

MODUL 6 PENGULANGAN 1

1. Deskripsi Singkat

Dalam modul ini, akan dipelajari Struktur pengulangan sederhana dengan menggunakan FOR, WHILE-DO dan REPEAT-UNTIL, serta Struktur Pengulangan Bersarang

2. Tujuan Praktikum

Setelah praktikum pada modul 6 ini diharapkan mahasiswa mempunyai kompetensi yaitu memahami Struktur pengulangan FOR, WHILE-DO, REPEAT-UNTIL, serta Struktur Pengulangan Bersarang, dan dapat menggunakannya dalam program.

3. Material Praktikum

Kegiatan pada modul 6 ini memerlukan material berupa penjelasan singkat mengenai Struktur pengulangan dan contoh program dalam Pascal.

4. Kegiatan Praktikum

A. Struktur FOR

Secara garis besar, Struktur FOR terbagi ke dalam dua bentuk, Struktur FOR menaik dan Struktur For menurun. Dalam banyak keadaan, keduanya dapat digunakan pada kasus yang sama walaupun pada kondisi-kondisi tertentu lebih mudah atau lebih tepat menggunakan salah satunya.

Format FOR Menaik: (FOR-TO)

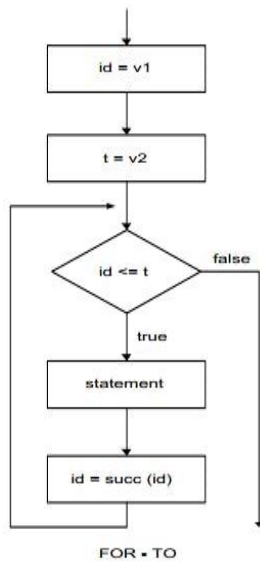
Bentuk umum *pseudocode*:

```
for id ← nilaiawal to nilaiakhir do  
    aksi  
endfor
```

Translasi *pseudocode* pada bahasa pascal:

```
for id:=v1 to v2 do  
    statement;
```

atau dapat digambarkan dengan flowchart seperti di bawah ini.



Format FOR Menurun: (FOR-DOWNTO)

Bentuk umum *pseudocode*:

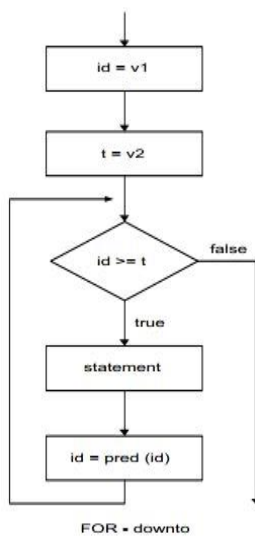
```

for id ← nilaiawal downto nilaiakhir do
    aksi
endfor
  
```

Translasi *pseudocode* pada bahasa pascal:

```

for id:=v1 DOWNto v2 do
    statement;
  
```



Catatan:

1. Variabel id harus bertipe ordinal.
2. Untuk FOR-TO, statement akan dieksekusi secara berulang selama $id \leq t$ dan jika $v1 > v2$ maka otomatis statement tidak dieksekusi sekalipun. Berlaku kebalikannya untuk FOR-DOWNT0.

Contoh pseudocode FOR-TO:

```
algoritma cetak_halo;
{mencetak 'HALO' sebanyak 10 kali}
deklarasi
    n:integer {pencacah pengulangan}
deskripsi
    for n ← 1 to 10 do
        write('HALO')
    endfor
    {kondisi berhenti: n>10}
```

Terjemahan dalam bahasa Pascal.

```
{Program Cetak Halo FOR menaik}
{ 1}program Contoh61;
{ 2}var n:integer;
{ 3}begin
{ 4}    for n:=1 to 10 do
{ 5}        writeln('HALO');
{ 6}    readln;
{ 7}end.
```

Salin program tersebut dan simpan dengan nama **Praktikum6A.pas**. Apakah output dari program tersebut?

Contoh pseudocode FOR-DOWNT0:

```
algoritma cetak_halo;
{mencetak 'HALO' sebanyak 10 kali}
deklarasi
    n:integer {pencacah pengulangan}
deskripsi
```

```

for n ← 10 downto 10 do
    write('HALO')
endfor
{kondisi berhenti: n<1}

```

Terjemahan dalam bahasa Pascal.

```

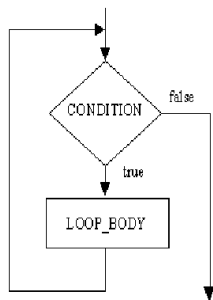
{Program Cetak Halo For Menurun}
{ 1}Program Contoh62;
{ 2}var n:integer;
{ 3}begin
{ 4}    for n:=10 downto 1 do
{ 5}        writeln('HALO');
{ 6}    readln;
{ 7}end.

```

Salin program tersebut dan simpan dengan nama **Praktikum6B.pas**. Apakah output dari program tersebut?

