



## MODUL – 4

### Penjelasan Pernyataan Penyeleksian IF

IF adalah salah satu pernyataan penyeleksian yang memungkinkan kita memanipulasi aliran jalannya program berdasarkan *conditional expression*. Hal ini dapat memungkinkan kita membuat program yang berjalan secara fleksibel sesuai keadaan dari pengguna dan mesin.

Penyeleksian **if** adalah pernyataan penyeleksian yang mencari kebenaran dari *conditional expression* yang disebutkan. *conditional expression* harus berupa bilangan *Boolean* atau operasi yang menghasilkan bilangan *Boolean* dan menyatakan benar atau salah atas *expression* tersebut.

Ketika mesin eksekusi bertemu dengan penyeleksian *if* maka CPU akan memeriksa kebenaran dari *conditional expression* yang disebutkan, jika benar (*true*) maka perintah yang ada di dalamnya akan dijalankan, jika salah (*false*) maka akan memeriksa pernyataan **else if** (jika ada), hal itu dilakukan berulang satu demi satu hingga menemukan kondisi yang bernilai benar (*true*). Jika tidak ditemukan maka akan melakukan perintah pernyataan **else**. Jika **else** tidak ada maka pernyataan **if** secara keseluruhan akan diabaikan.

```
1
2   if(Kondisi) {
3       Pernyataan;
4   }else if(Kondisi) {
5       Pernyataan;
6   }else{
7       Pernyataan;
8   }
```

Secara keutuhan pernyataan penyeleksian **if** memiliki bentuk struktur seperti di atas, dan juga ada bagian yang termasuk sebagai opsional (anda bebas untuk menyertakan atau tidak) yaitu *else if* dan *else*. Dan setiap bentuk dari pernyataan *if* memiliki nama-nama yang unik.

### Bentuk Pernyataan IF

```
1   if(Kondisi) {
2       Pernyataan;
```



```
2    }
```

```
3
```

### Bentuk

```
1    if (Kondisi) {  
2        Pernyataan;  
3    } else {  
4        Pernyataan;  
5    }
```

### Bentuk Pernyataan IF-ELSE Majemuk

```
1    if (Kondisi) {  
2        Pernyataan;  
3    } else if (Kondisi) {  
4        Pernyataan;  
5    } else {  
6        Pernyataan;  
7    }
```

### Pernyataan IF-ELSE

jika badan pernyataan merupakan pernyataan tunggal pada badan pernyataan *if* kita tidak diwajibkan menggunakan sepasang tanda kurung kurawal `{ }`.

```
1    if (Kondisi)  
2        Pernyataan;  
3    else if (Kondisi)  
4        Pernyataan;  
5    else  
6        Pernyataan;
```

Hal itu dapat dilakukan, [kompilator](#) akan menganggap bahwa satu pernyataan setelah pernyataan `if`, `else if` atau `else` akan dianggap sebagai *substatement* atau badan dari pernyataan tersebut.

### **Baca :** [Pengertian dan Fungsi Function Overloading](#)

`Kondisi` setelah keyword `if` atau `else if` adalah tempat dimana anda menaruh sebuah nilai *boolean* atau ekspresi seperti perbandingan, logika dan lain-lain yang menghasilkan



bilangan *Boolean*. Badan pernyataan **if** akan dieksekusi tergantung pada kebenaran dari **kondisi**

Jika evaluasi ekspresi bernilai benar (*true*) sesuai kondisi maka program akan melaksanakan perintah (pernyataan) di dalamnya. Jika tidak maka akan diabaikan dan beralih ke pilihan lain atau keluar dari pernyataan penyeleksian **if**.

**else if** adalah pilihan alternative dari penyeleksian untuk mencari kondisi yang diinginkan, **else if** memiliki fungsi yang sama seperti pernyataan **if** dan diletakan setelah pernyataan **if**.

**else** adalah pilihan terakhir yang akan dijalankan jika semua pilihan tidak memiliki nilai benar (*true*) pada kondisi yang ada. **else** merupakan pernyataan opsional untuk digunakan berdasarkan kebutuhan, jika tidak menggunakan pernyataan **else** maka keseluruhan dari pernyataan **if** akan diabaikan.

```
1    #include <iostream>
2    using namespace std ;
3
4    int main() {
5        int nomer ;
6
7        cout<<"Masukkan [1/2/3] : ";cin>>nomer;
8        if (nomer<=0){
9            cout<<"kurang dari 1"<<endl;
10       }else if (nomer==1){
11           cout<<"1"<<endl;
12       }else if (nomer==2){
13           cout<<"2"<<endl;
14       }else if (nomer==3){
15           cout<<"3"<<endl;
16       }else{
17           cout<<"lebih dari 3"<<endl;
```



```
17     }  
18     return 0 ;  
19 }  
20
```

Cara kerja CPU dalam melaksanakan pernyataan `if` adalah; pertama CPU masuk dan mengevaluasi ekspresi yang tertera pada kondisi pernyataan `if`, jika bernilai benar (*true*) maka pernyataan-pernyataan di dalam pernyataan `if` akan dijalankan, setelah dijalankan maka CPU akan keluar dari keseluruhan pernyataan `if`.

