



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

STRUKTUR PENGULANGAN

BAGIAN 2

Ibnu Santoso, S.S.T., M.T., Nori Wilantika, S.S.T., M.T.I.



- Struktur **FOR**
 - **FOR** Menaik
 - **FOR** Menurun
- *Nested Loop* / Pengulangan bersarang



FOR



- Struktur FOR digunakan untuk menghasilkan pengulangan sejumlah kali tanpa penggunaan kondisi apapun.
- Struktur ini menyebabkan aksi diulangi sejumlah kali (tertentu)

- **FOR** menaik
- **FOR** menurun

- Bentuk umum:

```
for variabel ← ekspresi-1 to ekspresi-2 do  
    aksi  
endfor
```

- Translasi dalam Bahasa Pascal:

```
for variabel:=a to b do  
    aksi;
```

- variabel harus bertipe sederhana kecuali tipe real
- ekspresi-1 harus lebih kecil atau sama dengan ekspresi-2
- Bila ekspresi-1 lebih besar dari ekspresi-2 maka badan pengulangan tidak dimasuki
- Pada awalnya, variabel dinisialisasi dengan ekspresi-1. Nilai variabel tersebut otomatis **bertambah satu** setiap kali aksi pengulangan dimasuki, sampai akhirnya nilai variabel sama dengan ekspresi-2.
- Jumlah pengulangan yang terjadi adalah $\text{ekspresi2} - \text{ekspresi1} + 1$

```
program cetak_halo;  
var  
    n:integer;  
begin  
    for n:=1 to 10 do  
        writeln('HALO');  
end.
```

Output yang dihasilkan:

HALO

HALO

HALO

HALO

HALO

HALO

HALO

HALO

HALO

HALO

n	OUTPUT
1	HALO
2	HALO
...	
10	HALO

```
program cetak_angka;  
var  
    angka:integer;  
begin  
    for angka:=1 to 20 do  
        writeln(angka);  
end.
```

Output yang dihasilkan:

1

2

3

4

5

...

...

18

19

20

angka

1

2

...

20

OUTPUT

1

2

20


```
program cetak_n_angka;  
var  
    n, angka: integer;  
begin  
    write('masukkan angka: ');  
    readln(n);  
    for angka:=1 to n do  
        writeln(angka);  
end.
```

Output yang dihasilkan:

1		
2		
3		
4		
5		
6	angka	OUTPUT
7	1	1
...	2	2
...	...	
n	n	n

- Bentuk umum:

```
for variabel ← ekspresi-1 downto ekspresi-2 do  
    aksi  
endfor
```

- Translasi dalam Bahasa Pascal:

```
for variabel:=a downto b do  
    aksi;
```

- variabel harus bertipe sederhana kecuali tipe real
- ekspresi-1 harus lebih besar atau sama dengan ekspresi-2
- Bila ekspresi-1 lebih kecil dari ekspresi-2 maka badan pengulangan tidak dimasuki
- Pada awalnya, variabel dinisialisasi dengan ekspresi-1. Nilai variabel tersebut otomatis **berkurang satu** setiap kali aksi pengulangan dimasuki, sampai akhirnya nilai variabel sama dengan ekspresi-1.
- Jumlah pengulangan yang terjadi adalah $\text{ekspresi1} - \text{ekspresi2} + 1$

```
program cetak_n_angka_menurun;
var
    n, angka: integer;
begin
    write('masukkan angka: ');
    readln(n);
    for angka:=n downto 1 do
        writeln(angka);
    end.
```

Output yang dihasilkan:

n		
...		
...		
5		
4		
3	angka	OUTPUT
2	n	n
1
	2	2
	1	1

```
program peluncuran_roket;  
var  
    angka:integer;  
begin  
    for angka:=100 downto 0 do  
        writeln(angka);  
    writeln('Roket Meluncur');  
    readln;  
end.
```

Output yang dihasilkan:

```
100  
99  
98  
...  
...  
3  
2  
1  
0  
Roket Meluncur
```



NESTED LOOP

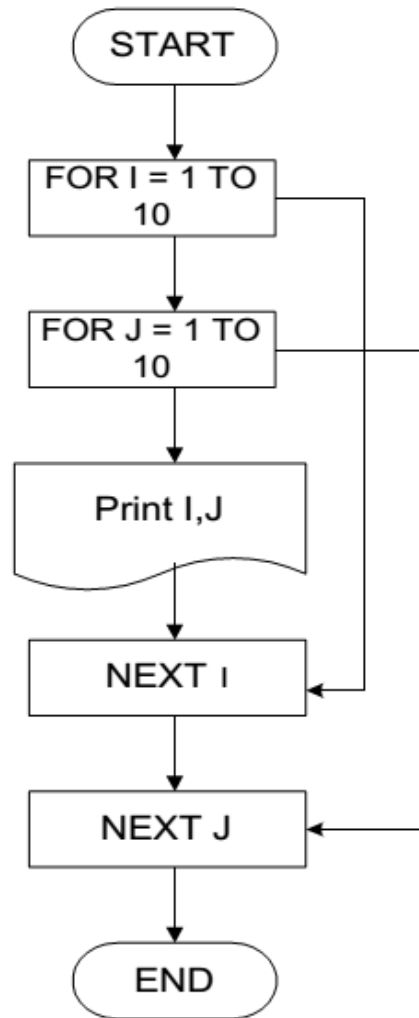


- Di dalam sebuah struktur pengulangan diperbolehkan untuk membuat pengulangan lainnya.
- Dengan kata lain, **pengulangan di dalam pengulangan**.
- Pada operasi FOR dalam FOR, FOR yang paling dalam akan diselesaikan lebih dahulu, baru dieksekusi FOR yang di luar
- sehingga instruksi-instruksi yang didapat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

