



# TIB01 - ALGORITMA



# **PENGULANGAN**

**Pertemuan ke-7 dan 8**

# Sub-CPMK

Mahasiswa mampu mengaplikasikan struktur pengulangan pada persoalan tertentu

## Materi

1. Struktur Pengulangan
2. Konsep Pengulangan FOR  
For (inisialisasi\_pencacah; kondisi; penambahan\_pencacah)
  1. Konsep WHILE
  2. Konsep DO-WHILE
  3. Konsep Repeat
  4. Contoh kasus percabangan



# 1. Konsep Looping (FOR)

## PENGULANGAN (*looping*)

Struktur Pengulangan terdiri dari :

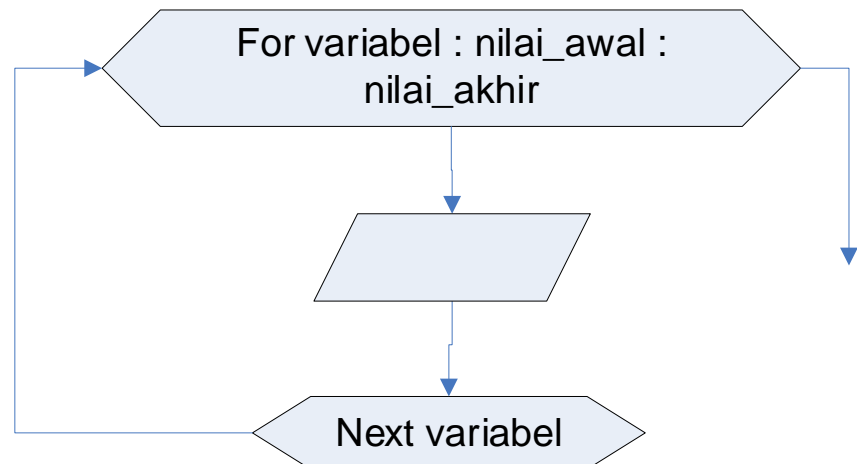
Kondisi pengulangan, apabila ekspresi *boolean* terpenuhi,  
Body pengulangan, yaitu satu atau lebih aksi yang akan diulang