Jika salah (*false*) dan disertakan pernyataan `else if` maka CPU akan mengevaluasi ekspresi pada `else if`, jika bernilai benar (*true*) maka akan dijalankan, jika salah (*false*) akan beralih ke-lainnya, hal itu terjadi berulang-ulang hingga menemukan kondisi yang bernilai benar. jika dari semua pilihan tersebut tidak ada yang memiliki nilai benar (*true*) maka pernyataan secara keseluruhan `if` akan diabaikan atau menjalankan pernyataan `else` jika disertakan.

**Baca :** [Pernyataan Break](#)

CPU akan membaca program dari atas kebawah, jika telah menemukan kondisi yang benar (*true*) maka akan dijalankan dan keluar dari keseluruhan pernyataan `if`, tidak peduli jika ada pilihan alternative lainnya yang memiliki nilai benar (*true*) pada kondisi yang dimilikinya.

## Nested IF

*Merupakan hal yang dimungkinkan dalam bahasa pemrograman C++ yaitu membuat pernyataan IF di dalam pernyataan IF. hal ini dapat memungkinkan anda untuk membuat tahapan penyeleksian yang berlipat-lipat.*

Contoh Program :



### Pengantar Operasi Penyeleksian

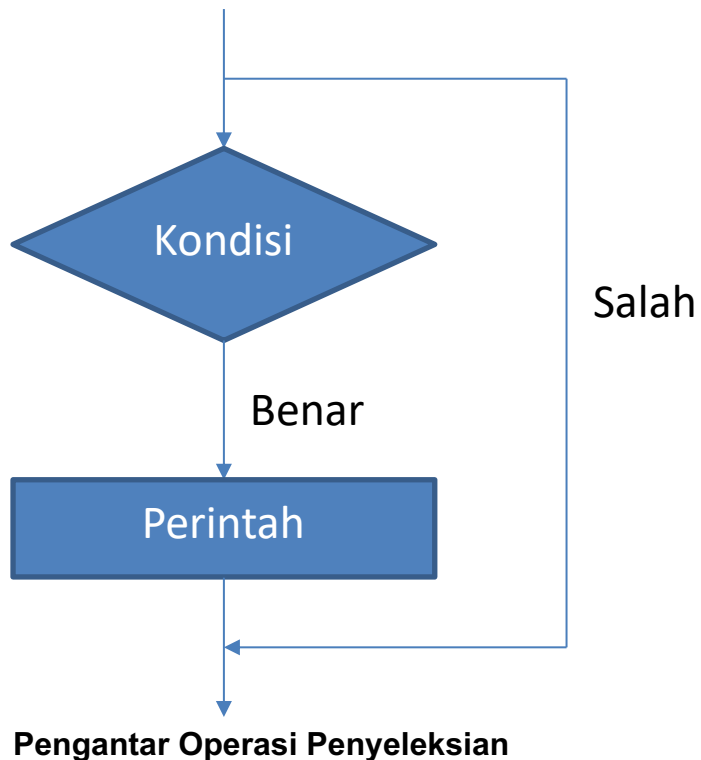
- Pernyataan percabangan digunakan untuk memecahkan persoalan untuk mengambil suatu keputusan diantara sekian pernyataan yang ada
- Dalam bahasa C++ terdapat beberapa perintah, diantaranya adalah :
- Pernyataan IF
- Pernyataan IF – ELSE
- Pernyataan NESTED IF
- Pernyataan IF – ELSE Majemuk
- Pernyataan SWITCH - CASE

### Pernyataan IF (lanjutan)

- Cara Pendeklarasian :
- Penulisan kondisi harus dalam tanda kurung dan berupa ekspresi relasi dan penulisan pernyataan dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong
- Jika pernyataan IF diikuti dengan pernyataan majemuk, bentuk penulisan sebagai berikut :

### Pernyataan IF

- Pernyataan IF mempunyai pengertian jika kondisi bernilai benar, maka perintah akan dikerjakan dan jika tidak memenuhi syarat maka akan diabaikan.
- Pengertian tersebut dapat dilihat dari diagram alir, berikut ini :



