

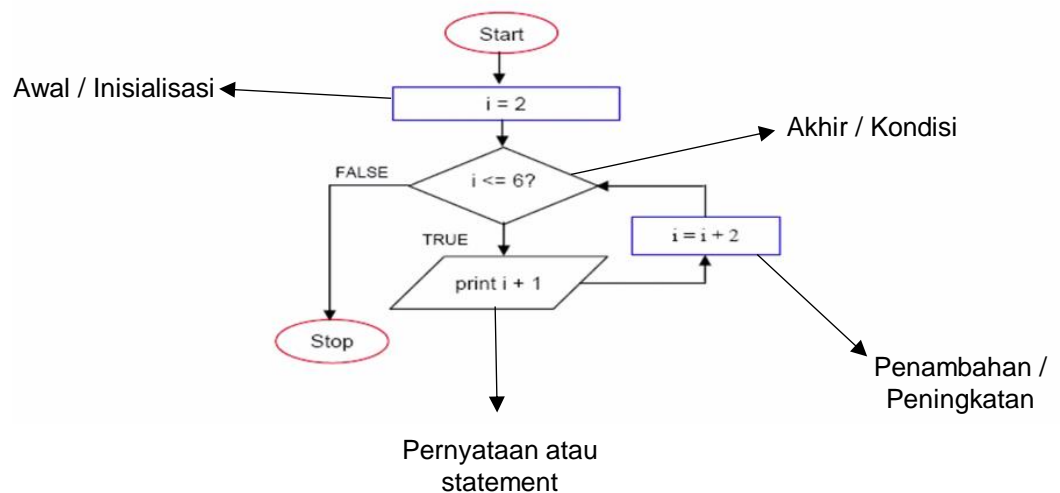
LOOP

3

OBJEKTIF :

1. Mahasiswa Mampu Memahami Tentang Perulangan dengan While.
2. Mahasiswa Mampu Memahami Tentang Perulangan dengan Do - While.
3. Mahasiswa Mampu Memahami Tentang Perulangan dengan For Loop.
4. Mahasiswa Mampu Memahami Tentang Switch Statement.

Untuk memudahkan dalam membuat keluaran yang berulang secara efisien, maka digunakan statement perulangan. Statement perulangan berguna agar program yang dibuat menjadi lebih praktis. Dengan menggunakan statement perulangan maka statement yang diproses akan diulang – ulang cukup dituliskan sekali saja. Terdapat beberapa jenis perulangan pada Bahasa C, yaitu While, Do – While, dan For.



1.1 While

Sintaks dari bentuk perulangan While adalah sebagai berikut :

```
while (kondisi)
{
    Statement;
    increment;
}
```

Statement dapat berupa statement kosong, statement tunggal atau blok statement yang akan diproses berulang – ulang. Proses perulangan akan masih dilakukan jika kondisi yang diseleksi di statement while masih bernilai benar dan perulangan akan dihentikan jika kondisinya sudah bernilai salah.

Contoh :

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

int main()
{
    int x,y;
    x=1;
    y=6;
    while (x<=y)
    {
        printf("perulangan while ke %d\n",x);
        x++;
    }
}
```

1.2 DO - WHILE

Sintaks dari bentuk perulangan Do - While adalah sebagai berikut :

```
do
{
    Statement;
    iterasi;
} while(kondisi);
```

Perbedaan antara statement do-while dengan statement while adalah letak dari kondisi yang akan diseleksinya. Untuk statement while, kondisi yang diseleksi terletak di awal lingkup perulangannya, sehingga sebelum masuk kedalam lingkup perulangan while, kondisinya harus sudah benar terlebih dahulu. Untuk statement do-while, kondisi yang diseleksi terletak diakhir lingkup perulangannya, ini berarti bahwa paling sedikit sebuah perulangan akan dilakukan oleh statement do-while, karena untuk masuk pertama kali ke lingkup perulangan ini tidak diseleksi lebih dulu.

Contoh :

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int n=1;
    int times = 5;
    do {
        printf("c do while loops : %d\n",n);
        n=n+1;
    } while(n <= times);

    return 0;
}
```

1.3 FOR LOOP

Sintaks dari bentuk perulangan For adalah sebagai berikut :

```
for(kondisi awal;kondisi;iterasi)
{
    Statement;
}
```