### Bagian Struktur Pengulangan :

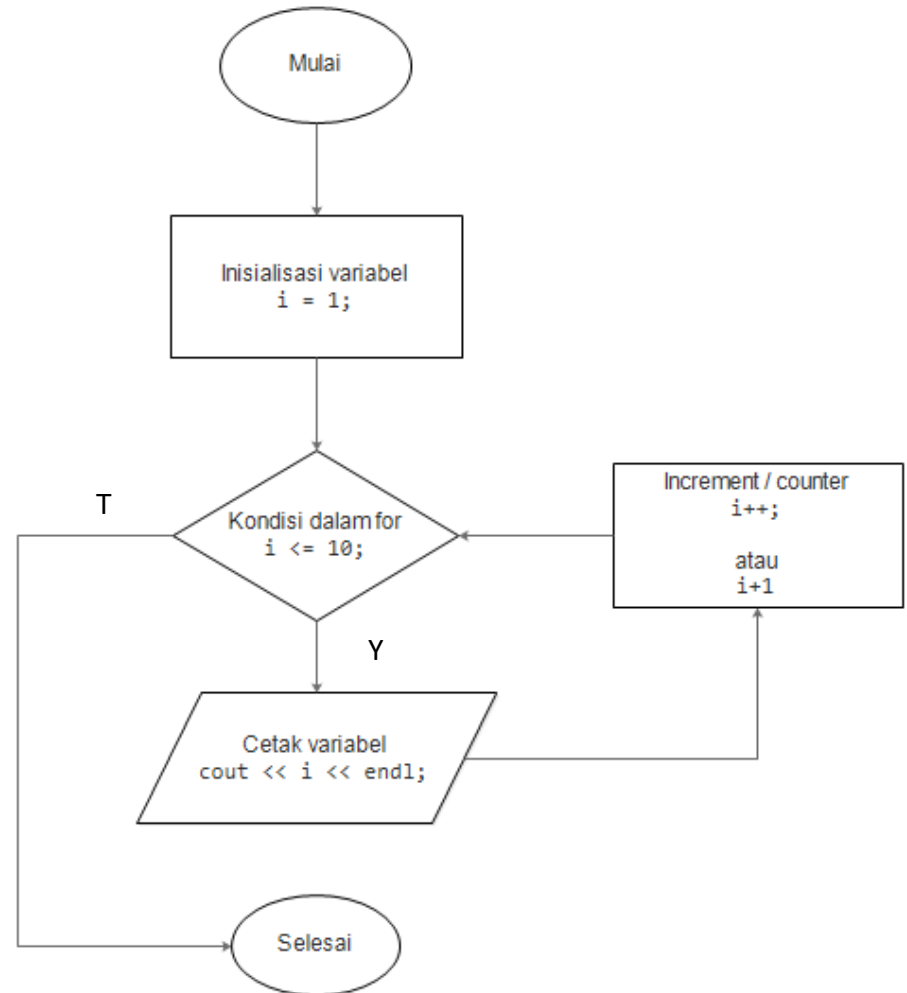
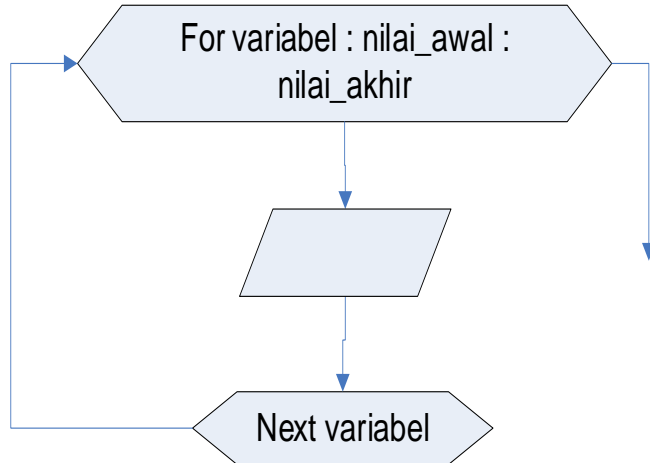
1. **Inisialisasi**, aksi dilakukan sebelum pengulangan dilakukan satu kali
2. **Terminasi**, aksi dilakukan setelah pengulangan selesai dilaksanakan.

- Notasi algoritma perulangan dengan menggunakan konstruksi FOR, berbentuk:

```
For (inisialisasi_pencacah; kondisi; penambahan_pencacah)
{
    aksi
}
```



- Flowchart perulangan dengan menggunakan konstruksi FOR, berbentuk:



## Contoh : Menampilkan angka 1 sampai 5

Contoh algoritma :

Algoritma Menampilkan\_Angka  
{ Menampilkan angka 1 sampai  
5, menggunakan  
pengulangan FOR }

DEKLARASI

i : integer

DESKRIPSI :

