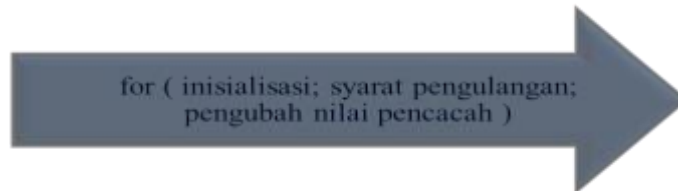


BAB 5

PROSES PERULANGAN

A. Statement FOR

Statement FOR digunakan untuk menyatakan perulangan pernyataan FOR:



Jika FOR lebih dari satu maka harus di letakan di dalam tanda kurung pernyataan:

for (inisialisasi; syarat pengulangan; pengubah nilai pencacah)

```
{  
pernyataan / perintah;  
pernyataan / perintah;  
pernyataan / perintah;  
}
```

- ✓ Inisialisas : Merupakan bagian untuk memberikan nilai awal untuk variable variabel tertentu.
- ✓ Syarat Pengulangan : Memegang kontrol terhadap pengulangan, karena bagian ini yang akan menentukan suatu perulangan diteruskan atau dihentikan.
- ✓ Pengubah Nilai Pencacah : Mengatur kenaikan atau penurunan nilai pencacah.

Contoh-1

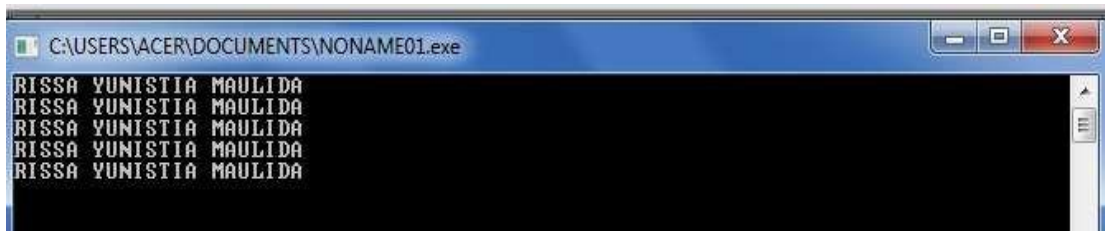
```
#include<stdio.h>  
#include<conio.h>  
#include<iostream.h>  
main( )  
{  
int a;  
clrscr( );  
for(a = 1; a <= 5; ++a)  
cout<<"RISSA YUNISTIA MAULIDA"<<endl;
```

```

getch( );
}

```

Maka tampilan nya akan seperti ini :



Gambar 5.1

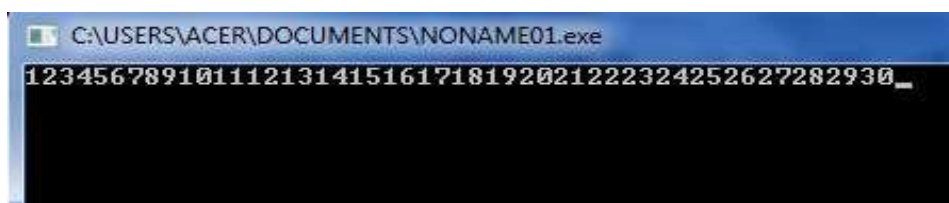
Contoh-2

```

#include<stdio.h>
#include<conio.h>
#include<iostream.h>
main( )
{
int a;
clrscr( );
for(a = 1; a <= 30; ++a)
cout<<a;
getch( ); }

```

Hasil Outputnya :



Gambar 5.2

B. Pernyataan Nested - For

Pernyataaan Nested *for* adalah suatu perulangan *for* didalam perulangan *for* yang lainnya. Bentuk umum pernyataan *Nested for* sebagai berikut :

```

for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )
{
    for ( inisialisasi; syarat pengulangan; pengubah nilai pencacah )

```

```

{
    pernyataan / perintah;
}
}

```

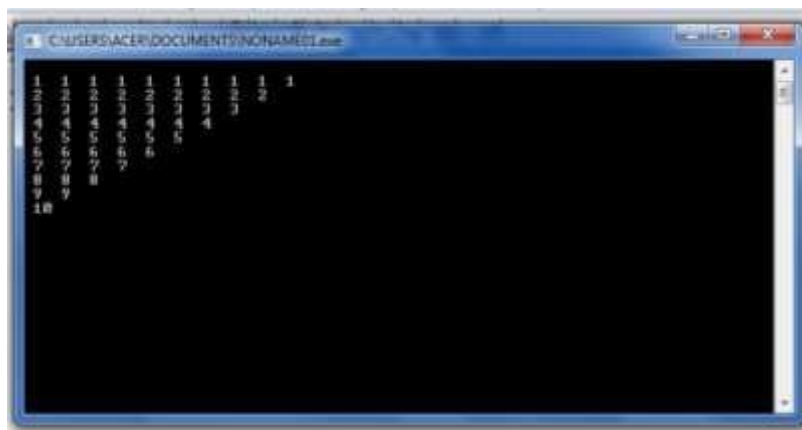
Contoh-3

```

#include<stdio.h>
#include<conio.h>
main( )
{
    int a, b;
    clrscr( );
    for(a = 1; a <= 10; a++)
    {
        printf("\n");
        for(b = a; b <= 10; b++)
            printf(" %d ",a);
    }
    getch(); }

