

BAB 2

Variables, Data Type dan Operator

❖ Variables

Seperti pada bahasa pemrograman pada umumnya pasti memiliki yang namanya Variables. Pada umumnya tipe variable yang dimiliki oleh semua bahasa pemrograman relative hampir sama. Begitu juga pada C#. Tipe Variables pada C# dibagi menjadi beberapa kategori seperti berikut.

Type	Contoh
Integral	sbyte, byte, short, ushort, int, uint, long, ulong, and char
Floating point	float and double
Decimal	decimal
Boolean	true or false
Nullable	Nullable data types

Untuk deklarasi variabel adalah sebagai berikut :

```
<data_type> <variable_list>;
```

Dari cara deklarasi di atas, data_type adalah tipe data dari variabel misalnya, int, float double dll, sedangkan variable_list adalah variable yang anda deskripsikan. Contoh cara mendeskripsikan variable adalah sebagai berikut :

```
int i, j, k;  
char c, ch;
```

Untuk jenis tipe data pada C# mengenal 3 jenis tipe data yaitu.

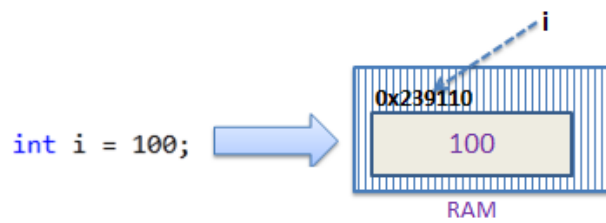
1. Value types
2. Reference types
3. Pointer types

❖ Value Type

Untuk tipe data Value Type adalah tipe data yang pada umumnya dipakai untuk menampung nilai dari variabel untuk melakukan operasi pada statemen program. Detail dari Tipe Data Value adalah sebagai berikut.

Untuk tipe data ini ketika user mendefinikan variable maka variable tersebut akan dialokasikan kedalam memory beserta isi dari variable tersebut, misalkan :

```
int i=100;
```



Perhatikan pada gambar di atas, bahwa variabel `i` di alokasikan pada alamat memori (`0x239110`) beserta isi dari variable `i` yaitu 100.

Type	Panjang	Range	Nilai Default
bool	8-bit	True or False	False
byte	8-bit	0 to 255	0
char	16-bit	U +0000 to U +ffff	'\0'
decimal	128-bit	$(-7.9 \times 10^{28} \text{ to } 7.9 \times 10^{28}) / 10^0 \text{ to } 28$	0.0M
double	64-bit	$(+/-)5.0 \times 10^{-324} \text{ to } (+/-)1.7 \times 10^{308}$	0.0D
float	32-bit	$-3.4 \times 10^{38} \text{ to } + 3.4 \times 10^{38}$	0.0F
int	32-bit	-2,147,483,648 to 2,147,483,647	0
long	64-bit	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807	0L
sbyte	8-bit	-128 to 127	0
short	16-bit	-32,768 to 32,767	0
uint	32-bit	0 to 4,294,967,295	0

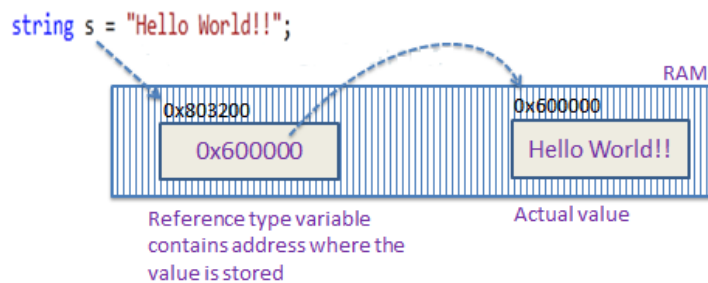
ulong	64-bit	0 to 18,446,744,073,709,551,615	0
ushort	16-bit	0 to 65,535	0

❖ Reference Types

Tidak seperti pada tipe data *Value Types*, untuk tipe data *Reference Types* adalah tidak menyimpan nilai variable secara langsung. Tetapi menyimpan alamat memori dimana nilai dari variable disimpan. Misalkan.

```
string s = "Hello World!!";
```

Perhatikan Gambar berikut sebagai ilustrasi.



Variabel **s** tidak menyimpan nilai dari variabel **s** yang berisi "Hello World!!" tetapi menyimpan alamat dimana nilai dari variabel **s** yang berisi "Hello World!!" disimpan. Untuk tipe Pointer akan dibahas pada bab tertentu

❖ Operators

Operator adalah simbol yang dikenali oleh kompiler yang digunakan untuk melakukan operasi aritmatik dan logika pada program. Pada C# tipe dari operator adalah sebagai berikut.