Deklarasi

i : integer

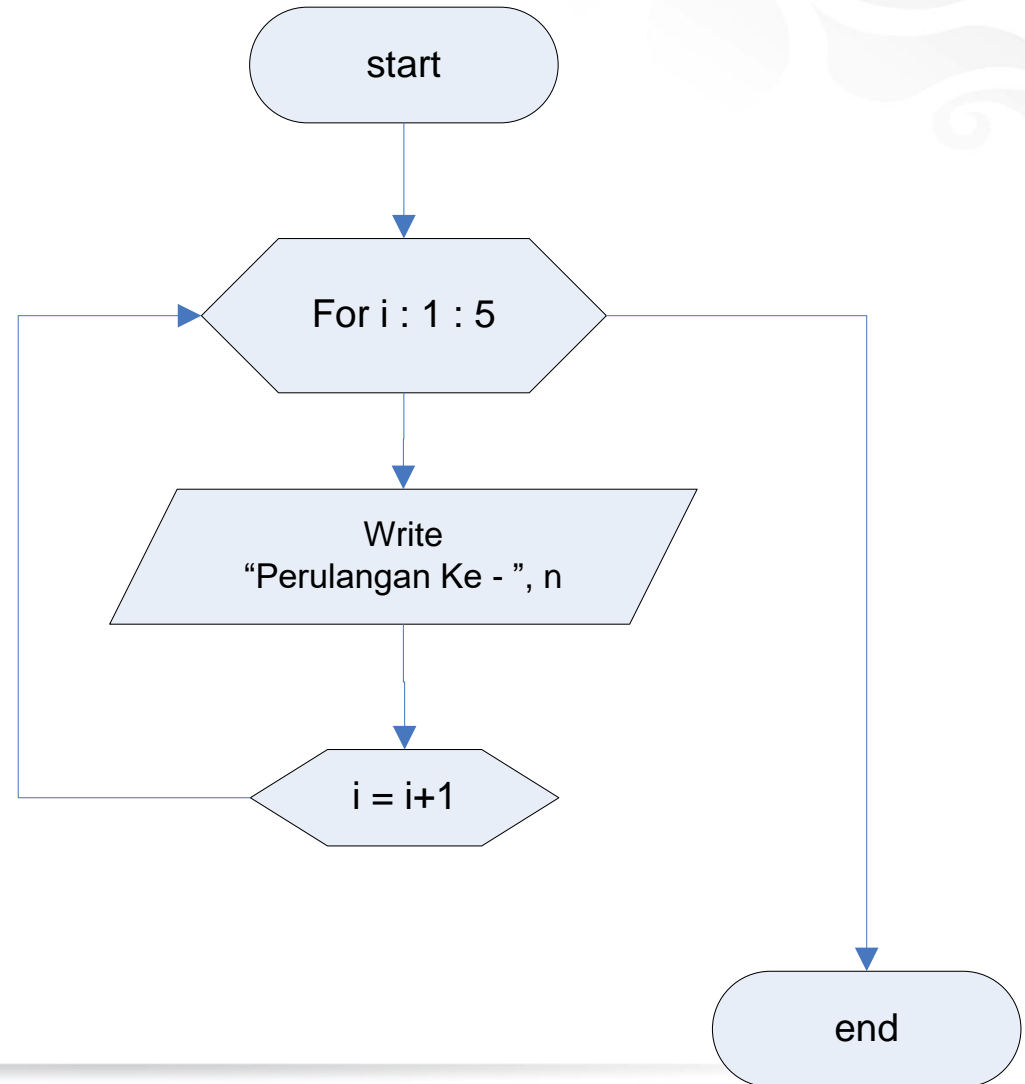
Begin

FOR ( i ← 0 ; i < 5 ; i ++ )

write (i)

END FOR

END



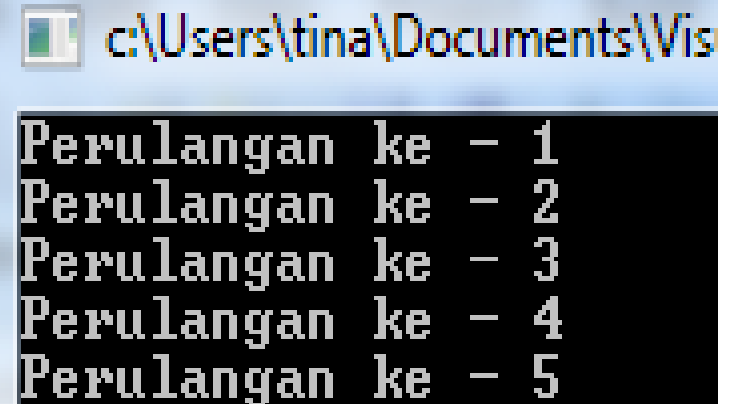


## Contoh : Menampilkan angka 1 sampai 5

### Contoh Program C++ :

```
#include <iostream>
#include <conio.h>
using namespace std;

void main()
{
    for (int i = 1; i<=5; i++)
    {
        cout <<"Perulangan ke - 
        "<<i<<endl;
    }
    _getch();
}
```



```
c:\Users\tina\Documents\Vis
Perulangan ke - 1
Perulangan ke - 2
Perulangan ke - 3
Perulangan ke - 4
Perulangan ke - 5
```



