pemberian comment pada baris-baris yang tidak ingin kita jalankan dan juga menggunakan fungsi-fungsi yang ada , maupun penggunaan konversi dan compare , semua ini nantinya akan kita pakai ke depan untuk menghadapi kasus yang ada pada business logic yang akan kita hadapi , bila yang belum membaca artikel pertama saya tentang ASP.Net dapat membaca di www.suryodesign.asia/freedownload atau dapat di download di <http://ilmukomputer.org/author/suryodesign/> , semoga artikel saya ini dapat lebih membuka wawasan anda mengenai penggunaan dataType dan teman-temannya

Isi

Ada beberapa point yang akan kita pelajari disini seperti penggunaan comment , Variabel dan DataType , Assignment dan Initializers , String And Escaped Characters , Arrays , Array List , Enumeration , Variabel Operations , Type Conversion, Object-Based Manipulation , String Class , DateTime and TimeSpan Classes , Conditional Structure. Saya akan membahasnya satu per-satu agar lebih mudah di pahami atau easy to understand , dan setelah anda memahami ini kita dapat memulai pembuatan web ASP.Net pada pertemuan selanjutnya bila telah memiliki basic nya .

Comment

Comment kita butuhkan pada C# untuk membatasi baris2 / line yang tidak ingin kita kerjakan berikut adalah contoh comment pada C#

```
//Comment pertama
/*Comment ke dua
    Tes Baris ke 3 / *
```

Kita juga dapat memisah Code Statement pada C# sesuai kebutuhan kita untuk memotong baris code yang panjang agar lebih mudah dilihat dan di manage.

Code Statement Single Line
MyValue = MyValue1 + MyValue2 + MyValue3 ;

Code Statement yang di pisah menjadi 2 baris.
MyValue = MyValue1 + MyValue2 +
MyValue3 ;

Code Statement menjadi 3 baris
MyValue = MyValue1 +
MyValue2 +
MyValue3 ;

Variabel dan Data Type

```
//Membuat variable integer errorCode
Int errorCode ;
```

```
//membuat string Variabel myName
String myName;
```

Catatan : pada C# variable dengan nama name dan Name adalah berbeda, mungkin terkadang ini menjadi suatu hal yang membingungkan atau malah menjadi suatu hal yang menguntungkan bagi seorang programmer.

Setiap bahasa .Net memiliki type data variable yang sama , seperti pada VB.Net dan C# Memiliki 2 cara pendefinisian yang berbeda yaitu dim A as string (VB.Net) dan string A (C#) , tetapi CLR tidak membuat perbedaan , faktanya merka hanya 2 nama yang berbeda dengan base data type yang sama .

Berikut adalah Data Type pada VB.Net dan C# dengan Class Library name dan batasannya :

Class Library Name	VB Name	C# Name	Contains
Byte	Byte	byte	An integer from 0 to 255.
Int16	Short	short	An integer from -32,768 to 32,767.
Int32	Integer	int	An integer from -2,147,483,648 to 2,147,483,647.
Int64	Long	long	An integer from about -9.2e18 to 9.2e18.
Single	Single	float	A single-precision floating point number from approximately -3.4e38 to 3.4e38.
Double	Double	double	A double-precision floating point number from approximately -1.8e308 to 1.8e308.
Decimal	Decimal	decimal	A 128-bit fixed-point fractional number that supports up to 28 significant digits.
Char	Char	char	A single 16-bit Unicode character.
String	String	string	A variable-length series of Unicode characters.
Boolean	Boolean	bool	A true or false value.
DateTime	Date	*	Represents any date and time from 12:00:00 AM, January 1 of the year 1 in the Gregorian calendar, to 11:59:59 PM, December 31 of the year 9999. Time values can resolve values to 100 nanosecond increments. Internally, this data type is stored as a 64-bit integer.
TimeSpan	*	*	Represents a period of time, as in ten seconds or three days. The smallest possible interval is 1 <i>tick</i> (100 nanoseconds).
Object	Object	object	The ultimate base class of all .NET types. Can contain any data type or object.