- Pernyataan percabangan digunakan untuk memecahkan persoalan untuk mengambil suatu keputusan diantara sekian pernyataan yang ada
- Dalam bahasa C++ terdapat beberapa perintah, diantaranya adalah :

- Pernyataan IF
  - Pernyataan IF – ELSE
  - Pernyataan NESTED IF
- Pernyataan IF – ELSE Majemuk
  - Pernyataan SWITCH - CASE

#### Pernyataan IF (lanjutan)

- Cara Pendeklarasian :
  - Penulisan kondisi harus dalam tanda kurung dan berupa ekspresi relasi dan penulisan pernyataan dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong
  - Jika pernyataan IF diikuti dengan pernyataan majemuk, bentuk penulisan sebagai berikut :

#### Pernyataan IF (lanjutan)



- Contoh :
- Menentukan besarnya potongan dari pembelian barang yang diberikan seorang pembeli, dengan kriteria :
  - Tidak ada potongan jika total pembelian kurang dari Rp 50.000,-
  - Jika total pembelian lebih dari atau sama dengan Rp 50.000,- potongan yang diterima sebesar 20% dari total pembelian

#### Pernyataan IF (lanjutan)

```
aritmatika1.cpp  majemuk.cpp  majemuk1.cpp  penaikan.cpp  depan.cpp  if.cpp
1  #include<stdio.h>
2  #include<conio.h>
3  #include<iostream.h>
4
5  main()
6  {
7      double tot_beli, potongan = 0, jum_bayar = 0;
8
9      cout<<"Total Pembelian Rp. ";cin>>tot_beli;
10
11     if(tot_beli >= 50000)
12         potongan = 0.2 * tot_beli;
13
14     cout<<"besarnya potongan Rp. "<<potongan<<endl;
15
16     jum_bayar = tot_beli - potongan;
17
18     cout<<"jumlah yang harus dibayarkan Rp. "<<jum_bayar;
19
20     getch;
21 }
```

#### Pernyataan IF – ELSE (lanjutan)

- Bentuk umum pernyataan if-else
- Perintah-1 dan perintah-2 dapat berupa sebuah pernyataan tunggal, pernyataan majemuk atau pernyataan kosong

Jika pemakaian if-else diikuti dengan pernyataan majemuk, bentuk penulisannya sebagai berikut

#### Pernyataan NESTED IF



- NESTED IF merupakan pernyataan if yang berada di dalam pernyataan if yang lainnya.
- Bentuk penulisan pernyataan NESTED IF :

```
if (syarat)
```

```
if(syarat)
```

```
....perintah ;
```

```
else
```

```
....perintah;
```

```
If(syarat)
```





## Pernyataan IF – ELSE Majemuk (lanjutan)

- Bentuk penulisan IF – ELSE

```
if (syarat)
{
    ....perintah ;
    ....perintah;
}

Else If(syarat)
{
```

### penyeleksian

dua macam pernyataan penyeleksian

ada pernyataan penyeleksian, yang menentukan alanya eksekusi program berdasarkan suatu pilihan yang terdiri dari pernyataan penyeleksian berupa bilangan Boolean, jika kondisi penyeleksian dijalankan jika 0 (*false*) maka akan dijalankan *else* atau *else if*.



## Macam-macam Pernyataan Penyeleksian

Dan untuk melakukan penyeleksian terdapat dua macam pernyataan penyeleksian yaitu `if` dan `switch`

### Pernyataan `if`

Merupakan salah satu pernyataan dari dua pernyataan penyeleksian, yang memungkinkan kita untuk memanipulasi aliran jalanya eksekusi program berdasarkan kondisi program pada saat itu. Dapat membuat banyak pilihan yang terdiri dari pernyataan tunggal atau majemuk. Kondisi dari pernyataan `if` berupa bilangan Boolean, jika kondisi sesuai `1` (*true*) maka pilihan tersebut akan dijalankan jika `0` (*false*) maka akan mengabaikan atau memilih pilihan alternative yaitu `else` atau `else if`.

### Pernyataan `switch`

#### Pernyataan IF – ELSE Majemuk (lanjutan)

- **PerSwitch – case** merupakan pernyataan yang dirancang khusus untuk menangani pengambilan keputusan yang melibatkan sejumlah atau banyak alternative
- **Pernyataan switch – case** memiliki kegunaan yang sama seperti `if-else` bertingkat, tetapi penggunaannya untuk memeriksa data yang bertipe karakter atau integer

nyataan `switch` – case

## PERBEDAAN IF DAN SWITCH

### **Flow Control ( IF Dan Switch)**

Flow control (struktur kendali) dapat dibagi menjadi dua jenis yaitu , Struktur percabangan dan pengulangan (looping). Namun kali ini akan membahas struktus percabangan.