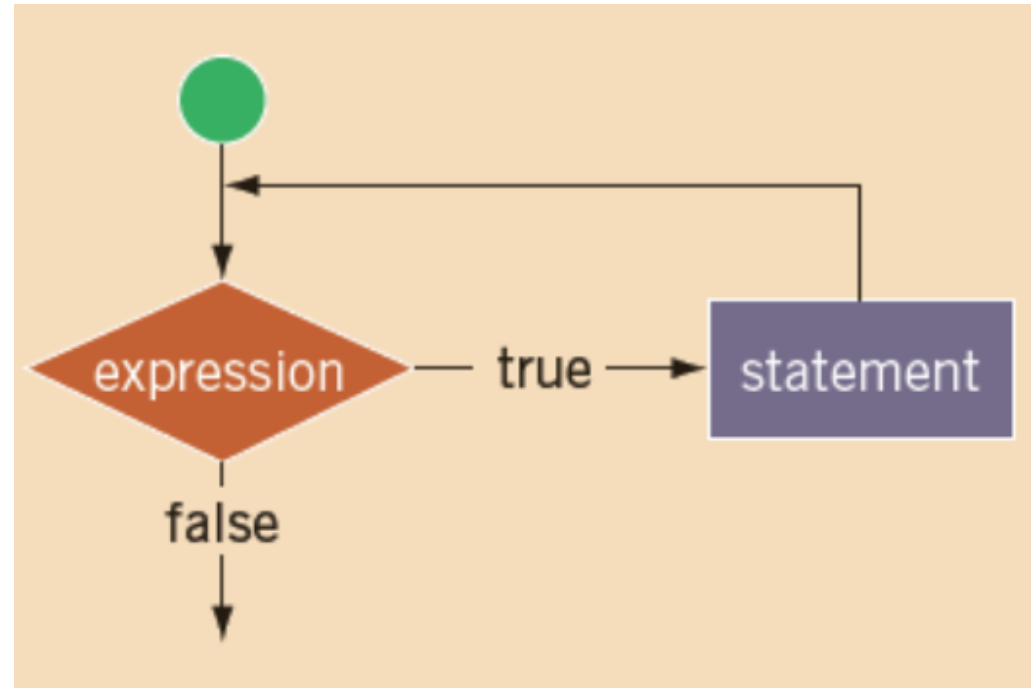
## 2. Konsep Perulangan (While)

## 2.1 Konsep **while**

- Perulangan ini banyak digunakan bila jumlah perulangannya belum diketahui.
- Proses perulangan akan terus berlanjut selama kondisinya bernilai benar (true) dan akan berhenti bila kondisinya bernilai salah.

```
while (syarat)
{ instruksi;
  ...
}
```

- Notasi algoritma untuk perulangan dengan WHILE, berbentuk:  
**while (kondisi)**  
  {  
    **aksi**  
  }