Kita juga dapat mendefinisikan variabel dengan menggunakan nama dari .Net class Library seperti `System.String myName` ;

Assignment dan Initializers

ketika kita membuat variable , kita dapat assign value ke dalamnya , sepanjang value tersebut memiliki tipe data yang sesuai , berikut adalah Two Step Process pendefinisikan dengan menggunakan C# Language

//Definisikan variable

`Int errorCode;`

`String myName;`

//assign value

Komunitas eLearning IlmuKomputer.Com

Copyright © 2003-2009 IlmuKomputer.Com

```
errorCode = 10;  
MyName = "Suryo";
```

Kita juga dapat assign perintah di atas menjadi one step process menjadi

```
Int errorcode = 10;  
String myName = "Suryo";
```

Sekali lagi saya ingin memberitahu , dalam C# kita tidak dapat menggunakan variable yang belum memiliki value untuk menjadi factor pengali / pembagi dan sejenisnya seperti contoh berikut :

```
Int number ; //number tidak memiliki value hanya inisialisasi pembuatannya  
Number = number + 1;    //Error.
```

Harusnya ketika di inisialisasi number sendiri memiliki nilai awal yang kita assign sebelum kita menggunakan variabelnya itu sendiri.

```
Int number = 0 ; //number bernilai 0  
Number = number + 1 ;    /sukses , number bernilai 1
```

C# juga mesti memperhatikan tipe data dan isi dari value nya tersebut , seperti

```
Decimal myDecimal = 14.5;
```

14.5 otomatis menjadi tipe data double , dan syntax di atas tidak akan bekerja , untuk menghadapi masalah ini , C# mendefinisikan beberapa special characters yang dapat kita tambahkan pada akhir value , berikut

M(decimal)
D(double)
F(float)
L(long)

Sehingga kita dapat menulis variable di atas menjadi :

```
Decimal myDecimal = 14.5M ;
```

String And Escaped Characters

Berikut adalah contoh dari String dan Escaped Character :

```
\ " ( Double Quoted )  
\ n ( New line )  
\ t ( horizontal Tab )  
\ \ ( Backward slash )
```

Contoh penggunaan string dan escaped variable adalah pada penempatan directori ,

```
// path = "C:\\MyApp\\MyFiles";
```

Varibel di atas memiliki alamat C:\MyApp\MyFiles path , alternative nya adalah penggunaan @ di awal kalimat untuk menon-aktifkan C# escaping dan string seperti berikut :

```
path = @"c:\MyApp\MyFiles";
```

Arrays

Array mengijinkan kita untuk menyimpan nilai dengan memiliki data type yang sama , dan dapat di akses berdasarkan dimensi dan index number dimana dia disimpan , array dapat kita buat sesuai dengan dimensi kebutuhan yang kita perlukan .

Semua array dimulai dengan batas bawah / lower bound 0 , ketika kita membuat array dalam C# , kita mesti spesifikasi number pada elementnya , karena array di mulai dari 0 , maka untuk highest index untuk array yg memiliki 4 data maka highest index nya bernilai 3 atau jumlah data – 1 = highest index karena dimulai dari 0 untuk lower bound nya .

Contoh pembuatan array

```
//Array dengan 4 string , memiliki index dari 0 hingga 3
```

```
String[] stringArray = new string[4];
```

```
//Array dengan 2x4 grid array ( dengan total 8 data )
```

```
Int[,] intArray = new int[2,4];
```

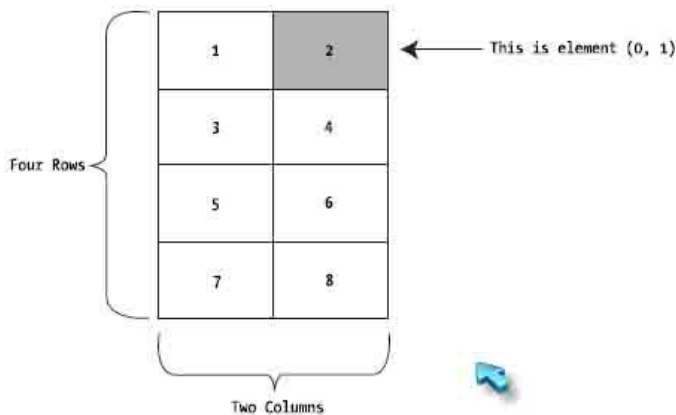
Kita dapat mengisi array berikut dengan cara seperti berikut :

```
String[] stringArray = { "a"."b"."c"."d" };
```

