

## BAB 4. PERULANGAN

Pernyataan perulangan digunakan untuk mengulang satu atau lebih pernyataan eksekusi. Pada C++ terdapat 3 macam bentuk pernyataan perulangan, yaitu :

### 1 Statemen FOR

Pernyataan **for** dipakai untuk mengulang pengekseskusion terhadap satu atau sejumlah pernyataan.

**Bentuk umum :**

```
for ( ungkapan_1; ungkapan_2; ungkapan_3)  
pernyataan;
```

#### **Keterangan :**

- *Ungkapan\_1*, merupakan inialisasi variabel kendali kalang.
- *Ungkapan\_2*, merupakan kondisi yang menentukan pengulangan atau merupakan tes kelanjutan kalang.
- *Ungkapan\_3*, merupakan pengatur / pemodifikasi nilai variabel kendali kalang.
- *Pernyataan*, merupakan pernyataan yang akan di ulang.

*Contoh 1:*

```
#include <stdio.h>  
main()  
{  
    int x;  
    for (x =1; x <=5; x++)  
        printf(“ %i ” x); }  
}
```

Output :

*Contoh 2 :*

```
#include <stdio.h>  
void main()  
{  
    int x;  
    for (x =1; x <=5; )  
        printf(“ %i ” ++x); }  
}
```

output :

#### **For Bersarang**

*Contoh 3 :*

```
#include <stdio.h>  
void main()  
{  
    int x,j ;  
    for (x = 1 ; x <= 3 ; ++x ) {  
        for (j = 1; j <= x; j++ )  
            printf(“ %d “,j);  
        printf(“\n”);  
    }  
}
```

Ouput :

## 2. Statemen WHILE

Pernyataan while merupakan kalang kondisional yang dipakai untuk mengulang pengekseskuan terhadap satu atau sejumlah pernyataan selama kondisi terpenuhi. Pengujian terhadap kondisi dilakukan di awal kalang.

**Bentuk umum :**

```
while (kondisi)
{
    pernyataan_1;
    pernyataan_2;
    .....
    pernyataan_n; }

```

### **Keterangan :**

Pada bagian pernyataan\_1 hingga pernyataan\_n ada kemungkinan tidak akan dieksekusi sama sekali ( kondisi awal tidak terpenuhi ).

Contoh4:

```
#include <stdio.h>
main()
{   int x= 1;
    while (x <= 5) {
        printf("%i ", x );
        x++; }
}
```

Output :

## 3. Statemen DO\_WHILE

Pernyataan do\_while merupakan kalang kondisional yang dipakai untuk mengulang pengekseskuan terhadap satu atau sejumlah pernyataan. Pengujian terhadap kondisi dilakukan di akhir kalang. Kalang ini melakukan iterasi/pengulangan setidaknya satu kali.

**Bentuk umum :**

```
do
{
    pernyataan_1;
    pernyataan_2;
    .....
    pernyataan_n;
} while (kondisi) ;

```

### **Keterangan :**

- Kalang ini melakukan iterasi/pengulangan setidaknya satu kali. Bagian pernyataan\_1 hingga pernyataan\_n dieksekusi setidaknya satu kali. Selanjutnya, akan dieksekusi jika selama kondisi terpenuhi.

**Contoh 5:**

```

#include <stdio.h>
main()
{   int x= 1;
    while (x <= 5) {
        printf(“%i “, x );
        x++;   }
}

```

### a. Statemen Continue

Menyebabkan proses perulangan ke awal mulainya perulangan dengan mengabaikan statement-statement berikutnya setelah continue. Dapat digunakan untuk perulangan for, while dan do-while.

Contoh :

```

#include<stdio.h>
main()
{   int I, N; N = 2;
    for(I = 1; I<5 ; I++) {
        N= N * 2 ;
        if (N>16) continue;
        N = N + 1 ;   }
    Printf(“Nilai N adalah %d”,N);
}

```

Output : Nilai N adalah 44

Penjelasan :      I          N = N\*2          N=N+1

### b. Statemen Break

Selain digunakan pada pernyataan switch, break juga digunakan pada pernyataan for, while dan do-while. Kegunaanya untuk memaksa keluar dari pernyataan-pernyataan itu.

Contoh:

```

#include<stdio.h>
main()
{   int I, N; N = 2;
    for(I = 1; I<5 ; I++) {
        N= N * 2 ;
        if (N>16) break;
        N = N + 1 ;   }
    printf(“Nilai N adalah %d”,N);
}

```

**Output : Nilai N adalah 22**

**Penjelasan : I      N = N\*2      N=N+1**

**c. Statemen goto**

Dgunakan untuk melompat dari suatu proses ke bagian proses yang lain di dalam program.

Bentuk umum : goto label;

Contoh goto :

```
#include<stdio.h>
main()
{   int A, B;
    printf("Masukkan Nilai A ? ");scanf("%d",&A);
    printf("Masukkan Nilai B ? ");scanf("%d",&B);
    if(B= =0) goto tak_berhingga ;
    printf("%d dibagi dengan %d adalah %d \n", A,B,A/B);
    goto selesai ;

    tak_berhingga: /*label*/
        printf("%d dibagi dengan nol adalah tak berhingga",A);

    selesai : /*label*/
        ; //statement kosong hanya diisi dengan titik koma.
}
```

**Output :**

```
Masukkan Nilai A ? 4
Masukkan Nilai B ? 2
4 dibagi dengan 2 adalah 2
```

```
Masukkan Nilai A ? 4
Masukkan Nilai B ? 0
4 dibagi dengan nol adalah tak berhingga
```

Penjelasan : Bila B bernilai 0 maka program akan melompat menuju label tak\_berhingga. Dan bila B tidak sama dengan nol statement yang dijalankan adalah printf("%d dibagi dengan %d adalah %d \n", A,B,A/B); goto selesai ;