## 2.2 Ekuivalensi **for** dan **while**

**for(initial statement; logical expression; update expression)  
statement**

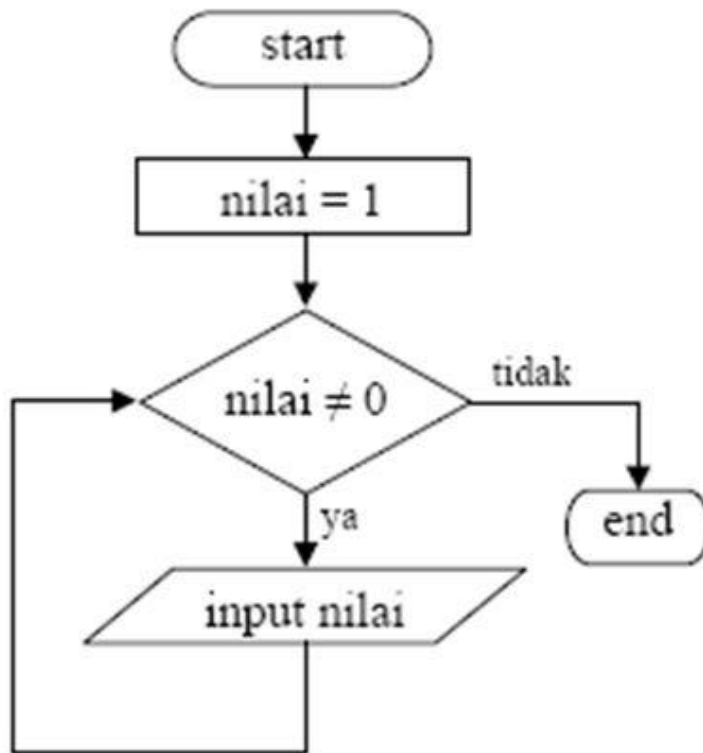
**Initial expression  
while(expression)  
{  
statement  
update expression  
}**

## 2.3 Counter-Controlled **while** Loops

- Digunakan jika kita sudah mengetahui jumlah pengulangan di awal.
- Maka jika **counter** < **n**, aksi yang berada di dalam **while** akan dieksekusi.
- Struktur:

```
counter=0 //inisialisasi variable counter
while (counter<N)
{
    ...
    Counter++;
    ...
}
```

# Pseudocode dan flowchart



## Algoritma Input\_Bilangan

{ Menginput bilangan integer  
secara terus menerus selama yang di input  
bukan 0 }

### DEKLARASI

nilai : integer

### DESKRIPSI :

nilai ← 1

while (nilai ≠ 0) do

read(nilai)

endwhile



### **3. Konsep perulangan (Do-While)**



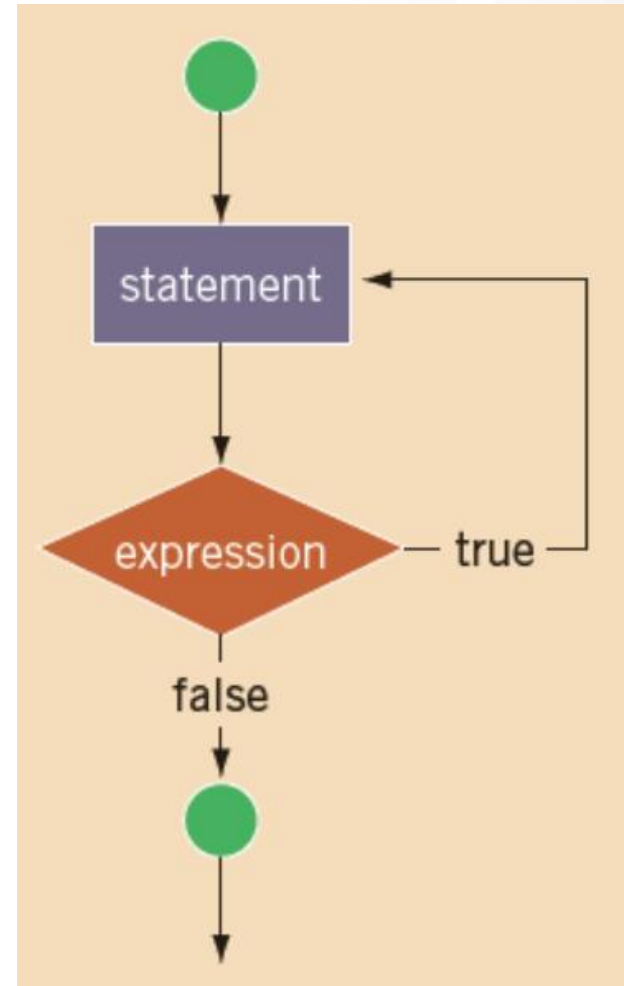
## 3.1 Struktur

- Statement ini akan melakukan pengecekan kondisi di akhir blok statement.
- Dalam hal ini pengulangan juga hanya akan dilakukan jika kondisi yang didefinisikan di dalamnya terpenuhi (bernilai benar).

```
do  
{  
    statement_yang_akan_diulang;  
}  
While
```