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Bitwise Operators
5. Assignment Operators
6. Misc Operators

Arithmetic Operators

Operator	Description	Example
+	Penambahan dua operator	$A + B = 30$
-	Pengurangan dua operator	$A - B = -10$
*	Perkalian dua operator	$A * B = 200$
/	Pembagian dua operator	$B / A = 2$
%	Modulus/ mencari sisa nilai pembagian	$B \% A = 0$
++	Increment operator, menaikkan nilai 1 di setiap operator di panggil/dijalankan	$A++ = 11$
--	Decrement operator menurunkan nilai 1 di setiap operator di panggil/dijalankan	$A-- = 9$

Relational Operators

Operator	Deskripsi	Contoh
==	Operator pembandingan sama dengan	$(A == B)$ is not true.
!=	Operator pembandingan tidak sama dengan	$(A != B)$ is true.
>	Operator pembandingan lebih besar dari	$(A > B)$ is not true.
<	Operator pembandingan lebih kecil dari	$(A < B)$ is true.
>=	Operator pembandingan lebih besar sama dengan	$(A >= B)$ is not true.

<=	Operator pembandingan lebih kecil sama dengan	(A <= B) is true.
----	---	-------------------

Logical Operators

Operator	Deskripsi	Contoh
&&	Disebut operator logic AND. Operator ini akan benar jika kedua kondisi bernilai benar	(A && B) is false.
	Disebut operator logic OR. Operator ini akan benar jika salah satu kondisi bernilai benar	(A B) is true.
!	Disebut operator logic NOT. Operator ini digunakan untuk membalik nilai dari sebuah kondisi, jika kondisi true menjadi false dan sebaliknya	!(A && B) is tr

Bitwise Operators

Operator	Ddeskripsi	Contoh
&	Operasi AND	(A & B) = 12, which is 0000 1100
	Operasi OR	(A B) = 61, which is 0011 1101
^	Operasi XOR	(A ^ B) = 49, which is 0011 0001
<<	Operasi Left Shift	A << 2 = 240, which is 1111 0000
>>	Operasi Right Shift	A >> 2 = 15, which is

		0000 1111
--	--	-----------

Assignment Operators

Operator	Deskripsi	Contoh
=	Operator untuk memberikan nilai dari operator yang di sebelah kanan ke sebelahkiri	$C = A + B$ assigns value of $A + B$ into C
+=	Gabungan operator = dan +	$C += A$ is equivalent to $C = C + A$
-=	Gabungan operator = dan -	$C -= A$ is equivalent to $C = C - A$
*=	Gabungan operator = dan *	$C *= A$ is equivalent to $C = C * A$
/=	Gabungan operator = dan /	$C /= A$ is equivalent to $C = C / A$
%=	Gabungan operator = dan %	$C \% = A$ is equivalent to $C = C \% A$
<<=	Gabungan operator = dan <<	$C <<= 2$ is same as $C = C << 2$
>>=	Gabungan operator = dan >>	$C >>= 2$ is same as $C = C >> 2$
&=	Gabungan operator = dan &	$C \&= 2$ is same as $C = C \& 2$
^=	Gabungan operator = dan ^	$C \wedge= 2$ is same as $C =$

		$C \wedge 2$
<code> =</code>	Gabungan operator = dan	<code>C = 2</code> is same as <code>C = C 2</code>

Misc Operators

Operator	Deskripsi	Contoh
<code>sizeof()</code>	Memberikan nilai balik ukuran tipe data	<code>sizeof(int)</code> , returns 4.
<code>typeof()</code>	Memberikan nilai balik tipe dari kelas	<code>typeof(StreamReader);</code>
<code>&</code>	Meberikan nilai balik berupa alamat pada memori suatu variabel.	<code>&a</code> ; returns actual address of the variable.
<code>*</code>	Variabel pointer	<code>*a</code> ; creates pointer named 'a' to a variable.
<code>?:</code>	Kondisional ekspresi, digunakan untuk mempersingkat kondisional ekspresi pada pada kondisi tertentu	If Condition is true ? Then value X : Otherwise value Y

PERCOBAAN

1. Tipe Data dan Variabel

Berikut adalah contoh penggunaan tipe data dalam program

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Hello
{
    class Program
    {
        static void Main(string[] args)
        {
            int a = 4;
            long b = 46L;
            float c = 5.0F;
            double d = 46.4D;
            char e = 'A';
            Console.WriteLine(a);
            Console.WriteLine(b);
            Console.WriteLine(c);
            Console.WriteLine(d);
            Console.WriteLine(e);
            Console.ReadKey();
        }
    }
}
```

2. Passing by Value dan Passing by Reference