Same teknik juga bekerja untuk multidimensional arrays , contoh dengan penggunaan sepasang bracket { } :

```
Int[,] intArray = { {1,2},{3,4},{5,6},{7,8} }
```

Berikut adalah pemetaan array 2 dimensi dalam memory :



Untuk mengakses element dalam Array , kita dapat menspesifikasian index number dalam array bracket seperti berikut :

Int element ;

```
Element = intArray[0,1]; //element sekarang bernilai 2
```

Berikut adalah Array Members yang dapat kita gunakan :

Member	Description
Length	Returns an integer that represents the total number of elements in all dimensions of an array. For example, a 3×3 array has a length of 9.
GetLowerBound() and GetUpperBound()	Determines the dimensions of an array. As with just about everything in .NET, you start counting at zero (which represents the first dimension).
Clear()	Empties an array's contents.
IndexOf() and LastIndexOf()	Searches a one-dimensional array for a specified value and returns the index number. You cannot use this with multidimensional arrays.
Sort()	Sorts a one-dimensional array made up of comparable data such as strings or numbers.
Reverse()	Reverses a one-dimensional array so that its elements are backward, from last to first.

Array List

Array biasa tidak support redimensioning atau maksudnya disini adalah ketika kita membuat array , kita tidak dapat mengganti size yang telah kita buat berbeda dengan Array List , dan Array biasa membutuhkan proses penomoran index yang akan di copy untuk mengcopy dari 1 Array dan Array lainnya berbeda dengan Array List , kita dapat menggunakan collection classes yang di sediakan oleh .Net language , salah satunya adalah ArryList , berikut adalah snipet C# untuk menggunakan ArrayList :

```
ArrayList dynamicList = new ArrayList();  
dynamicList.Add("satu");  
dynamicList.Add("dua");  
dynamicList.Add("tiga");
```

berikut adalah contoh untuk mengambil string pertama , ketika kita mengambil kita mesti meng convert data yang ada di dalamnya , karena kita tidak mengetahui tipe data apa yang di kandung dalam ArrayList tersebut .

```
string item = Convert.ToString(dynamicList[0]);
```

Enumeration

Enumeration adalah sekumpulan group yang saling terhubung , yang mana memiliki nama yang mendeskripsikan grup ini .

Contoh enumeration untuk mendefinisikan beberapa tipe user yang berbeda :

```
Enum UserType
{
    Admin;
    Guest;
    Invalid;
}
```

Kita dapat menggunakan UserType enumeration sebagai special data type yang sebagai salah satu dari tiga kemungkinan value yang ada , kita dapat assign value menggunakan titik(dot) seperti contoh berikut :

```
UserType newUserType = UserType.Admin;
```

Pada contoh di atas , kita membuat UserType Baru dan kita set sebagai Admin , enumeration sebenarnya dapat kita akses dengan Number berdasarkan urutannya dimana 0 sebagai Admin , 1 sebagai Guest , dan 2 sebagai Invalid , dalam beberapa scenario , kita mungkin ingin mengontrol number yang berbeda untuk setiap kebutuhan atau bagian informasi :

```
Enum ErrorCode
{
    NoResponse = 166 ;
    TooBusy = 167 ;
    Pass = 0 ;
}
```

Dengan ErrorCode enumeration yang kita definisikan sebelumnya, kita dapat menggunakan nya seperti berikut :

```
ErrorCode err;
Err = DoSomething();
If(err = ErrorCode.Pass)
{
    //Sukses
}
```


Variabel Operations :