## 3.2 Struktur

```
do  
    statement  
while (expression) ;
```



- CONTOH
- Algoritma JUMLAH\_DERET { menjumlahkan deret  $1 + 2 + 3 + \dots + N$  }
- DEKLARASI
  - N, Angka, Jumlah : Integer
- DESKRIPSI
  - read (N) { banyak suku deret }
  - Jumlah  $\leftarrow 0$  { inialisasi jumlah deret }
  - Angka  $\leftarrow 1$  { suku deret }
  - DO {
    - jumlah  $\leftarrow$  jumlah + angka { jumlah deret sekarang }
    - Angka  $\leftarrow$  Angka + 1 { suku deret berikutnya }
  - } WHILE angka > N { angka > N ; kondisi setelah looping berhenti }
  - write (jumlah)



## 4. Repeat - Until

## 4.1 Konsep Repeat...Until

- Digunakan bila jumlah pengulangan belum dapat ditentukan saat program ditulis.
- Pada pernyataan repeat ... Until , kondisi di cek pada akhir loop.
- Bentuk umum:

repeat

{pernyataan yang akan diulang}

until kondisi

### Algoritma Cetak\_Angka

{Mencetak Angka 1 -5 dengan  
menggunakan struktur perulangan  
do .....while}

#### DEKLARASI

k: integer

#### DESKRIPSI

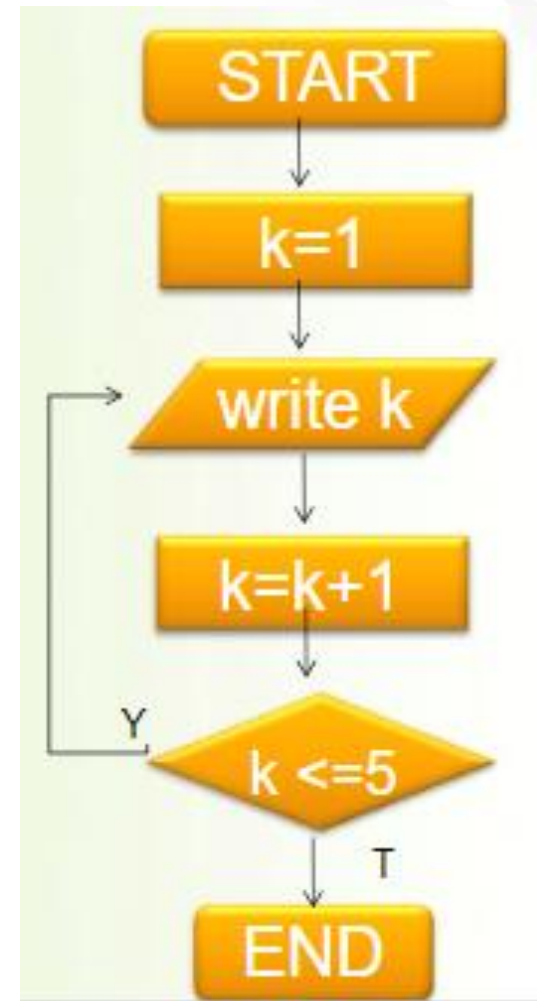
$k \leftarrow 1$

Repeat

    write(k)

$k \leftarrow k+1$

until  $k \leq 5$





# Terima Kasih

U N I V E R S I T A S   B U N D A   M U L I A