



Back End Development 1

sesi 3



C#

+

**Programming
Logic**

C# Programming Logic - Sesi 03

INTRODUCTIONS

Logika Pemrograman dalam C# ini juga dikenal dengan Struktur Pengambilan Keputusan yang mana menyediakan perhitungan logika yang akan eksekusi beberapa bagian kode program.

Contoh.

Masih ingat dengan live code terakhir di sesi 2 ?

Dimana kita buat beberapa kondisi :

Jika Pengguna umur > 18 maka akan menampilkan text Welcome to the OCBC

Dan sebaliknya ...

```
16      // menggunakan logika AND
17      if(isAdult && isPasswordValid)
18      {
19          Console.WriteLine("WELCOME TO THE CLUB!");
20      }
21      else
22      {
23          Console.WriteLine("Sorry, try again!");
24      }
```

Nah dengan kondisi seperti ini kita bisa menentukan apa yang akan dan harus dikerjakan oleh aplikasi yang kita buat nantinya.

C# Programming Logic - Sesi 03

Statements - IF

```
16 // menggunakan logika AND
17 if(isAdult && isPasswordValid)
18 {
19     Console.WriteLine("WELCOME TO THE CLUB!");
20 }
21 else
22 {
23     Console.WriteLine("Sorry, try again!");
24 }
```

Statement If selalu erat kaitannya dengan logika BOOLEAN :

Apabila If menghasilkan nilai true, maka baris kode program yang berkaitan akan di eksekusi.
Begitupun sebaliknya : Apabila If menghasilkan nilai false, maka baris kode tersebut akan diabaikan.

```
if (condition) {
    //statements
}
```

Dalam Menggunakan Statement IF kita juga dapat memanfaatkan Operator Relational untuk membangun logika dengan Statement IF.

Misal kita akan membuat logika untuk mengetahui angka mana yang lebih besar. Pada kode Program di atas, IF menjelaskan bahwa Apabila 83 lebih dari 30 maka statement Console.WriteLine akan di jalankan yang mana menampilkan kata “y Lebih Besar daripada x” pada layar.

misal pada program di atas kita ubah operator lebih dari “>” menjadi kurang dari “<” maka kata “y Lebih Besar daripada x” tidak akan dicetak.

Contoh 1 :

```
if (10 == 10) {  
Console.WriteLine("Benar"); }  
// Outputs "Benar"
```

Kode Program di atas, IF menjelaskan bahwa Apabila 10 sama dengan 10 maka statement Console.WriteLine akan di jalankan, yang mana menampilkan kata Benar pada layar.

Contoh 2 :

```
if (10 != 10) {  
    Console.WriteLine("Benar");  
}
```

pada program di atas operator diubah dari “==” menjadi bukan sama dengan “!=” maka kata benar tidak akan dicetak.

```
1  using System;  
2    
3  public class Logika1  
4  {  
5      0 references  
6      public static void Main()  
7      {  
8          int a = 92;  
9          int b = 53;  
10         if (a > b) {  
11             Console.WriteLine("a lebih besar dari b"); }  
12             // Outputs "a lebih besar dari pada b"  
13         }  
14     }
```

```
Maliks-MacBook-Air:SourceCode swijaya$  
ut:Logika1 && mono Logika1  
a lebih besar dari b
```



C# Programming Logic - Sesi 03

Statement If-Else

Statement Else adalah pengikut dari Statement IF, apabila Statement IF adalah statement Utama maka statement Else adalah statement kedua atau statement Opsional.

Statement Else akan dieksekusi apabila Statement IF tidak mendapatkan hasil atau mempunyai hasil Salah.

```
if (condition) {  
    //statements  
}  
else {  
    //statements  
}
```

Compiler akan menguji kondisi:

- Apabila statement bernilai true, maka kode di dalam pernyataan IF akan dieksekusi.
- Apabila statement bernilai false, maka kode dalam pernyataan Else akan dieksekusi.

```

1  using System;
2
3  0 references
4  public class Logika2
5  {
6      0 references
7      public static void Main()
8      {
9          int nilai = 75;
10
11         if (nilai < 60) {
12             Console.WriteLine("Nilai Kamu C" );
13         }
14         else if (nilai < 80) {
15             Console.WriteLine("Nilai kamu B");
16         }
17         else {
18             Console.WriteLine("Nilai kamu A");
19         }
20     }
21 }

```

```

Maliks-MacBook-Air:Sesi3 swijaya$ cd "/Users/swijaya/
gika2 && mono Logika2
Nilai kamu B

```



Task : Buat Fungsi Login sederhana dengan “Username” & “Password” dimana :
Jika kondisi benar maka sistem akan menampilkan **Anda Berhasil Login**,