Percabangan adalah perintah yang memungkinkan pemilihan atas perintah yang akan dijalankan sesuai dengan kondisi tertentu yang menentukan alur perjalanan



program. Percabangan bertujuan untuk memilih atau mengkondisikan apakah statement tersebut akan dijalankan atau tidak. Percabangan seperti operasi logika, jika diberi sebuah pernyataan, dan pernyataan tersebut benar, maka program akan menjalankan statement-statement yang telah ditentukan. Ada tiga macam perintah percabangan, yaitu if, if ... else, dan switch.

### **1. IF**

If digunakan untuk satu kondisi saja. Jika pernyataan benar (terpenuhi) maka akan dijalankan, jika salah (tidak terpenuhi) maka diabaikan.

### **2. IF ... ELSE**

Perintah ini digunakan untuk lebih dari satu kondisi. Seperti biasa, perintah1 dan perintah2 bisa berbentuk blok yang terdiri dari beberapa perintah. Pernyataan if merupakan bentuk percabangan 2 arah, jika kondisi yang diuji tersebut terpenuhi, maka program akan menjalankan pernyataan-pernyataan tertentu. Jika kondisi yang diuji salah, program akan menjalankan pernyataan yang lain.

### **3. SWITCH**

Perintah ini digunakan sebagai alternatif pengganti dari sintaks if ... else dengan else lebih dari satu. Switch, kondisi hanya dinyatakan dengan bilangan bulat atau karakter/string. Dengan perintah ini program percabangan akan semakin mudah dibuat dan dipelajari. Perintah switch akan menyeleksi kondisi yang diberikan dan kemudian membandingkan hasilnya dengan konstanta-konstanta yang berada di case.

Pembandingan akan dimulai dari konstanta 1 sampai konstanta terakhir. Jika hasil dari kondisi terpenuhi dengan nilai konstanta tertentu, misalnya konstanta 1, maka pernyataan 1 akan dijalankan sampai ditemukan break. Pernyataan break akan membawa proses keluar dari perintah switch, agar eksekusi dilakukan pada kondisi yang terpenuhi saja, jika telah terpenuhi maka dihentikan. Jika hasil dari kondisi tidak terpenuhi dengan konstanta-konstanta yang diberikan, maka pernyataan pada default akan dijalankan.

### **“ IF Vs SWITCH “**

1. Apa perbedaan if dengan switch ?

Jawab :

Perbedaan if-else dan switch case :



- Switch, kondisi hanya dinyatakan dengan bilangan bulat atau karakter/string sedangkan if-else yang dapat menggunakan operasi seperti <, >, <= dan >=.
- Tidak ada konstanta yang sama dalam sebuah switch.
- Angka yang digunakan harus berkisar diantara 0 – 255.
- Harus menggunakan break, Gunanya untuk mengakhiri statement.
- Ada default, ini bisa dibilang sebagai 'Else'nya Switch Case.

2. Kapan Menggunakan If ? Kapan Menggunakan Switch ?

Jawab :

IF digunakan apabila seleksi memiliki lebih dari 1 kondisi dan nilai.

SWITCH digunakan apabila hanya ada 1 kondisi dengan nilai yang bervariasi.

Berikut contoh Program If dan Switch Dalam :

**A. BAHASA C#**

**using** System;

**namespace** Ifswitch

{

**public** class biodata

    {

**public** int pilih;

**public** void tampilkan()

    ,

    {

**switch** (pilih)

        {

**case** 1:

                Console.WriteLine("Ini Satu"); break;

**case** 2:

                Console.WriteLine("Ini Dua"); break;

**default:**

                Console.WriteLine("Selain Satu atau Dua"); break;

        }

    }

**public** void tampilkan\_if ()

    {

**if** (pilih == 1)

            Console.WriteLine("Ini Satu");



```
    else if (pilih == 2)
        Console.WriteLine("Ini Dua");
    else
        Console.WriteLine("Selain Satu atau Dua");
    }
}
}
//Pemanggilan Pada Program Utama
using System;
namespace Ifswitch
{
    class Program
    {
        public static void Main(string[] args)
        {
            biodata b = new biodata();

            b.pilih = 1;

            b.tampilkan();
            b.tampilkan_if();

            Console.ReadKey(true);
        }
    }
}
```

