

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 1
“TIPE DATA & VARIABEL”



DISUSUN OLEH:
MUHAMMAD ZAKY MUBAROK
103112400073
S1 IF-12-01
DOSEN:
Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024/2025

DASAR TEORI

Dasar Teori IF-THEN dalam Pemrograman

****Paradigma Percabangan**:**

Dalam pemrograman, eksekusi kode secara sekuensial adalah dasar yang umum, di mana setiap baris kode dieksekusi satu per satu dari awal hingga akhir. Namun, untuk situasi di mana eksekusi kode harus bergantung pada kondisi tertentu, kita menggunakan struktur kontrol percabangan seperti `if-then`.

****Karakteristik IF-THEN**:**

1. ****Kondisi (Condition)**:**

- Kondisi adalah syarat atau ketentuan yang harus terpenuhi agar aksi tertentu dilakukan. Dalam konteks ini, kondisi harus berupa ekspresi boolean yang mengevaluasi menjadi `true` atau `false`.

- Contoh kondisi dalam pemrograman:

```
``go
if a > b {
    // aksi yang dilakukan jika a lebih besar dari b
}
``
```

- Pada contoh di atas, `a > b` adalah kondisi yang dievaluasi. Jika `a` benar-benar lebih besar dari `b`, maka aksi dalam blok `if` akan dieksekusi.

2. ****Aksi (Action)**:**

- Aksi adalah instruksi atau kumpulan instruksi yang dijalankan apabila kondisi yang ditentukan bernilai `true`. Aksi ini hanya akan dijalankan jika kondisi terpenuhi.

- Contoh aksi dalam pemrograman:

```
``go
if a > b {
    fmt.Println("a lebih besar dari b")
}
``
```

- Pada contoh ini, aksi yang dilakukan adalah mencetak "a lebih besar dari b" ke output standar, tetapi hanya jika kondisi `a > b` terpenuhi.

****Struktur Dasar IF-THEN**:**

Struktur dasar `if-then` dalam berbagai bahasa pemrograman secara umum adalah:

```
```go
if kondisi {
 // aksi yang dilakukan jika kondisi bernilai true
}
```
```

- ****`if`****: Kata kunci yang digunakan untuk memulai pernyataan percabangan.
- ****`kondisi`****: Ekspresi yang menghasilkan nilai boolean (`true` atau `false`).
- ****Blok Aksi****: Kumpulan instruksi yang akan dieksekusi jika kondisi bernilai `true`.

Contoh Penerapan

```
```go
package main

import "fmt"

func main() {
 var a int
 fmt.Scan(&a)
 if a > 10 {
 fmt.Println("Nilai a lebih besar dari 10")
 } else {
 fmt.Println("Nilai a kurang atau sama dengan 10")
 }
}
```
```

Dalam contoh ini, program meminta input dari pengguna dan memeriksa apakah nilai `a` lebih besar dari 10. Jika kondisi tersebut terpenuhi, program mencetak "Nilai a lebih besar dari 10". Jika tidak, program mencetak "Nilai a kurang atau sama dengan 10".

CONTOH SOAL

1. Latihan1

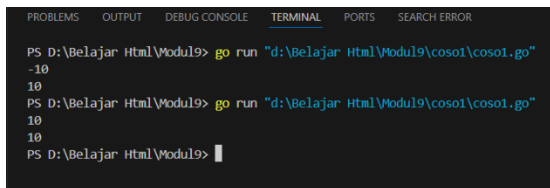
Source Code:

```
package main

import "fmt"

func main() {
    var bilangan int
    fmt.Scan(&bilangan)
    if bilangan < 0 {
        bilangan = -bilangan
    }
    fmt.Print(bilangan)
}
```

Output:



```
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso1\coso1.go"
-10
10
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso1\coso1.go"
10
10
PS D:\Belajar Html\Modul9> 
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var bilangan int:** Mendeklarasikan variabel bilangan bertipe integer untuk menyimpan nilai yang akan diinput oleh pengguna.

2. Input Pengguna:

- **fmt.Scan(&bilangan):** Meminta pengguna memasukkan sebuah bilangan bulat dan membacanya dari input standar.

3. Logika if :

- **if bilangan < 0:** Memeriksa apakah bilangan yang diinput bernilai negatif.
 - Jika bilangan bernilai negatif ($\text{bilangan} < 0$), maka nilai variabel bilangan diubah menjadi positif dengan menggunakan operator negatif ($-\text{bilangan}$). Ini adalah cara untuk mendapatkan nilai absolut dari bilangan.
 - Jika bilangan tidak bernilai negatif, maka tidak ada perubahan yang dilakukan pada bilangan.

4. Output:

- **fmt.Print(bilangan):** Mencetak nilai bilangan ke output standar. Jika bilangan awalnya negatif, program akan mencetak nilai absolutnya

(positif). Jika bilangan awalnya positif, program akan mencetak nilai itu sendiri.

Fungsi Program:

Program ini digunakan untuk membaca sebuah bilangan bulat positif atau negatif dari input pengguna dan kemudian mencetak nilai absolutnya (nilai positifnya). Jika pengguna memasukkan bilangan negatif, program akan mengubahnya menjadi positif sebelum mencetaknya.

Contoh:

- Jika memasukkan -5, program akan mencetak 5.
- Jika memasukkan 10, program akan mencetak 10.

Program ini membantu untuk memastikan bahwa output selalu berupa bilangan positif.

CONTOH SOAL

2. Latihan2

Source Code:

