

KONDITIONAL

OBJEKTIF :

1. Mahasiswa Mampu Memahami Tentang IF Statement.
 2. Mahasiswa Mampu Memahami Tentang IF – Else dan Nested IF.
 3. Mahasiswa Mampu Memahami Tentang Ternary Operator.
 4. Mahasiswa Mampu Memahami Tentang Go To.
-

1.1 IF STATEMENT

IF Statement digunakan untuk penyeleksian program berdasarkan kondisi. Jika kondisi bernilai benar maka statement yang mengikutinya akan diproses, tapi jika bernilai salah maka statement lain yang akan di proses. Terdapat beberapa jenis dalam IF Statement, diantaranya adalah IF Sederhana, IF – Else, Nested IF (IF Bersarang), dan IF Bertingkat.

1.1.1 IF SEDERHANA

Sintaks dari bentuk IF Sederhana adalah sebagai berikut :

```
IF (kondisi)
{
    statement 1
    statement 2
}
```

Pada IF sederhana terdapat tanda () tempat kondisi di tuliskan. Lalu terdapat tanda { } tempat dimana statement ditulis. Kemudian terdapat statement yang merupakan proses yang terjadi apabila kondisi terpenuhi. Pada bentuk IF ini

dapat digunakan untuk memproses beberapa statemen sekaligus dalam suatu blok kondisi jika kondisi yang diseleksi benar.

Contoh :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int umur;
    printf("Masukan Umur Anda= ");
    scanf("%d", &umur);
    if (umur < 18)
    {
        printf("umur anda kurang dari 18 tahun");
    }
    if (umur > 18)
    {
        printf("umur anda lebih dari 18 tahun");
    }
    if (umur == 18)
    {
        printf("umur anda 18 tahun");
    }
    return 0;
}
```

1.2 IF – ELSE DAN NESTED IF

Sintaks dari bentuk IF - Else adalah sebagai berikut :

```
IF (kondisi)
{
    statement 1
}
Else
{
    Statement 2
}
```

Pada IF – Else jika kondisi bernilai benar maka statement 1 yang dijalankan, tetapi jika kondisi bernilai salah maka statement 2 yang dijalankan.

Contoh :

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    int umur;
    printf("Masukan Umur Anda= ");
    scanf("%d", &umur);
    if (umur < 18){
        printf("umur anda kurang dari 18 tahun");
    }
    else if (umur > 18){
        printf("umur anda lebih dari 18 tahun");
    }
    else{
        printf("umur anda 18 tahun");
    }
    return 0;}
```

Sintaks dari bentuk Nested IF adalah sebagai berikut :

```
If (kondisi 1)
{
    If (kondisi 2)
    {
        pernyataan 1;
    }
    else
    {
        pernyataan 2;
    }
}
else
{
    pernyataan 3;
}
```

Pada bentuk Nested IF, kondisi yang akan diseleksi pertama kali adalah kondisi yang paling luar (kondisi1). Jika bernilai tidak benar maka statement setelah else yang terluar (pasangan dari if yang bersangkutan) yang akan diproses.

Contoh :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int umur;
    printf("Masukan Umur Anda= ");
    scanf("%d", &umur);
    if (umur < 18)
    {
        printf("umur anda kurang dari 18 tahun");
    } else if (umur>18)
    {
        printf("umur anda lebih dari 18 tahun\n");
    }
}
```

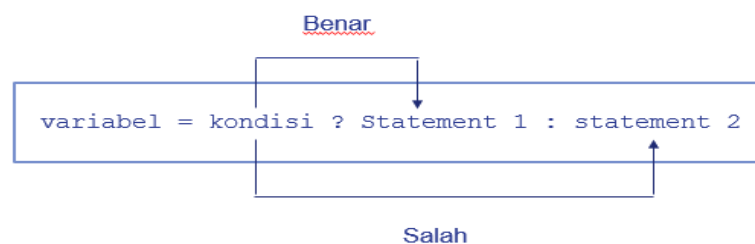
```
//nested if
if(umur>25)
{
    printf("umur anda lebih dari 18 tahun dan
           lebih dari 25 tahun");
} else
{
    printf("umur anda lebih dari 18 tahun
           tetapi kurang dari 25 tahun");
} else
{
    printf("umur anda 18 tahun");
}

return 0;
}
```

1.3 TERNARY OPERATOR

Operator Ternary digunakan untuk mengeksekusi code berdasarkan hasil dari binary condition. Operator Ternary memiliki kesamaan dengan IF - Else. Tetapi operator ternary memiliki struktur yang lebih singkat yaitu hanya 1 baris.

Struktur dari Penggunaan Operator Ternary adalah sebagai berikut :



Pada operator ternary terdapat kondisi lalu tanda tanya (?) kemudian statement 1 dan statement 2. jika kondisi variabel bernilai benar, maka statement 1 yang akan menjadi hasil dari variabel tetapi jika kondisi 1 bernilai salah, maka statement 2 yang akan menjadi hasil dari variabel.

Contoh :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int a,b,max;
    a = 25;
    b = 30;
    max=(b>a)? b : a;
    printf("nilai max %d\n",max);

    return 0;
}
```