Kita dapat membuat semua standard data type pada C# ketika bekerja dengan number , seperti menggunakan math symbol , pembagian , pengalian , penambahan , modulus , atau pengurangan seperti berikut ;

Int number ;

Number = 4 + 2 * 3

//Number bernilai 10 karena derajat * lebih tinggi dari + maka * akan di kerjakan terlebih dahulu , untuk menentukan mana yang lebih dahulu di kerjakan , gunakan () pada bagian yang ingin di kerjakan terlebih dahulu

Number = (4+2) * 3

//Number bernilai 18

Berikut adalah contoh penggunaan masing-masing operator :

Operator	Description	Example
+	Addition	1 + 1 = 2.
-	Subtraction (and to indicate negative numbers)	5 - 2 = 3.
*	Multiplication	2 * 5 = 10.
/	Division	5 / 2 = 2.5.
%	Gets the remainder left after integer division	7 % 3 = 1.

Untuk menggabung string kita dapat menggunakan + seperti berikut

//penggabungan 3 string

myName = firstName + " " + lastName ;

dalam penambahan C# menyediakan beberapa cara seperti berikut :

myValue += 10

myValue *= 3

myValues /=12;

pada contoh di atas myValue +=10 sama dengan myValue = myValue + 10

untuk penggunaan fungsi2 matematika yang lebih jauh kita dapat mendefinisikan seperti berikut :

int myValue;

myValue = math.Sqrt(81); //bernilai 9

myValue = math.Round(42.8999,2); // bernilai 42/89


```
myValue = math.Abs(-10);           // bernilai 10
myValue = math.Log(24.212);        // bernilai 3.18
myValue = math.PI;                 // bernilai 3.14
```

Type Conversion

Untuk mengubah informasi dari satu data type ke data type yang lainnya , seperti contoh sample nya kita ingin menerima inputan dari textbox yang kita miliki kemudian kita ingin menjumlahnya dengan variable yang kita miliki , disini kita mesti convert ke integer sebelum dapat kita gunakan untuk menghindari kesalahan penjumlahan yang di akibatkan oleh dataType.

Contoh konversi 32 bit integer ke 64 bit integer:

```
Int mySmallValue;
Long myLargeValue;
//MySmall value diisi dengan nilai dari batas atas int32
mySmallValue = int32.MaxValue;
//MyLargeValue diisi dengna nilai mySmallValue
myLargeValue = mySmallValue;
```

berikut contoh pengubahan 32 bit integer ke 16 bit integer

```
int count32 = 1000;
short count 16;

//konversi dari 32 bit integer menjadi 16 bit integer , bila coun32 lebih besar dari count16
//maka akan menyebabkan problem, karena itu kita gunakan short disini
Count16 = (short)count32;
```

Contoh pengubahan dari string menjadi number count

```
Int count ;
String countString = "10";

//convert string "10" menjadi numeric value 10
Count = Convert.ToInt32(countString);

//convert numeric 10 menjadi string "10"
countString = Convert.ToString(count);
```

kita juga dapat menggunakan Parse untuk mengubah dari string menjadi numeric seperti berikut :

```
count = Int32.Parse(countString)
```

Object-Based Manipulation :

.Net merupakan Object Oriended dimana setiap data yang di bangun memiliki handle basic operation untuk menangani aksi yang ada seperti manipulasi strings , date dan number dengan C# atau VB.Net.

Seperti contoh setiap .Net class Library memiliki ToString() method . kita akan mencoba menggunakannya :

```
String myString;  
Int myInteger = 100;  
  
//Convert number menjadi string  
myString = myInteger.ToString();
```

The String Class

Salah satu variable data type yang banyak kita gunakan adalah string , string sendiri memiliki banyak fungsi dan method yang dapat kita gunakan dalam string Class , berikut adalah cara penggunaanya dalam C# :

```
String myString = "This is a test string";  
myString = myString.Trim(); // bernilai "his is a test string"  
myString = myString.Substring(0,4); // "this"  
myString = myString.ToUpper(); // "THIS"  
myString = myString.Replcace("IS","AT"); // "THAT"  
  
int length = myString.Length; // 4
```

kita juga dapat menggunakan beberapa perintah di atas kemudian kita gabung menjadi seperti berikut :

```
myString = myString.Trim.SubString(0,4).ToUpper().Replace("IS","AT");
```

atau berdasarkan variable yang kita miliki :

```
myString = "hello".ToUpper() // "HELLO"
```