```
using System;

class Program
{
    static void ubahNilai(int x)
    {
        x = 46;
        Console.WriteLine(x);
    }

    public static void Main()
    {
        int i = 45;
        Console.WriteLine(i);
        ubahNilai(i);
        Console.WriteLine(i);
        Console.ReadKey();
    }
}
```



```
}
```

```
using System;
public class Student
{
    public string StudentName { get; set; }
}

public class Program
{
    public static void ubahReference(Student std2)
    {
        std2.StudentName = "Valentino";
    }

    public static void Main()
    {
        Student std1 = new Student();
        std1.StudentName = "Rossi";
        ubahReference(std1);
        Console.WriteLine(std1.StudentName);
        Console.ReadKey();
    }
}
```

3. Operators

```
using System;

public class Program
{
    public static void Main()
    {
        string message1 = "Hello";
        string message2 = message1 + " World!!";
        Console.WriteLine(message2);

        int i = 5, j = 20;
        int sum = i + j;
        Console.WriteLine("{0} + {1} = {2}", i, j, sum);
        Console.ReadKey();
    }
}
```

```
using System;

public class Program
{
    class Program
    {
        static void Main(string[] args)
        {
```

```

        int a = 45;
        int b = 46;
        int c;

        c = a + b;
        Console.WriteLine(" 1 - Nilai dari c adalah {0}", c);
        c = a - b;
        Console.WriteLine(" 2 - Nilai dari c adalah {0}", c);
        c = a * b;
        Console.WriteLine(" 3 - Nilai dari c adalah {0}", c);
        c = a / b;
        Console.WriteLine(" 4 - Nilai dari c adalah {0}", c);
        c = a % b;
        Console.WriteLine(" 5 - Nilai dari c adalah {0}", c);
        c = a++;
        Console.WriteLine(" 6 - Nilai dari c adalah {0}", c);
        c = a--;
        Console.WriteLine(" 7 - Nilai dari c adalah {0}", c);
        Console.ReadLine();
    }
}

```

```

using System;

class Program
{
    static void Main(string[] args)
    {
        int a = 45;
        int b = 46;

        if (a == b)
        {
            Console.WriteLine("Nilai a sama dengan b");
        }
        else
        {
            Console.WriteLine("Nilai a tidak sama dengan b");
        }

        if (a < b)
        {
            Console.WriteLine("Nilai a lebih besar b");
        }
        else
        {
            Console.WriteLine("Nilai a lebih kecil dari b");
        }
    }
}

```

```

using System;
class Program
{
    static void Main(string[] args)
    {
        bool a = true;
        bool b = true;

        if (a && b)
        {
            Console.WriteLine("Benar");
        }

        if (a || b)
        {
            Console.WriteLine("Benar");
        }
        /* nilai a dan b diganti */
        a = false;
        b = true;

        if (a && b)
        {
            Console.WriteLine("Benar");
        }
        else
        {
            Console.WriteLine("Salah");
        }

        if (!(a && b))
        {
            Console.WriteLine("Benar");
        }
        Console.ReadLine();
    }
}

```

```

using System;
class Program
{
    static void Main(string[] args)
    {
        int a = 21;
        int c;
        c = a;
        Console.WriteLine(" 1 - = Nilai dari c = {0}", c);

        c += a;
        Console.WriteLine(" 2 - += Nilai dari c = {0}", c);

        c -= a;
        Console.WriteLine(" 3 - -= Nilai dari c = {0}", c);

        c *= a;
        Console.WriteLine(" 4 - *= Nilai dari c = {0}", c);
    }
}

```

```

    c /= a;
    Console.WriteLine(" 5 - /= Nilai dari c = {0}", c);

    c = 200;
    c %= a;
    Console.WriteLine(" 6 - %= Nilai dari c = {0}", c);

    c <= 2;
    Console.WriteLine(" 7 - <= Nilai dari c = {0}", c);

    c >= 2;
    Console.WriteLine(" 8 - >= Nilai dari c = {0}", c);

    c &= 2;
    Console.WriteLine(" 9 - &= Nilai dari c = {0}", c);

    c ^= 2;
    Console.WriteLine(" 10 - ^= Nilai dari c = {0}", c);

    c |= 2;
    Console.WriteLine("Line 11 - |= Nilai dari c = {0}", c);
    Console.ReadLine();
}
}

```

PRAKTIKUM WORKSHOP

1. Cobalah program di atas, Capture hasilnya dan berikan analisa
2. Buatlah pada laporan sebagai tugas Workshop.