```
package main

import "fmt"

func main() {
    var a int
    var teks string
    fmt.Scan(&a)
    teks = "negatif"
    if a >= 0 {
        teks = "positif"
    }
    fmt.Println(teks)
}
```

Output:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso2\coso2.go"
-4
negatif
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso2\coso2.go"
2
positif
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso2\coso2.go"
0
positif
PS D:\Belajar Html\Modul9> |
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var a int:** Mendeklarasikan variabel a bertipe integer untuk menyimpan nilai bilangan yang diinput oleh pengguna.
- **var teks string:** Mendeklarasikan variabel teks bertipe string untuk menyimpan hasil keluaran yang akan dicetak.

2. Input Pengguna:

- **fmt.Scan(&a):** Meminta pengguna memasukkan sebuah bilangan bulat dan membacanya dari input standar, lalu menyimpannya dalam variabel a.

3. Inisialisasi Variabel teks:

- `teks = "negatif"`: Menginisialisasi variabel teks dengan nilai "negatif".

4. Logika if else:

- `if a >= 0`: Memeriksa apakah bilangan a yang diinput lebih besar atau sama dengan 0.
 - Jika kondisi ini benar (bilangan a positif atau nol), maka nilai variabel teks diubah menjadi "positif".
 - Jika kondisi ini salah (bilangan a negatif), nilai variabel teks tetap "negatif".

5. Output:

- `fmt.Println(teks)`: Mencetak nilai teks ke output standar. Jika bilangan a positif atau nol, program akan mencetak "positif". Jika bilangan a negatif, program akan mencetak "negatif".

Fungsi Program:

Program ini digunakan untuk menentukan apakah sebuah bilangan bulat yang diinput oleh pengguna adalah positif atau negatif. Program akan mencetak "positif" jika bilangan lebih besar atau sama dengan 0, dan mencetak "negatif" jika bilangan kurang dari 0.

Contoh:

- Jika memasukkan 5, program akan mencetak "positif".
- Jika memasukkan -3, program akan mencetak "negatif".

CONTOH SOAL

3. Latihan3

Source Code:

```
package main

import "fmt"

func main() {
    var a int
    var hasil bool
    fmt.Scan(&a)
    if a < 0 && a%2 == 0 {
        hasil = true
    }
    fmt.Print(hasil)
}
```

Output:



```
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso3\coso3.go"
-4
true
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso3\coso3.go"
-3
false
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso3\coso3.go"
2
false
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\coso3\coso3.go"
4
false
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var a int:** Mendeklarasikan variabel a bertipe integer untuk menyimpan nilai bilangan yang diinput oleh pengguna.
- **var hasil bool:** Mendeklarasikan variabel hasil bertipe boolean untuk menyimpan hasil pemeriksaan apakah bilangan memenuhi kondisi tertentu.

2. Input Pengguna:

- **fmt.Scan(&a):** Meminta pengguna memasukkan sebuah bilangan bulat dan membacanya dari input standar, lalu menyimpannya dalam variabel a.

3. Logika if:

- **if a < 0 && a%2 == 0:** Memeriksa apakah bilangan a memenuhi dua kondisi berikut:
 - **a < 0:** Apakah a adalah bilangan negatif.
 - **a%2 == 0:** Apakah a adalah bilangan genap (kelipatan 2).
- **Jika kedua kondisi tersebut benar, maka hasil diubah menjadi true. Jika salah satu atau kedua kondisi tidak terpenuhi, hasil tetap dengan nilai default false.**

4. Output:

- **fmt.Print(hasil):** Mencetak nilai hasil ke output standar. Jika a adalah bilangan negatif dan genap, hasil akan bernilai true. Jika tidak, hasil akan bernilai false.