Apabila salah satu data input salah baik itu username atau password maka sistem akan menampilkan **Username atau Password anda salah**.

```
Maliks-MacBook-Air:Sesi3 swijaya$ cd '
gika3 && mono Logika3
Username: ocbc
Password: bootcamp
Anda berhasil login
```

Benar

```
Maliks-MacBook-Air:Sesi3 swijaya$ cd '
gika3 && mono Logika3
Username: ocbc
Password: BootCamp
Username atau Password anda salah
```

Salah

Jawaban :

```
1  using System;
2
   0 references
3  public class Logika3
4  {
   0 references
5      public static void Main()
6      {
7          string Username;
8          String Password;
9
10         //digunakan untuk menginput, isikan username= ocbc dan password= bootcamp
11         Console.Write("Username: ");
12         Username = Console.ReadLine();
13         Console.Write("Password: ");
14         Password = Console.ReadLine();
15
16         //logika percabangan jika username dan pass sama maka kondisi pertama akan terpe
17         if (Username == "ocbc" && Password == "bootcamp")
18             Console.WriteLine("Anda berhasil login");
19
20         //jika tidak maka kondisi kedua akan terpenuhi
21         else
22             Console.WriteLine("Username atau Password anda salah");
23         // Console.ReadKey();
24     }
25 }
```



C# Programming Logic - Sesi 03

STATEMENT if...else...if

Jika percabangan sebelumnya sering digunakan pada sebuah program untuk fungsi login, maka percabangan ini juga tidak kalah penting kegunaanya.

Percabangan IF..ELSE..IF ini dapat digunakan untuk menetapkan output sesuai kondisi.

Contohnya pada sebuah program pendidikan yang mengkategorikan nilai angka menjadi grade. Misalkan nilai murid tersebut 80 maka akan mendapatkan grade B dan begitu seterusnya.

Fungsi tersebut biasanya menggunakan jenis percabangan ini supaya dapat berjalan dengan optimal.

C# Programming Logic - Sesi 03

STATEMENT if...else...if

```
1  using System;
2
3  public class Logika4
4  {
5      public static void Main()
6      {
7          double Nilai;
8          Console.Write("Nilai: ");
9          Nilai = Convert.ToDouble(Console.ReadLine());
10
11         if (Nilai >= 85)
12         {
13             Console.WriteLine("Kamu mendapat grade A");
14         }
15         else if (Nilai >= 65)
16         {
17             Console.WriteLine("Kamu mendapat grade B");
18         }
19         else if (Nilai >= 45)
20         {
21             Console.WriteLine("Kamu mendapat grade C");
22         }
23         else if (Nilai <= 25)
24         {
25             Console.WriteLine("Kamu mendapat grade D");
26         }
27     }
28 }
```

```
Maliks-MacBook-Air:Sesi3 swijaya$
gika4 && mono Logika4
Nilai: 70
Kamu mendapat grade B
```



C# Programming Logic - Sesi 03

NICE TO KNOW

Apabila terlalu banyak menggunakan pernyataan else if, kode program Anda bisa menjadi terlihat berantakan dan susah dibaca.

Pada skenario seperti ini, solusi yang lebih baik adalah menggunakan pernyataan switch.

Pernyataan switch pada dasarnya hanyalah menggantikan beberapa pernyataan else if

C# Programming Logic - Sesi 03

Statement Switch..Case..

Perintah switch adalah salah satu struktur kode yang pernah kita gunakan. Hampir selalu, struktur if dapat menggantikan struktur switch. Bentuk perintah switch pada C# adalah:

```
switch (varA) {  
    case [literal 1]:  
        //Blok kode 1  
        break;  
    case [literal 2]:  
        //Blok kode 2  
        break;  
    case [literal 3]:  
        //Blok kode 3  
        break;  
    default:  
        //Jika yang lain tidak  
        terpenuhi  
        break;  
}
```



```

1 using System;
2
3 public class Logika5
4 {
5     public static void Main()
6     {
7         int nilai;
8         nilai = 7;
9         switch (nilai)
10        {
11            case 1:
12                Console.WriteLine("Januari");
13                break;
14            case 2:
15                Console.WriteLine("Februari");
16                break;
17            case 3:
18                Console.WriteLine("Maret");
19                break;
20            case 4:
21                Console.WriteLine("April");
22                break;
23            case 5:
24                Console.WriteLine("Mei");
25                break;
26            case 6:
27                Console.WriteLine("Juni");
28                break;
29            case 7:
30                Console.WriteLine("Juli");
31                break;
32            case 8:
33                Console.WriteLine("Agustus");
34                break;
35            case 9:
36                Console.WriteLine("September");
37                break;
38            case 10:
39                Console.WriteLine("Oktober");
40                break;
41            case 11:
42                Console.WriteLine("November");
43                break;
44            case 12:
45                Console.WriteLine("Desember");
46                break;
47            default:
48                Console.WriteLine("Bulan ?");
49                break;
50        }
51        Console.ReadLine();
52    }
53 }

```

Maliks-MacBook-Air:Sesi
Juli



HACKTIV8

C# Programming Logic - Sesi 03

LOOPING

Konsep Dasar : Melakukan suatu hal dengan cara berulang.