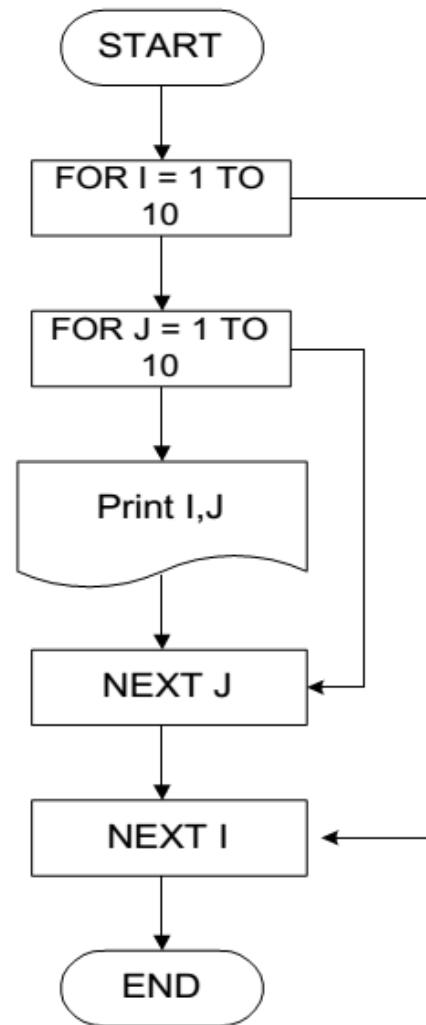
Syarat yang harus diperhatikan untuk operasi FOR pengulangan tersarang :

- Setiap pengulangan tidak boleh menggunakan variabel counter yang sama
- Antara pengulangan-pengulangan tersebut tidak boleh saling berpotongan (*overlapping*)

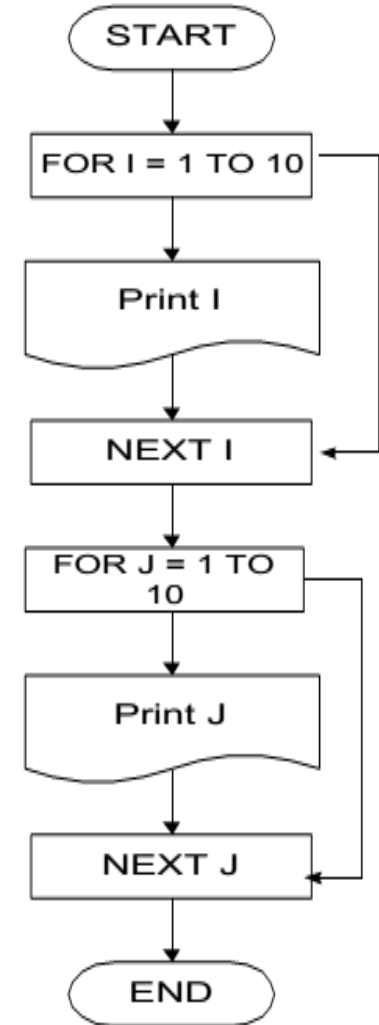
Pengulangan Berpotongan (Salah)



Pengulangan Bersarang (Benar)



Bukan Nested




```
for var1  $\leftarrow$  awal1 to akhir1 do  
    [pernyataan]  
    for var2  $\leftarrow$  awal2 to akhir2 do  
        [pernyataan]  
    endfor  
endfor
```

```
while [kondisi1] do  
    [pernyataan]  
    while [kondisi2] do  
        [pernyataan]  
    endwhile  
endwhile
```

```
repeat
    [pernyataan]
    repeat
        [pernyataan]
    until [kondisi2]
until [kondisi1]
```

```

program peluncuran_roket;
var
    k,p:integer;
begin
    for k:=100 downto 0 do
        begin
            for p:=1 to 1000 do
                writeln(p);
            end;
            writeln('Roket Meluncur');
        readln;
    end.

```

Output yang dihasilkan:

1			
2			
...			
999	k	p	OUTPUT
1000	100	1	1
		2	2
...			
...	
...		1000	1000
1	99	1	1
2		2	2
...	
999		1000	1000
1000			
Roket Meluncur			

- Buatlah algoritma dan program untuk mencetak output sebagai berikut:

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

0 1 2 3 4 5 6 7 8 9

Algoritma cetak_angka

Deskripsi

i, j: integer

Deklarasi

```
for i ← 1 to 5 do
    for j ← 0 to 9 do
        Write(j)
    endfor
endfor
```

```
program cetak_angka;  
var  
    i,j:integer;  
begin  
    for i:=1 to 5 do  
        for j:=0 to 9 do  
            begin  
                write(j, ' ');  
                if j=9 then writeln;  
            end;  
        readln;  
    end.
```

Modifikasi program pada Contoh 7, sehingga menghasilkan output sebagai berikut:

0	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	0
0	1	2	3	4	5	6	7	8	9
9	8	7	6	5	4	3	2	1	0
0	1	2	3	4	5	6	7	8	9



POLITEKNIK STATISTIKA STIS

For Better Official Statistics

TERIMA KASIH