Perulangan FOR merupakan perulangan yang simple dan banyak digunakan karena strukturnya yang mudah dipahami. Awal adalah suatu ungkapan yang memberikan nilai awal suatu variable untuk pengulangannya (misalnya $x=0$). Variable ini disebut dengan indeks atau loop-control variable. Variable ini merupakan penghitung dari banyaknya perulangan yang akan dilakukan, akhir adalah suatu ungkapan yang menunjukkan suatu kondisi yang harus dipenuhi supaya perulangan dapat terus dilakukan (misalnya $x \leq 20$ berarti perulangan akan tetap dilakukan selama nilai variabel x masih lebih kecil atau sama dengan nilai 20). Peningkatan adalah suatu ungkapan yang merubah nilai – nilai variable pengontrol perulangan setiap perulangan dilakukan (misalnya $x++$ menunjukkan nilai x ditambah dengan 1 setiap saat perulangan dilakukan).

Contoh :

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    for(int x=1; x<=5; x=x+1)
    {
        printf("Nilai x adalah %d\n", x);
    }

    return 0;
}
```

1.4 SWITCH STATEMENT

Sintaks dari bentuk Switch Statement adalah sebagai berikut :

```
switch (variabel) {  
    case 1:  
        break;  
    case 2:  
        break;  
    case 3:  
        break;  
    .  
    .  
    default:  
        statement-statement;  
}
```

Switch atau dapat disebut dengan case statement merupakan sebuah percabangan dimana kita dapat membandingkan isi sebuah variabel dengan beberapa nilai.

Contoh :

```
#include<stdio.h>  
int main()  
{  
    int input;  
    printf("pilih number (1,2,3) : ");  
    scanf("%d", &input);  
    switch(input){  
    case 1:  
        printf("Anda telah memilih case 1.");  
        break;  
    case 2:  
        printf("Anda telah memilih case 2.");
```

```
        break;
    case 3:
        printf("Anda telah memilih case 3.");
        break;
    default:
        printf("Anda tidak memiliki pilihan di
                (1,2,3) .");
        break;
}
```

RANGKUMAN

1. Perulangan yang terdapat pada Bahasa C terdiri dari While, Do – While, dan For dimana ketiganya memiliki sintaks yang berbeda – beda.
2. Switch atau dapat disebut dengan case statement merupakan sebuah percabangan dimana kita dapat membandingkan isi sebuah variabel dengan beberapa nilai.

TUNTUNAN LATIHAN

Berikut ini diberikan contoh program untuk mencetak 10 bilangan ganjil pertama dan deret 10 bilangan genap pertama dengan menggunakan perulangan for :

- (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 ganjil dan genap dengan tipe data integer. Cara mengetikkan langsung pada lembar kerja, yaitu :

```

6 | int ganjil, genap;
7 |

```

- (d) Membuat perintah untuk mencetak text dan membuat perulangan dengan for untuk bilangan ganjil. Cara mengetikkan langsung pada lembar kerja, yaitu :

```

8 | printf("Deret 10 bilangan ganjil pertama : ");
9 | for(ganjil=0;ganjil<20;ganjil++)
10 | {
11 |     ganjil%2!=0;ganjil++;
12 |     printf(" %d ",ganjil);
13 | }
14 | printf("\n");
15 |

```

- (e) Membuat perintah untuk mencetak text dan membuat perulangan dengan for untuk bilangan genap. Cara mengetikkan langsung pada lembar kerja, yaitu :

```

16 | printf("Deret 10 bilangan genap pertama : ");
17 | for(genap=0;genap<20;genap++)
18 | {
19 |     genap%2==0;genap++;
20 |     printf(" %d ",genap+1);
21 | }
22 | return 0;
23 | }

```

LATIHAN

1. Lengkapilah program berikut untuk mencetak segitiga siku – siku dengan tinggi berdasarkan input !

```

#include <stdio.h>

int main(void)
{
    int jumlah;

    printf("Input tinggi segitiga: ");
    scanf("%i",&jumlah);
    printf("\n");

    /*buatlah perulangan for untuk mencetak tanda '*'
    yang membentuk segitiga siku - siku dengan
    tinggi berdasarkan angka yang dimasukkan */

    .....

    return 0;
}

```