B. Struktur WHILE-DO

Bentuk flowchart:



Bentuk umum *pseudocode*:

```

while <kondisi> do
    aksi
endwhile

```

Translasi *pseudocode* pada bahasa pascal:

```

while kondisi do
    aksi;

```

Ketika instruksi dijalankan, pertama akan dicek pada bagian kondisi. Jika kondisi bernilai **true** maka aksi akan dijalankan. Aksi akan terus dijalankan berulang selama kondisi bernilai true. Jika kondisi sudah bernilai false maka aksi tidak akan dijalankan lagi. Dengan kata lain, `while kondisi=true do aksi`. Perhatikan algoritma cetak halo dan translasinya dalam bahasa pascal menggunakan struktur WHILE-DO berikut ini:

```
algoritma cetak_halo;
{mencetak 'HALO' sebanyak 10 kali}
deklarasi
    n:integer {pencacah pengulangan}
deskripsi
    n ← 1
    while n<=10 do
        write('HALO')
    n ← n+1;
    endwhile
    {kondisi berhenti: n>10}
```

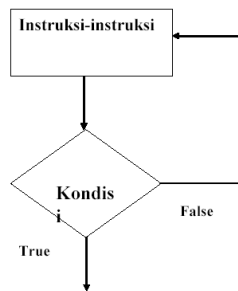
Translasi algoritma Cetak Halo menggunakan struktur WHILE-DO ke Bahasa Pascal.

```
{Program Cetak Halo WHILE-Do}
{ 1}program Contoh6.3;
{ 2}var n:integer;
{ 3}begin
{ 4}    n:=1;
{ 5}    while n<=10 do
{ 6}    begin
{ 7}        writeln('HALO')
{ 8}        n:=n+1;
{ 9}    end;
{10}    readln;
{11}end.
```

Salin program tersebut dan simpan dengan nama **Praktikum6C.pas**. Apakah output dari program tersebut?

C. Struktur REPEAT-UNTIL

Bentuk flowchart:



Bentuk umum algoritma:

```
repeat
aksi
until <kondisi>
```

Translasi dalam Bahasa Pascal:

```
Repeat
aksi;
until kondisi;
```

- 1) Aksi di dalam badan pengulangan terus diulang sampai kondisi bernilai true
- 2) Dengan kata lain, jika kondisi berhenti masih false, pengulangan masih terus dilakukan. Repeat aksi until kondisi=true
- 3) Karena pengulangan harus berhenti, di dalam badan pengulangan harus ada aksi yang mengubah nilai kondisi

Algoritma cetak halo menggunakan struktur REPEAT-UNTIL.

```
algoritma cetak_halo;
{mencetak 'HALO' sebanyak 10 kali}
deklarasi
    n:integer {pencacah pengulangan}
deskripsi
    n ← 1
    repeat
    write('HALO')
    n ← n+1;
    until n > 10
    {kondisi berhenti: n>10}
```

Translasi algoritma cetak halo menggunakan struktur REPEAT-UNTIL dalam bahasa pascal.

```
{Program Cetak Halo REPEAT-UNTIL}
{ 1}Program Contoh64;
{ 2}var n:integer;
{ 3}begin
{ 4}    n:=1;
{ 5}    repeat
{ 6}        writeln('HALO');
{ 7}        n:=n+1;
{ 8}    until n>10;
{ 9}    readln;
{10}end.
```

Salin program tersebut dan simpan dengan nama **Praktikum6D.pas**. Apakah output dari program tersebut? Apa yang dapat anda simpulkan dari **Praktikum6A.pas** hingga **Praktikum6D.pas**?

D. Struktur Pengulangan Bersarang

Di dalam sebuah struktur pengulangan diperbolehkan untuk membuat pengulangan lainnya. Dengan kata lain, pengulangan di dalam pengulangan. Pada operasi FOR dalam FOR, FOR yang paling dalam akan diselesaikan lebih dahulu, baru dieksekusi FOR yang di luar. Sehingga instruksi-instruksi yang didapat pada FOR yang paling dalam akan paling banyak dieksekusi.

Contoh Program Persegi Panjang:

```
{Program Cetak Persegi Panjang '#' Full
{ 1}Program Contoh73;
{ 2}var
{ 3}    i,j:integer;
{ 4}begin
{ 5}    for i:=1 to 5 do
{ 6}    begin
{ 7}        for j:=1 to 9 do
{ 8}            write('#':2);
{ 9}        writeln;
```

```
{10}    end;  
{11}end.
```

1. Salin program tersebut dan simpan dengan nama **Praktikum6E.pas**. Apakah output dari program tersebut?
2. Bagaimana outputnya jika writeln pada baris 9 dihapus?

Coba modifikasi program **Praktikum6E.pas** sedemikian sehingga karakter yang digunakan dan ukuran persegi panjangnya terserah pengguna. Kemudian, output yang dihasilkan outputnya menjadi persegi panjang yang tengahnya kosong seperti berikut ini:

```
# # # # # # #  
#           #  
#           #  
# # # # # # #
```

Petunjuk: Tambahkan variabel dan juga statemen untuk menginput karakter, nilai maksimal untuk i, dan nilai maksimal untuk j. sehingga programnya menjadi Contoh74 di bawah.

5. Penugasan

Buatlah program yang meminta masukan bilangan bulat N, kemudian program akan menampilkan hasil berikut (misal N=5):

Masukkan N: 5

```
*  
**  
***  
****  
*****
```