## B. BAHASA C++

```
#include <iostream>
using namespace std;
class kondisi_switch
{
public:
    int pilih;
    void tampilkan()
    {
        switch (pilih)
        {
            case 1:
                cout << "Ini Satu" << endl; break;
            case 2:
                cout << "Ini Dua" << endl; break;
            default:
                cout << "Selain Satu atau Dua" << endl; break;
        }
    }
}
```



```
    }  
  }  
  void tampilkan_if ()  
  {  
    if (pilih == 1)  
      cout << "Ini Satu" << endl;  
    else if (pilih == 2)  
      cout << "Ini Dua" << endl;  
    else  
      cout << "Selain Satu atau Dua" << endl;  
  }  
  //Pemanggilan Pada Program Utama  
  int main()  
  {  
    kondisi_switch k;  
    k.pilih = 1;  
    k.tampilkan();  
    k.tampilkan_if();  
  }  
}
```

## Pengertian Kondisi IF ELSE IF bahasa C

Pada dasarnya, kondisi **IF ELSE IF** adalah sebuah struktur logika program yang di dapat dengan cara menyambung beberapa kondisi **IF ELSE** menjadi sebuah kesatuan.

Jika kondisi pertama tidak terpenuhi atau bernilai **false**, maka kode program akan lanjut ke kondisi **IF** di bawahnya. Jika ternyata tidak juga terpenuhi, akan lanjut lagi ke kondisi **IF** di bawahnya, dst hingga blok **ELSE** terakhir atau terdapat kondisi **IF** yang bernilai **true**.

Berikut format dasar penulisan kondisi **IF ELSE IF** dalam bahasa C:

```
if (condition_1) {  
  
    // Kode program yang dijalankan jika condition_1 berisi nilai True  
  
}  
  
else if (condition_2) {  
  
    // Kode program yang dijalankan jika condition_2 berisi nilai True  
  
}  
  
else if (condition_3) {  
  
    // Kode program yang dijalankan jika condition_3 berisi nilai True  
  
}
```



```
}  
  
else {  
  
    // Kode program yang dijalankan jika semua kondisi tidak terpenuhi  
  
}
```

Mari kita lihat contoh prakteknya.

## Contoh Kode Program Percabangan IF ELSE IF Bahasa C

Sebagai contoh pertama, saya ingin membuat sebuah kode program untuk menampilkan nilai. User diminta menginput sebuah huruf antara 'A' – 'E'. Kemudian kode program akan menampilkan hasil tampilan yang berbeda-beda untuk setiap huruf yang diinput, termasuk jika huruf tersebut di luar 'A' – 'E'.

Berikut contoh kode programnya:

```
1    #include <stdio.h>  
2  
3    int main(void)  
4    {  
5        char nilai;  
6  
7        printf("Input Nilai Anda (A - E): ");  
8        scanf("%c",&nilai);  
9  
10       if (nilai == 'A') {  
11           printf("Pertahankan! \n");  
12       }  
13       else if (nilai == 'B') {  
14           printf("Harus lebih baik lagi \n");  
15       }  
16   }
```



```
14     else if (nilai == 'C') {
15         printf("Perbanyak belajar \n");
16     }
17     else if (nilai == 'D') {
18         printf("Jangan keseringan main \n");
19     }
20     else if (nilai == 'E') {
21         printf("Kebanyakan bolos... \n");
22     }
23     else {
24         printf("Maaf, format nilai tidak sesuai \n");
25     }
26     return 0;
27 }
28
```

#### Hasil kode program:

Input Nilai Anda (A - E): A

Pertahankan!

Input Nilai Anda (A - E): D

Jangan keseringan main

Input Nilai Anda (A - E): E

Kebanyakan bolos...

Input Nilai Anda (A - E): F

Maaf, format nilai tidak sesuai





Di baris 5 saya mendefinisikan sebuah variabel **nilai** sebagai **char**. Variabel **nilai** ini kemudian dipakai untuk menampung input dari perintah **scanf** di baris 8.

Mulai dari baris 10 hingga 27, terdapat 5 buah pemeriksaan kondisi, yakni satu untuk setiap block **IF ELSE**. Dalam setiap kondisi, isi variabel **nilai** akan di diperiksa, apakah itu berupa karakter 'A', 'B', hingga 'E'. Jika salah satu kondisi ini terpenuhi, maka block kode program yang sesuai akan di eksekusi.

Jika ternyata nilai inputan bukan salah satu dari karakter 'A' – 'E', maka **block ELSE** di baris 25 lah yang akan dijalankan.