Pada contoh diatas nilai variabel max adalah hasil jawaban dari kondisi b lebih besar dari a. Jika nilai b bernilai benar, maka nilai variabel max adalah b yang bernilai 30. Karena b merupakan statement pertama. Tetapi, jika nilai b bernilai salah, maka nilai variabel max adalah a yang bernilai 25 karena a adalah statement kedua.

1.4 GOTO

Goto statement digunakan oleh programmer untuk mengubah urutan eksekusi program, dengan menggeser kontrol bagian yang berbeda dari program. Statement goto akan mengarahkan eksekusi program ke pernyataan yang diawali oleh sebuah label.

Struktur dari statement Go To adalah sebagai berikut :

```
goto label;
    statement 1-----
    statement2-----
label:
```

Pada goto statement terdapat sebuah label,yaitu sebuah pengenalan atau identifier yang diakhiri dengan tanda titik dua (:). Saat sebuah program berjalan dan masuk dalam goto statement, maka program tersebut akan langsung diarahkan menuju label. Jadi, statement yang ada akan dilewati dan langsung menuju label.

Contoh :

```
#include <stdio.h>
#include <stdlib.h>

int main()
{
    goto labelku;
    printf("terlewati");
    labelku:
    printf("Ada bagian yang tidak ditampilkan?\n");
    return 0;
}
```

RANGKUMAN

1. IF Statement digunakan untuk penyeleksian program berdasarkan kondisi. Jika kondisi bernilai benar maka statement yang mengikutinya akan diproses, tapi jika bernilai salah maka statement lain yang akan di proses.
2. Statement goto dapat digunakan untuk melompat dari suatu proses ke bagian proses yang lainnya di dalam program.

TUNTUNAN LATIHAN

Berikut ini diberikan contoh program membandingkan 3 buah bilangan untuk mencari nilai max :

- (a) Mendeklarasikan pre-processor include diikuti dengan file judul atau header file. Caranya mengetikkan langsung pada lembar kerja, yaitu :

```
1  #include <stdio.h>
2  #include <conio.h>
3
```

- (b) Mendeklarasikan fungsi int main. Caranya mengetikkan langsung pada lembar kerja, yaitu :

```
4  int main ()
5  {
```

- (c) Mendeklarasikan variabel a,b,c dan max dengan tipe data integer. Cara mengetikkan langsung pada lembar kerja, yaitu :

```
6      int a,b,c,max;
7
```

- (d) Membuat format masukkan menggunakan fungsi printf dan scanf untuk memasukkan bilangan yang akan dibandingkan. Cara mengetikkan langsung pada lembar kerja, yaitu :

```
8      printf("masukkan bilangan pertama : ");
9      scanf("%i",&a);
10
11     printf("masukkan bilangan kedua : ");
12     scanf("%i",&b);
13
14     printf("masukkan bilangan ketiga : ");
15     scanf("%i",&c);
16
```

- (e) Membuat kondisi if untuk membandingkan ketiga bilangan dengan kondisi sebagai berikut :

- ➔ Jika bilangan a lebih besar dari bilangan b dan bilangan a lebih besar dari bilangan c, maka bilangan a adalah max.
- ➔ Jika bilangan b lebih besar dari bilangan a dan bilangan b lebih besar dari bilangan c, maka bilangan b adalah max.
- ➔ Jika bilangan c lebih besar dari bilangan a dan bilangan c lebih besar dari bilangan b, maka bilangan c adalah max.

Cara mengetikkan langsung di lembar kerja, yaitu :

```
17      if ((a>b) && (a>c))
18          max=a;
19      else if ((b>a) && (b>c))
20          max=b;
21      else if ((c>a) && (c>b))
22          max=c;
23
24      }
```

LATIHAN

1. Buatlah sebuah Program untuk mencetak bilangan terbesar dari 5 buah bilangan yang dimasukkan oleh user, dengan cara membandingkan bilangan sebelumnya dengan bilangan berikutnya dan tentukan apakah bilangan terbesar tersebut adalah bilangan negatif atau positif. Misalnya bilangan tersebut A, B, C, D, dan E maka A dan B diperbandingkan. Jika A lebih besar dari B maka A dibandingkan dengan C, jika A lebih besar dari C maka A dibandingkan dengan D, demikian seterusnya sampai didapat nilai yang terbesar.