

MODUL 6 PENGULANGAN BAGIAN 2

1. Deskripsi Singkat

Dalam modul 6 ini akan dipelajari pengulangan FOR, statemen BREAK ,dan CONTINUE dalam struktur pengulangan serta memperdalam konsep struktur pengulangan melalui penugasan.

2. Tujuan Praktikum

Setelah praktikum menggunakan modul 6 ini diharapkan mahasiswa memahami dan mampu mengimplementasikan pengulangan FOR, statemen BREAK dan CONTINUE dalam program Pascal.

3. Material Praktikum

Kegiatan pada modul 6 ini memerlukan material berupa penjelasan singkat serta beberapa contoh program 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)

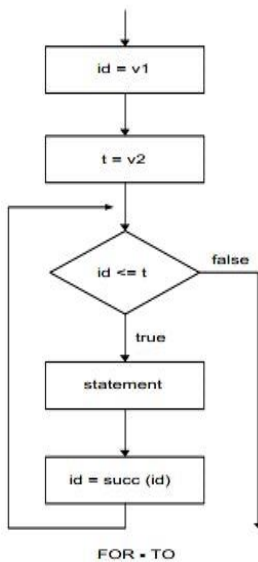
Bentuk umum *pseudocode*:

```
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.



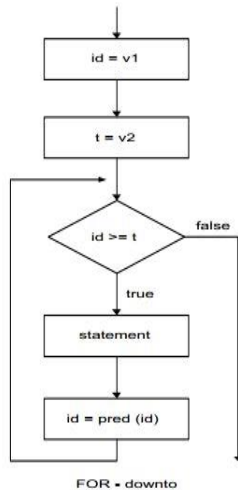
Format FOR Menurun: (FOR-DOWNT0)

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-DOWNTO.

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-DOWNTTO:

```
algoritma cetak_halo;  
{mencetak 'HALO' sebanyak 10 kali}  
deklarasi  
    n:integer {pencacah pengulangan}  
deskripsi  
    for n ← 10 downto 1 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. Statemen BREAK

Sesuai dengan namanya, perintah BREAK digunakan untuk berhenti. Berhenti dari apa? Berhenti dari perulangan yang sedang berlangsung. Apakah itu perulangan FOR, WHILE-DO maupun REPEAT-UNTIL, akan dipaksa berhenti saat ketemu perintah BREAK.

Umumnya perintah **BREAK** disimpan dalam sebuah kondisi IF-ELSE. Selama kondisi belum terpenuhi, perulangan tetap terus berlangsung. Jika kondisi sesuai, maka BREAK.

CONTOH:

```
{Contoh Program Break}

Program contoh_break;

var i:integer;

begin
  for i:=1 to 6 do
  begin
    if (i=4) then break;
    writeln('Coba Break ',i);
  end;
  readln
end.
```

Salin program tersebut dan simpan dengan nama **praktikum6C.pas**. Apakah output dari program tersebut? Apakah sama dengan hasil di bawah ini?

Hasil:

```
Coba Break 1
Coba Break 2
Coba Break 3
```

Simpan ulang **praktikum6C.pas** dengan nama **praktikum6D.pas**. Modifikasi struktur FOR dengan stuktur WHILE-DO. Apakah statement Break dapat bekerja pada praktikum6D.pas? Apakah hasilnya sama dengan praktikum6C.pas?

C. Statemen CONTINUE

Sesuai dengan namanya, perintah CONTINUE digunakan untuk melanjutkan ke perulangan berikutnya tanpa mengeksekusi statemen di bawahnya atau dengan kata lain hanya menghentikan satu kali iterasi saja saat perintah CONTINUE dijalankan.

Sama seperti BREAK, Umumnya perintah **CONTINUE** disimpan dalam sebuah kondisi IF-ELSE.

CONTOH :

```
{Contoh Program Continue}

Program contoh_continue;

var i:integer;

begin
  for i:=1 to 6 do
  begin
    if (i=4) then continue;
    writeln('Coba continue ',i);
  end;
  readln
end.
```

Salin program tersebut dan simpan dengan nama **praktikum6E.pas**. Apakah output dari program tersebut? Apakah sama dengan hasil di bawah ini?

Hasil :

```
Coba continue 1
Coba continue 2
Coba continue 3
Coba continue 5
Coba continue 6
```

Simpan ulang **praktikum6E.pas** dengan nama **praktikum6F.pas**. Modifikasi struktur FOR dengan struktur REPEAT-UNTIL. Apakah statement Break dapat bekerja pada praktikum6F.pas? Apakah hasilnya sama dengan praktikum6E.pas?

