

## **Materi 2: Math,String dan Percabangan (If...else,Switch Case)**

---

# Silabus

Pertemuan	Materi
1	Pengenalan .NET Framework (namespace,class,method,variabel,tipe data dan operator)
2	Math ,String dan Percabangan ( If...else , Switch Case )
3	Perulangan ( While , Do While , For , For Each )
4	Array
5	Method
6	Form
7	Event
8	UTS
9	OOP (class & object,constructor,access modifier)
10	OOP (property,inheritance,polymorphism, enum )
11	Exception (try catch,finally,trow keyword)
12	Koneksi Database (Konsep MVC,koneksi database,create,read)
13	Koneksi Database (update dan delete)
14	Membuat tampilan responsive dan validasi
15	Persiapan UAS dan sharing-sharing
16	Projek

# Daftar isi

---

- **Materi 2: Math ,String dan Percabangan ( If...else ,Switch Case )**
- Silabus
- Daftar isi
- C# Math
  - Math.Maks( x,y )
  - Math.Min( x,y )
  - Math.Sqrt( x )
  - Math.Abs( x )
  - Math.Round()
- C# Strings
  - C # String
  - String Length
  - Other Methods
  - Penggabungan String
  - String Interpolation
  - Akses String
- C# If ... Else
  - C# Conditions and If Statements
  - If
  - Else
  - Else If
  - Ternary Operator
- C# Switch
  - C# Switch Statements

# C# Math

---

Class C# Math memiliki banyak metode yang memungkinkan Anda melakukan tugas matematika pada angka.

## Math.Maks( x,y )

Metode ini dapat digunakan untuk mencari nilai tertinggi dari x dan y : `Math.Max(x,y)`

contoh :

```
Math.Max(8,4);
```

## Math.Min( x,y )

Metode ini dapat digunakan untuk mencari nilai terendah dari x dan y : `Math.Min(x,y)`

contoh :

```
Math.Max(8,4);
```

## Math.Sqrt( x )

Metode mengembalikan akar kuadrat dari x : `Math.Sqrt(x)`

contoh :

```
Math.Sqrt(64);
```

## Math.Abs( x )

Metode mengembalikan nilai absolut (positif) dari x : `Math.Abs(x)`

contoh :

```
Math.Abs(-4.7);
```

## Math.Round()

`Math.Round()` membulatkan angka ke bilangan bulat terdekat.

contoh :

```
Math.Round(9.99);
```

# C# Strings

---

## C # String

String digunakan untuk menyimpan teks. Variabel string berisi kumpulan karakter yang dikelilingi oleh tanda kutip ganda:

Contoh : Dapatkan Server C# Anda sendiri Buat variabel tipe string dan berikan nilai:

```
string greeting = "Hello";
```

Variabel string dapat berisi banyak kata, jika Anda ingin: contoh :

```
string greeting2 = "Nice to meet you!";
```

## String Length

Sebuah string dalam C# sebenarnya adalah sebuah objek, yang berisi properti dan metode yang dapat melakukan operasi tertentu pada string. Misalnya, panjang string dapat ditemukan dengan properti `Length` :

```
string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
Console.WriteLine("The length of the txt string is: " + txt.Length);
```

## Other Methods

Ada banyak metode string yang tersedia, misalnya `ToUpper()` dan `ToLower()` , yang mengembalikan salinan string yang diubah menjadi huruf besar atau kecil:

contoh :

```
string txt = "Hello World";  
Console.WriteLine(txt.ToUpper()); // Outputs "HELLO WORLD"  
Console.WriteLine(txt.ToLower()); // Outputs "hello world"
```

## Penggabungan String

Operator `+` dapat digunakan di antara string untuk menggabungkannya. Ini disebut gabungan :

contoh :

```
string firstName = "John ";
string lastName = "Doe";
string name = firstName + lastName;
Console.WriteLine(name);
```

Anda juga dapat menggunakan `string.Concat()` metode ini untuk menggabungkan dua string:

contoh :

```
string firstName = "John ";
string lastName = "Doe";
string name = string.Concat(firstName, lastName);
Console.WriteLine(name);
```

## String Interpolation

Opsi lain dari penggabungan string , adalah **interpolasi string** , yang menggantikan nilai variabel menjadi placeholder dalam sebuah string. Perhatikan bahwa Anda tidak perlu khawatir tentang spasi, seperti dengan penggabungan:

```
string firstName = "John";
string lastName = "Doe";
string name = $"My full name is: {firstName} {lastName}";
Console.WriteLine(name);
```

## Akses String

Anda dapat mengakses karakter dalam sebuah string dengan mengacu pada nomor indeksinya di dalam tanda kurung siku `[]`.