```

Maka tampilanya akan seperti ini



Gambar 5.3

C. Perulangan Tak Berhingga

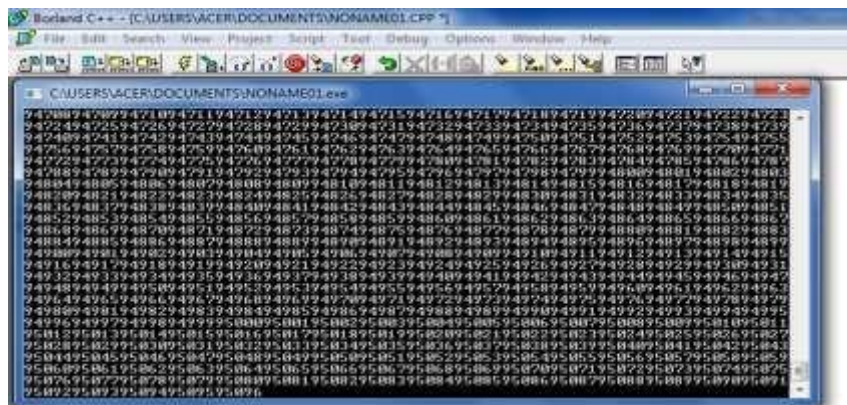
Perulangan tak berhingga merupakan perulangan (loop) yang tak pernah berhenti atau mengulang terus, hal ini sering terjadi disebabkan adanya kesalahan penanganan kondisi yang dipakai untuk keluar dari loop.

Contoh -4

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

main( )
{
    int bil;
    clrscr( );
    for (bil = 100; bil >=10; bil--)
        printf("%d", bil);
    getch( );
}
```

Maka yang akan di tampilkan nya :



Gambar 5.4

Pada pernyataan ini tidak akan berhenti untuk menampilkan bilangan menurun, kesalahan terjadi pada pengubah nilai pencacah, seharusnya penulisan yang benar berupa

bil - -

Akan tetapi yang ditulis adalah :

bil ++

D. Pernyataan While

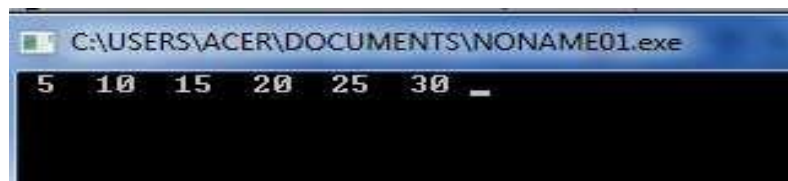
Pernyataan while merupakan salah satu pernyataan yang berguna untuk memproses suatu pernyataan atau beberapa pernyataan beberapa kali. Pernyataan while memungkinkan statemen-statemen yang ada didalamnya tidak dilakukan sama

```
while ( syarat )  
{  
    Pernyataan / perintah ;  
    Pernyataan / perintah ;  
}
```

Contoh-5

```
#include <stdio.h>  
#include <conio.h>  
main( )  
{  
    int bil=5;  
    clrscr( );  
    while(bil<=30)  
    {  
        printf(" %d ",bil);  
        bil+=5; }  
    getch( ); }
```

Maka akan di tampilkan nya :



Gambar 5.5

E. Statement Do...While Karakteristik

- Perulangan akan dilakukan minimal 1x terlebih dahulu, kemudian baru dilakukan pengecekan terhadap kondisi, jika kondisi benar maka perulangan masih akan tetap dilakukan.
- Perulangan dengan do...while() akan dilakukan sampai kondisi false

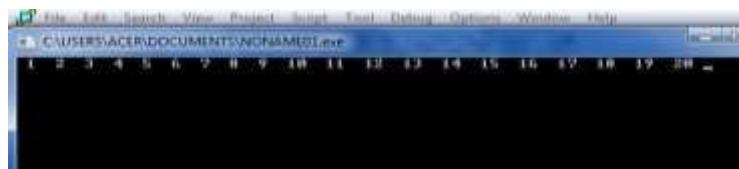
Contoh-6

```

#include <stdio.h>
#include <conio.h>
main( )
{
    int bil=1;
    clrscr( );
    while(bil<=20)
    {
        printf(" %d ",bil);
        ++bil; }
    getch( ); }

```

Output yang akan dihasilkan :



Gambar 5.6

F. Pernyataan Break

Pernyataan *break* ini berfungsi untuk keluar dari struktur *switch*. Dan pernyataan *break* juga berfungsi keluar dari perulangan (*for*, *while* dan *do-while*).

Contoh-7

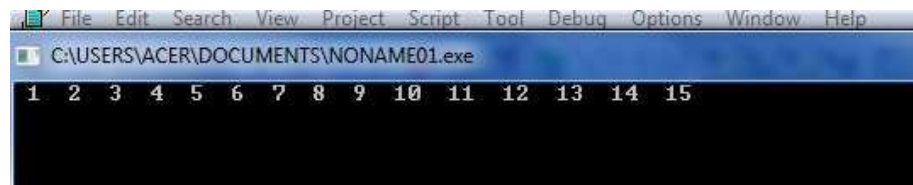
```

#include <stdio.h>
#include <conio.h>
main( )
{
    int bil = 1;
    clrscr( );
    do
    {
        if (bil >= 6)
            break;
        printf(" %d ",bil);
    }
}

```

```
while(bil++);
getch( );
}
```

Output yang akan dihasilkan :



Gambar 5.7

LATIHAN 5

1. Buatlah program untuk menampilkan bilangan ganjil dari kecil ke besar sampai batas tertentu sampai menemukan break, program selesai.
2. Bulatlah program untuk menghitung 20 deret bilangan genap dengan hasilnya :

$$2 + 4 + 6 + 8 + 10 + 12 + 14 + 16 + 18 + 20 + 22 + 24 + 26 + 28 + 30 + 32 + 34 + 36 + 38 + 40 = 420$$