



# TIB26 – ALGORITMA PEMROGRAMAN

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



# Seleksi

Pertemuan 19, 20

## Sub-CPMK

- Mahasiswa mampu menggunakan operasi seleksi pada algoritma (C3, A3)

## Materi

1. One Way Selection
2. Two Way Selection
3. Operator Logika

# Perhatian

- Tidak disarankan copy-paste kode program dari presentasi ini, karena ada beberapa symbol yang dianggap sebagai Unicode oleh editor yang anda gunakan, sehingga akan dianggap sebagai symbol yang salah oleh compiler, sebaiknya diketik ulang saja



# 1. One Way Selection

## 1.1 Seleksi

- Seleksi atau nama lainnya adalah percabangan adalah proses seleksi perintah yang akan dikerjakan
- Seleksi dilakukan berdasarkan kondisi.

## 1.2 Kondisi pada pernyataan if

- Kondisi pada pernyataan if merupakan kondisi logika bernilai true atau false
- Kondisi pada pernyataan if dapat memiliki lebih dari satu, relasi antara kondisi yang satu dengan kondisi yang lainnya merupakan relasi dengan operator logika dan ekspresi logika

## 1.3 Kondisi pada pernyataan if (lanj.)

- Pada dasarnya kondisi pada pernyataan if hanya satu saja, yaitu true atau false, karena itu jika lebih dari satu kondisi, maka harus direlasikan dengan operator logika dan disatukan dengan tanda kurung
- Contoh

```
if ( (kondisi1) && (kondisi2) )  
{  
}
```



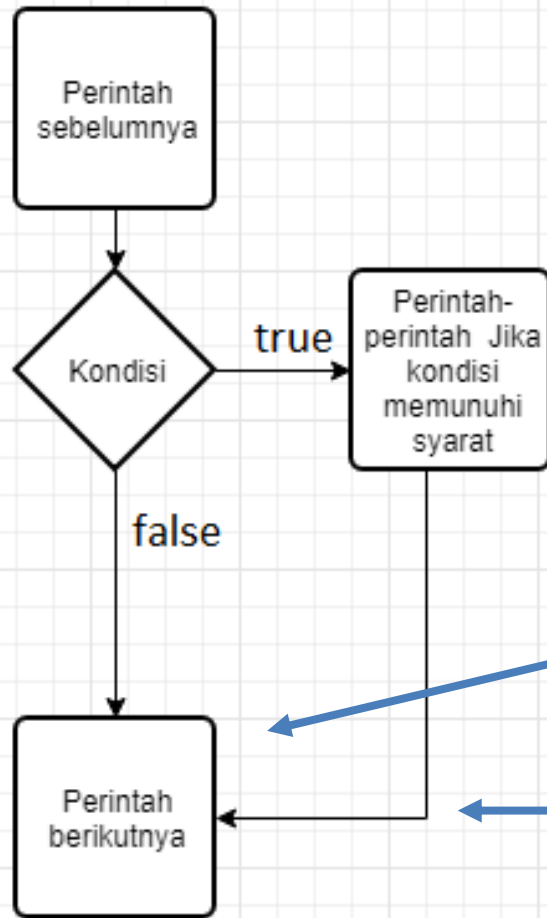
## 1.4 Operator persamaan

- $== \rightarrow$  equal / sama dengan
- $!= \rightarrow$  not equal / tidak sama dengan
- $> \rightarrow$  greater then / lebih besar dari
- $>= \rightarrow$  less then equal / kurang dari sama dengan
- $< \rightarrow$  less then / kurang dari
- $<= \rightarrow$  less then equal / kurang dari sama dengan

## 1.5 One Way Selection

- Secara umum One Way akan menempatkan perintah percabangan pada satu bagian saja, yaitu pada bagian dimana jika kondisi memenuhi persyaratan, maka perintah-perintah tersebut akan dieksekusi

## 1.6 Pola One Way Selection



- Pola dari one way selection adalah seperti pada contoh gambar flowchart di samping
- Jika kondisi memenuhi syarat, lakukan perintah pada seleksi, kemudian lanjutkan ke proses berikutnya
- Jika kondisi tidak memenuhi syarat lanjutkan ke proses berikutnya
- Setelah proses percabangan selesai dilakukan, maka proses akan dilanjutkan ke proses berikutnya

## 1.7 Syntax one way selection

```
if(kondisi)
{
    //perintah yang dikerjakan jika kondisi bernilai true
}
```

## 1.8 Contoh one way selection

Contoh menentukan apakah suatu input adalah bilangan genap;

```
scanf(a);  
if(a%2 == 0) //syarat bilangan genap adalah habis dibagi 2  
{  
    printf("input %d adalah bilangan genap", a);  
}
```