Contoh ini mencetak **karakter pertama** di `myString` :

```
string myString = "Hello";
Console.WriteLine(myString[0]); // Outputs "H"
```

Anda juga dapat menemukan posisi indeks dari karakter tertentu dalam sebuah string, dengan menggunakan `IndexOf()` metode:

```
string myString = "Hello";  
Console.WriteLine(myString.IndexOf("e")); // Outputs "1"
```

Metode lain yang berguna adalah `Substring()`, yang mengekstrak karakter dari string, mulai dari posisi/indeks karakter yang ditentukan, dan mengembalikan string baru. Metode ini sering digunakan bersama `IndexOf()` untuk mendapatkan posisi karakter tertentu:

```
// Full name  
string name = "John Doe";  
  
// Location of the letter D  
int charPos = name.IndexOf("D");  
  
// Get last name  
string lastName = name.Substring(charPos);  
  
// Print the result  
Console.WriteLine(lastName);
```



# C# If ... Else

---

## C# Conditions and If Statements

Gunakan `if` pernyataan untuk menentukan blok kode C# yang akan dieksekusi jika kondisinya `True`.  
contoh :

```
if (20 > 18)
{
    Console.WriteLine("20 is greater than 18");
}
```

Gunakan `else` pernyataan untuk menentukan blok kode yang akan dieksekusi jika kondisinya `False`.

sintaks

### If

```
if (condition)
{
    // block of code to be executed if the condition is True
}
else
{
    // block of code to be executed if the condition is False
}
```

### Else

```
int time = 20;
if (time < 18)
{
    Console.WriteLine("Good day.");
}
else
{
    Console.WriteLine("Bad day.");
}
```

```
    Console.WriteLine("Good evening.");  
}  
// Outputs "Good evening."
```

## Else If

Gunakan `else if` pernyataan untuk menentukan kondisi baru jika kondisi pertama adalah `False`

sintaks

```
if (condition1)  
{  
    // block of code to be executed if condition1 is True  
}  
else if (condition2)  
{  
    // block of code to be executed if the condition1 is false and condition2  
}  
else  
{  
    // block of code to be executed if the condition1 is false and condition2
```

```
int time = 22;  
if (time < 10)  
{  
    Console.WriteLine("Good morning.");  
}  
else if (time < 20)  
{  
    Console.WriteLine("Good day.");  
}  
else  
{  
    Console.WriteLine("Good evening.");  
}  
// Outputs "Good evening."
```

## Ternary Operator

sintaks:

```
variable = (condition) ? expressionTrue : expressionFalse;
```

```
int time = 20;  
string result = (time < 18) ? "Good day." : "Good evening.";   
Console.WriteLine(result);
```

# C# Switch

---

## C# Switch Statements

Gunakan `switch` pernyataan untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

```
switch(expression)
{
    case x:
        // code block
        break;
    case y:
        // code block
        break;
    default:
        // code block
        break;
}
```

```
int day = 4;
switch (day)
{
    case 1:
        Console.WriteLine("Monday");
        break;
    case 2:
        Console.WriteLine("Tuesday");
        break;
    case 3:
        Console.WriteLine("Wednesday");
        break;
    case 4:
        Console.WriteLine("Thursday");
        break;
    case 5:
        Console.WriteLine("Friday");
        break;
    case 6:
        Console.WriteLine("Saturday");
}
```

```
        break;
    case 7:
        Console.WriteLine("Sunday");
        break;
}
// Outputs "Thursday" (day 4)
```