Berikut adalah Table penggunaan Sytem.String class

Member	Description
Length	Returns the number of characters in the string (as an integer).
ToUpper() and ToLower()	Returns a copy of the string with all the characters changed to uppercase or lowercase characters.
Trim(), TrimEnd(), and TrimStart()	Removes spaces or some other characters from either (or both) ends of a string.
PadLeft() and PadRight()	Adds the specified character to either side of a string, the number of times you indicate. For example, PadLeft(3, " ") adds three spaces to the left side.
Insert()	Puts another string inside a string at a specified (zero-based) index position. For example, Insert(1, "pre") adds the string <i>pre</i> after the first character of the current string.
Remove()	Removes a specified number of strings from a specified position. For example, Remove(0, 1) removes the first character.
Replace()	Replaces a specified substring with another string. For example, Replace("a", "b") changes all <i>a</i> characters in a string into <i>b</i> characters.
Substring()	Extracts a portion of a string of the specified length at the specified location (as a new string). For example, Substring(0, 2) retrieves the first two characters.
StartsWith() and EndsWith()	Determines whether a string ends or starts with a specified substring. For example, StartsWith("pre") will return either true or false, depending on whether the string begins with the letters <i>pre</i> in lowercase.
IndexOf() and LastIndexOf()	Finds the zero-based position of a substring in a string. This returns only the first match and can start at the end or beginning. You can also use overloaded versions of these methods that accept a parameter that specifies the position to start the search.
Split()	Divides a string into an array of substrings delimited by a specific substring. For example, with Split(".") you could chop a paragraph into an array of sentence strings.
Join()	Fuses an array of strings into a new string. You can also specify a separator that will be inserted between each element.

The DateTime and TimeSpan Classes

DateTime dan TimeSpan class memiliki fungsi methods dan properties yang mengizinkan kita untuk melakukan beberapa kegiatan berikut :

- Extract bagian dari datetime(misalkan hanya bulan / tahun) atau konversi dari TimeSpan menjadi total hari atau total number dari menit yang ada .
- Memudahkan penghitungan tanggal
- Menentukan tanggal yang sekarang dan waktu sekarang dan informasi lainnya seperti hari / minggu atau tanggal kejadian sesuai dengan aktifitas yang kita miliki

Sebagai contoh :

```
//mendapatkan waktu sekarang , kemudian kita tambah 100 hari ke depan kemudian kita  
//dapatkan tahun nya saat itu
```

```
DateTime myDate = DateTime.Now;  
myDate = myDate.AddDays(100);  
string dateString = myDate.Year.ToString();
```

untuk TimeSpan sendiri penggunaannya seperti berikut :

```
DateTime myDate1 = DateTime.Now;  
DateTime myDate2 = DateTime.Now.AddHours(3000);
```

```
TimeSpan difference ;  
Difference = myDate2.Subtract(myDate1);  
Double numberOfMinutes;  
numberOfMinutes = difference.TotalMinutes;
```

Berikut adalah DateTime members yang dapat kita gunakan :

Member	Description
Now	Gets the current date and time.
Today	Gets the current date and leaves time set to 00:00:00.
Year, Date, Day, Hour, Minute, Second, and Millisecond	Returns one part of the DateTime object as an integer. For example, Month will return 12 for any day in December.
DayOfWeek	Returns an enumerated value that indicates the day of the week for this DateTime, using the DayOfWeek enumeration. For example, if the date falls on Sunday, this will return DayOfWeek.Sunday.
Add() and Subtract()	Adds or subtracts a TimeSpan from the DateTime.
AddYears(), AddMonths(), AddDays(), AddHours(), AddMinutes(), AddSeconds(), AddMilliseconds()	Adds an integer that represents a number of years, months, and so on, and returns a new DateTime. You can use a negative integer to perform a date subtraction.
DaysInMonth()	Returns the number of days in the month represented by the current DateTime.
IsLeapYear()	Returns true or false depending on whether the current DateTime is in a leap year.
ToString()	Changes the current DateTime to its string representation. You can also use an overloaded version of this method that allows you to specify a parameter with a format string.