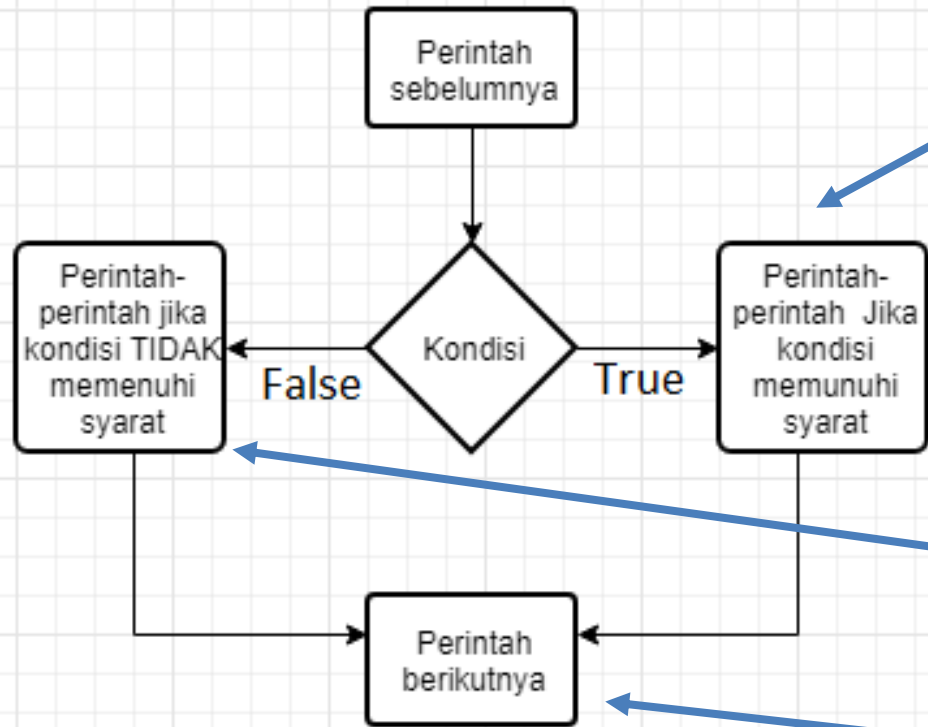


## 2. Two Way Selection

## 2.1 Two Way Selection

- Two way selection akan menempatkan proses-proses pada dua bagian dari seleksi / percabangan
- Satu bagian dari proses-proses ditempatkan pada bagian jika kondisi memenuhi persyaratan
- Bagian yang lain merupakan proses-proses jika kondisi tidak memenuhi syarat

## 2.2 Pola Two Way Selection



- Jika kondisi memenuhi syarat, lakukan perintah pada seleksi, **kemudian lanjutkan ke proses berikutnya**
- Jika kondisi tidak memenuhi syarat lakukan perintah pada pengecualian, **kemudian lanjutkan ke proses berikutnya**
- Semua proses percabangan akan dilanjutkan dengan proses berikutnya yang sama



## 2.3 Syntax two way selection

```
if(kondisi)
{
    //perintah yang dikerjakan jika kondisi bernilai true
} else
{
    //perintah yang dikerjakan jika kondisi bernilai false
}
```

## 2.4 contoh two way selection

- Contoh menentukan apakah suatu input adalah bilangan genap atau ganjil;

```
scanf(a);
```

```
if(a%2 == 0) //syarat bilangan genap adalah habis dibagi 2
```

```
{
```

```
    printf("input %d adalah bilangan genap", a);
```

```
} else
```

```
{
```

```
    printf("input %d adalah bilangan ganjil", a);
```

```
}
```



### 3. Operator Logika

## 3.1 Operator Logika

- `&&` → relasi pernyataan logika AND
- `||` → relasi pernyataan logika OR
- `!` → NOT

## 3.2 Logika Relasi AND

- Menggunakan operator &&
- Untuk merelasikan kedua kondisi yang mana kedua-keduanya harus memenuhi syarat agar melakukan blok perintah
- Penulisan:

```
if ( (kondisi1) && (kondisi2) )  
{  
}
```

## 3.3 Logika Relasi OR

- Menggunakan operator `||`
- Untuk merelasikan kedua kondisi yang mana cukup salah satu kondisi saja yang memenuhi syarat agar melakukan blok perintah
- Penulisan:

```
if ( (kondisi1) || (kondisi2) )  
{  
  
}
```

## 3.4 Logika Relasi NOT

- Menggunakan operator !
- Untuk Membalikkan kondisi yang mana jika kondisi bernilai false, maka akan dibalik menjadi true, ataupun jika true akan bernilai false.
- Dalam hal ini berarti perintah di dalam blok seleksi justru akan dikerjakan bila kondisi bernilai false
- Penulisan:

```
if !(kondisi1)
{
}
```

## 3.4 Logika Relasi NOT (lanj.)

- Kondisi ini sama seperti percabangan one way selection pada blok perintah “**else**”
- Penulisan:

```
if (kondisi1)
{ //tidak mengerjakan apapun }
else
{
    //perintah yang dikerjakan jika
    //kondisi tidak memenuhi syarat
}
```



# Ringkasan

- Kondisi pada pernyataan if merupakan kondisi logika bernilai true atau false
- Kondisi pada pernyataan if dapat memiliki lebih dari satu, relasi antara kondisi yang satu dengan kondisi yang lainnya merupakan relasi dengan operator logika dan ekspresi logika
- Secara umum One Way akan menempatkan perintah percabangan pada satu bagian saja, yaitu pada bagian dimana jika kondisi memenuhi persyaratan, maka perintah-perintah tersebut akan dieksekusi
- Two way selection akan menempatkan proses-proses pada dua bagian dari seleksi / percabangan



*Terimakasih*

*TUHAN Memberkati Anda*

Teady Matius Surya Mulyana (tmulyana@bundamulia.ac.id)