Dalam Programming, kita menggunakan looping untuk meng-iterasi item didalam sebuah Collection atau elemen didalam sebuah array.

Dalam hal ini C# menyediakan konstruksi dasar yang dikenal sebagai “loop” untuk kita implementasikan logika iterasi.

Contoh metode looping seperti while, for & do..while

Mari kita bahas 1 per 1

C# Programming Logic - Sesi 03

While


`while` akan menjalankan atau mengeksekusi kode selama kondisi yang diberikan masih `TRUE`.

Untuk mendeklarasikan struktur `while` dapat dilihat pada potongan kode berikut:

```
while (kondisi)
{
    kodeA;
    kodeB;
    kodeC;
}
```

`while` akan terus mengeksekusi `kodeA`, `kodeB` dan `kodeC` secara terus-menerus hingga kondisi menjadi `FALSE`. Saat kondisi berubah menjadi `FALSE`, `while` akan berhenti mengeksekusi `kodeA`, `kodeB` dan `KodeC`.

Contoh penggunaanya dapat dilihat pada potongan kode berikut:

```
1  using System;
2  
   0 references
3  public class Logika6
4  {
   0 references
5      public static void Main()
6      {
7          // membuat variable
8          int a = 10;
9
10         // loop while
11         while (a < 20) // menghasilkan TRUE jika variable a lebih kecil dari 20
12         {
13             Console.WriteLine("Nilai a = " + a );
14             a = a + 1;
15         }
16     }
17 }
```

Pada potongan kode tersebut dapat dilihat jika while akan terus melakukan:

- Print nilai variable a
- menambahkan nilai a sebanyak 1

Sehingga, program akan melakukan print dari 10 hingga 19 sebelum berhenti.


C# Programming Logic - Sesi 03

FOR

Berbeda dengan `while`, loop dengan `for` dapat dibatasi berapa kali iterasi yang harus dilalui sebelum berhenti.

```
for ( init; condition; increment )  
{  
    kode;  
}
```

Untuk mengisi posisi init, condition dan increment lebih mudah jika langsung melihat contoh penggunaannya.

```
1  using System;
2  
   0 references
3  public class Logika7
4  {
   0 references
5      public static void Main()
6      {
7          // ( init;          condition;  increment )
8          for (int a = 10;    a < 20;    a = a + 1)
9          {
10             Console.WriteLine("Iterasi ke - " + a);
11         }
12     }
13 }
```



Pada posisi init, kita mendeklarasikan variable yang akan digunakan untuk di evaluasi pada posisi condition.

Pada posisi condition, variable yang dideklarasikan pada posisi init akan dievaluasi setiap sebelum iterasi atau loop di jalankan.

Jika menghasilkan TRUE iterasi akan berlanjut, dan jika FALSE iterasi akan berhenti. Biasanya kondisi berisikan pengecekan variable deklarasi lebih kecil dari nilai stop.

Pada posisi increment, nilai varibale pada posisi init akan ditambahkan sebanyak 1 atau sesuai kebutuhan. increment akan dijalankan setelah iterasi selesai atau sebelum variable deklarasi di evaluasi pada posisi condition.

C# Programming Logic - Sesi 03


Do..while

Pada `for` dan `while` kondisi di evaluasi atau dicek diawal atau sebelum iterasi di mulai. `do..while` melakukan evaluasi kondisi pada akhir iterasi.

`do..while` sendiri mirip sekali dengan `while` yang membedakannya hanyalah posisi evaluasi atau pengecekan kondisinya.

Karena `do..while` melakukan evaluasi pada akhir iterasi atau loop, maka dipastikan pasti akan menjalankan setidaknya satu kali iterasi.

```
do
{
    kode;
} while( condition );
```

```
1 using System;
2 
3 0 references
4 public class Logika8
5 {
6     0 references
7     public static void Main()
8     {
9         int i = 0;
10
11         do
12         {
13             Console.WriteLine("i = {0}", i);
14             i++;
15
16             if (i > 5)
17                 break;
18         } while (i < 10);
19     }
20 }
```

```
Maliks-MacBook-Air:Sesi3 swijaya$ cd "/Users/
i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
Maliks-MacBook-Air:Sesi3 swijaya$ █
```