Berikut adalah TimeSpan member yang dapat kita gunakan :

Member	Description
Days, Hours, Minutes, Seconds, Milliseconds	Returns one component of the current TimeSpan. For example, the Hours property can return an integer from 0 to 23.
TotalDays, TotalHours, TotalMinutes, TotalSeconds, TotalMilliseconds	Returns the total value of the current TimeSpan, indicated as a number of days, hours, minutes, and so on. For example, the TotalDays property might return a number like 234.342.
Add() and Subtract()	Combines TimeSpan objects together.
FromDays(), FromHours(), FromMinutes(), FromSeconds(), FromMilliseconds()	Allows you to quickly specify a new TimeSpan. For example, you can use TimeSpan.FromHours(24) to define a TimeSpan object exactly 24 hours long.

Conditional Structure

Kita dapat mendefinisikan Conditiona Structure untuk membandingkan kondisi yang ada saat ini , atau untuk membuat keputusan sesuan dengan logic kita .

Contoh Logical Operator :

Member	Description
Length	Returns an integer that represents the total number of elements in all dimensions of an array. For example, a 3×3 array has a length of 9.
GetLowerBound() and GetUpperBound()	Determines the dimensions of an array. As with just about everything in .NET, you start counting at zero (which represents the first dimension).
Clear()	Empties an array's contents.
IndexOf() and LastIndexOf()	Searches a one-dimensional array for a specified value and returns the index number. You cannot use this with multidimensional arrays.
Sort()	Sorts a one-dimensional array made up of comparable data such as strings or numbers.
Reverse()	Reverses a one-dimensional array so that its elements are backward, from last to first.

Kita dapat menggunakan banyak cara untuk membandingkan seperti pada operasi string dengan menggunakan (==) atau (!=) . kita juga dapat menggunakan String.Compare() method.

```
Int result;  
Result = String.Compare("apple","attach");           // result = -1  
Result = String.Compare("apple","all");               // result = 1  
Result = String.Compare("apple","apple");             // result = 0
```

Atau dapat juga kita gunakan seperti berikut :

```
String word = "apple";  
Result = word.CompareTo("attach"); // result = -1
```

Penutup

Banyak nya tipe data dan kebutuhan konversi terkadang membingungkan kita , ada banyak cara yang bisa kita gunakan tergantung dengan cara / alternative mana yang lebih kita kuasai , semoga contoh di atas dapat membuka logika piker kita tentang penggunaan dataType dan Konversi . diharap pembaca dapat lebih creative dan tidak hanya tergantung pada artikel ini untuk dapat menguasai lebih lanjut .

REFERENSI :

- MSDN 2008
- <http://msdn.microsoft.com>
- <http://wikipedia.com/>
- <http://www.microsoft.com>
- www.w3schools.com/
- <http://asp.net-tutorials.com/>
- Ebook – Beginning ASP.Net From Novice To Professional

Biografi Penulis



M.Suryo Pranoto – Alumni Mahasiswa Perguruan Tinggi Universitas Bina Nusantara , Aktif dalam beberapa komunitas komputer , dan beberapa project terutama berbasis aplikasi seperti VB.Net atau C#, dan sedang berusaha keras untuk menabung dan berencana untuk melanjutkan Cisco CCNP setelah menyelesaikan CCNA , memiliki hobby untuk sharing mengenai komputer mulai dari software hingga jual beli hardware maupun modding komputer.

Ym : suryolovetyka ,
Email : suryodesign@yahoo.co.id ,
Website : www.suryodesign.asia

Blog : www.suryodesign.wordpress.com

Ilmu Komputer : <http://ilmukomputer.org/author/suryodesign/>