D. Penugasan

- 1) Buatlah program yang meminta masukan bilangan bulat N, kemudian program akan menampilkan hasil berikut (misal N=8):

```
Masukkan N: 8
8
7 7
```

```
6 6 6
5 5 5 5
4 4 4 4 4
3 3 3 3 3 3
2 2 2 2 2 2 2
1 1 1 1 1 1 1 1
```

Simpan program dengan nama praktikum6G.pas.

- 2) Buatlah program yang meminta masukan N dan masukan angka-angka sebanyak N . Program akan menghitung jumlah nilai bilangan yang ganjil saja. Misal pengguna ingin memasukkan 3 bilangan. Maka kira-kira outputnya sebagai berikut:

```
Masukkan jumlah bilangan N: 3
Masukkan bilangan ke-1: 12
Masukkan bilangan ke-2: 15
Masukkan bilangan ke-3: 17
Jumlah bilangan ganjil = 32
```

Simpan program dengan nama praktikum6H.pas.

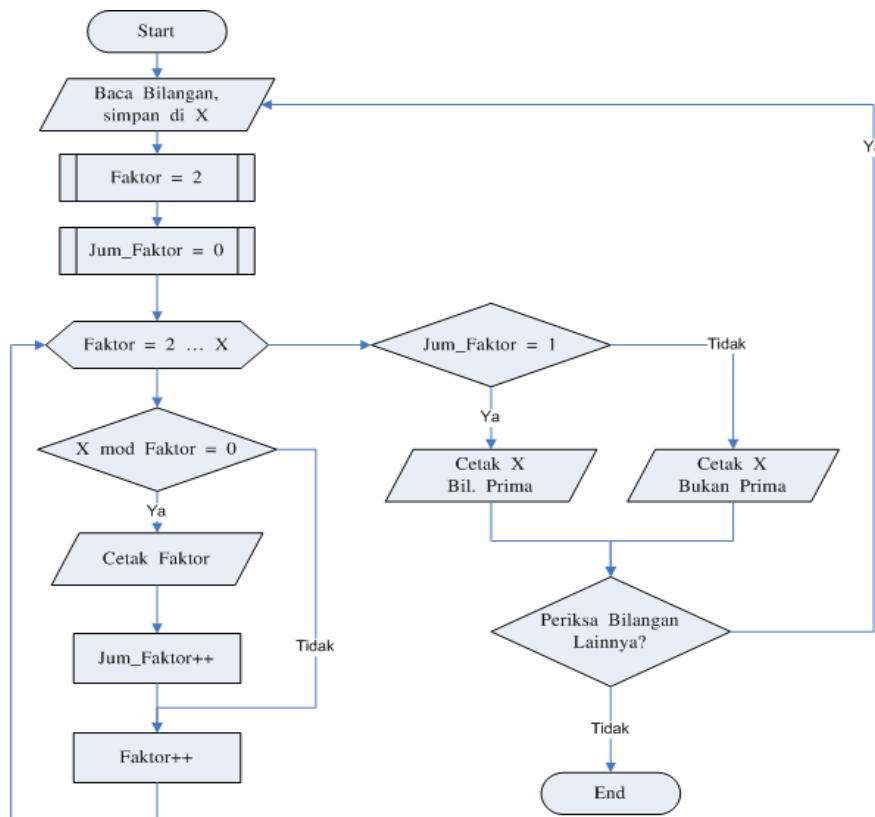
- 3) Dibaca data usia N orang mahasiswa STIS dari masukan. Buatlah program untuk menghitung jumlah mahasiswa yang berusia 17 tahun, 18 tahun, 19 tahun, dan 20 tahun, dan jumlah mahasiswa yang berusia selain itu. Sehingga kira-kira outputnya:

```
Masukkan jumlah mahasiswa : 3
Masukkan usia mahasiswa ke-1: 17
Masukkan usia mahasiswa ke-2: 19
Masukkan usia mahasiswa ke-3: 21
Jumlah mahasiswa berusia 17 tahun = 1
Jumlah mahasiswa berusia 18 tahun = 0
Jumlah mahasiswa berusia 19 tahun = 1
Jumlah mahasiswa berusia 20 tahun = 0
Jumlah mahasiswa berusia selain itu = 1
```

Simpan program dengan nama praktikum6I.pas.

- 4) Buatlah program Pascal untuk mengetahui apakah sebuah bilangan merupakan prima atau bukan, mengacu kepada diagram alir berikut.

Simpan program dengan nama praktikum6J.pas.



Gabungkan file praktikum6G.pas hingga praktikum6J.pas menggunakan **zip** dan beri nama dengan format **praktikum6_nim.zip**. Unggah file **zip** pada Google Classroom sesuai dengan batas waktu yang telah ditetapkan.