Fungsi Program:

Program ini digunakan untuk menentukan apakah sebuah bilangan bulat yang diinput oleh pengguna adalah bilangan negatif dan genap. Program akan mencetak true jika bilangan memenuhi kedua kondisi tersebut (negatif dan genap), dan mencetak false jika tidak memenuhi salah satu atau kedua kondisi.

Contoh:

- **Jika memasukkan -4, program akan mencetak true (karena -4 adalah bilangan negatif dan genap).**
- **Jika memasukkan -3, program akan mencetak false (karena -3 adalah bilangan negatif tetapi tidak genap).**
- **Jika memasukkan 5, program akan mencetak false (karena 5 adalah bilangan positif).**

SOAL LATIHAN

Statement perulangan

1.

Source Code:

```
package main

import "fmt"

func main() {
    var n int
    fmt.Scan(&n)

    k := 2
    m := n / k

    if n % k != 0 {
        m++
    }

    fmt.Print(m)
}
```

Output:



```
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol1\latsol1.go"
10
5
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol1\latsol1.go"
25
13
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol1\latsol1.go"
1
1
PS D:\Belajar Html\Modul9> █
```

Deskripsikan Program :

1. Deklarasi Variabel:

- **var n int:** Mendeklarasikan variabel n bertipe integer untuk menyimpan jumlah orang yang akan melakukan touring.

2. Input Pengguna:

- **fmt.Scan(&n):** Membaca input dari pengguna dan menyimpannya dalam variabel n.

3. Logika Penghitungan:

- **k := 2:** Menyimpan kapasitas satu motor (2 orang) dalam variabel k.
- **m := n / k:** Menghitung jumlah motor yang dibutuhkan dengan membagi jumlah orang (n) dengan kapasitas motor (k).
- **if n % k != 0:** Memeriksa apakah ada sisa orang yang tidak cukup dalam satu motor.
 - **Jika ada sisa orang (n % k != 0), tambahkan 1 motor ekstra (m++).**

4. Output:

- **fmt.Print(m):** Mencetak jumlah motor yang diperlukan ke output.

Program ini memastikan bahwa setiap motor diisi maksimal 2 orang, dan menambahkan motor jika ada sisa orang yang tidak cukup dalam satu motor.

SOAL LATIHAN

2. Latihan2

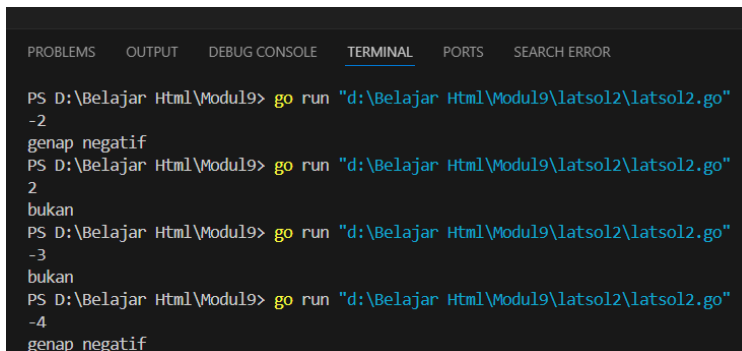
Source Code:

```
package main

import "fmt"

func main() {
    var a int
    var hasil string
    fmt.Scan(&a)
    hasil = "bukan"
    if a < 0 && a%2 == 0 {
        hasil = "genap negatif"
    }
    fmt.Print(hasil)
}
```

Output:



```
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol2\latsol2.go"
-2
genap negatif
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol2\latsol2.go"
2
bukan
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol2\latsol2.go"
-3
bukan
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol2\latsol2.go"
-4
genap negatif
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var a int:** Mendeklarasikan variabel a bertipe integer untuk menyimpan nilai bilangan yang akan diinput oleh pengguna.
- **var hasil string:** Mendeklarasikan variabel hasil bertipe string untuk menyimpan hasil keluaran yang akan dicetak.

2. Input Pengguna:

- **fmt.Scan(&a):** Membaca input dari pengguna dan menyimpannya dalam variabel a.

3. Inisialisasi Variabel hasil:

- **hasil = "bukan":** Menginisialisasi variabel hasil dengan nilai "bukan".

4. Logika if:

- **if $a < 0$ && $a \% 2 == 0$:** Memeriksa apakah bilangan a memenuhi dua kondisi berikut:
 - **$a < 0$:** Apakah a adalah bilangan negatif.
 - **$a \% 2 == 0$:** Apakah a adalah bilangan genap (kelipatan 2).
- **Jika kedua kondisi tersebut benar, maka hasil diubah menjadi "genap negatif".** Jika salah satu atau kedua kondisi tidak terpenuhi, hasil tetap dengan nilai "bukan".

5. Output:

- **fmt.Print(hasil):** Mencetak nilai hasil ke output standar. Jika a adalah bilangan negatif dan genap, hasil akan bernilai "genap negatif". Jika tidak, hasil akan bernilai "bukan".

Fungsi Program:

Program ini digunakan untuk menentukan apakah sebuah bilangan bulat yang diinput oleh pengguna adalah bilangan genap negatif. Program akan mencetak "genap negatif" jika bilangan memenuhi kedua kondisi tersebut (negatif dan genap), dan mencetak "bukan" jika tidak memenuhi salah satu atau kedua kondisi.

Contoh:

- **Jika memasukkan -4, program akan mencetak "genap negatif"** (karena -4 adalah bilangan negatif dan genap).
- **Jika memasukkan -3, program akan mencetak "bukan"** (karena -3 adalah bilangan negatif tetapi tidak genap).
- **Jika memasukkan 5, program akan mencetak "bukan"** (karena 5 adalah bilangan positif).

SOAL LATIHAN

Statement perulangan

3. Latihan3

Source Code:

```
package main

import "fmt"

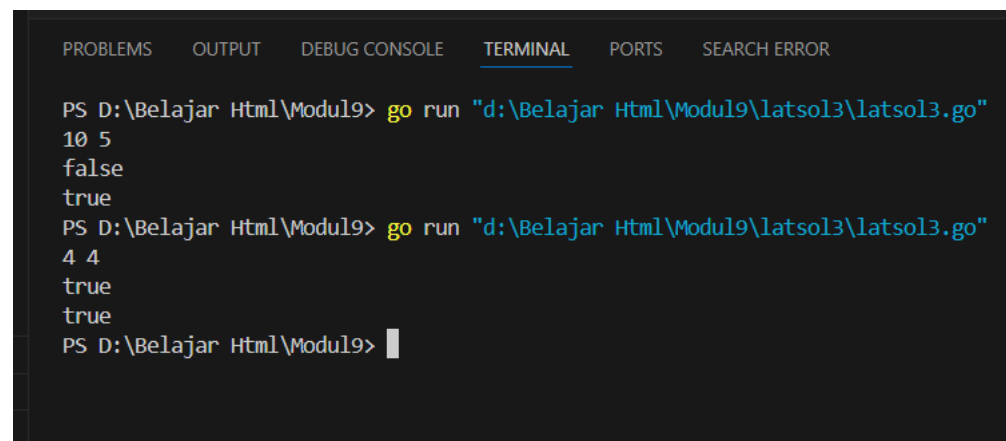
func main() {
    var x, y int
    fmt.Scan(&x, &y)

    xFaktorY := false
    if y % x == 0 {
        xFaktorY = true
    }

    yFaktorX := false
    if x % y == 0 {
        yFaktorX = true
    }

    fmt.Println(xFaktorY)
    fmt.Println(yFaktorX)
}
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  SEARCH ERROR

PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol3\latsol3.go"
10 5
false
true
PS D:\Belajar Html\Modul9> go run "d:\Belajar Html\Modul9\latsol3\latsol3.go"
4 4
true
true
PS D:\Belajar Html\Modul9> |
```

Deskripsi Program:

1. Deklarasi Variabel:

- **var x, y int:** Mendeklarasikan variabel x dan y bertipe integer untuk menyimpan nilai bilangan yang akan diinput.

2. Input Pengguna:

- **fmt.Print("Masukkan dua bilangan bulat positif: "):** Menampilkan pesan untuk meminta input.
- **fmt.Scan(&x, &y):** Membaca dua bilangan bulat positif yang diinput dan menyimpannya dalam variabel x dan y.

3. Logika Pemeriksaan Faktor:

- **xFaktorY := false:** Inisialisasi variabel xFaktorY sebagai false.
- **if y % x == 0:** Memeriksa apakah x adalah faktor dari y (dengan memeriksa apakah sisa hasil bagi y dengan x adalah 0). Jika benar, ubah xFaktorY menjadi true.
- **yFaktorX := false:** Inisialisasi variabel yFaktorX sebagai false.
- **if x % y == 0:** Memeriksa apakah y adalah faktor dari x (dengan memeriksa apakah sisa hasil bagi x dengan y adalah 0). Jika benar, ubah yFaktorX menjadi true.

4. Output:

- **fmt.Println(xFaktorY):** Mencetak hasil pemeriksaan apakah x adalah faktor dari y (true atau false).
- **fmt.Println(yFaktorX):** Mencetak hasil pemeriksaan apakah y adalah faktor dari x (true atau false).

Program ini akan mencetak true atau false berdasarkan apakah bilangan pertama adalah faktor dari bilangan kedua, dan sebaliknya

DAFTAR PUSTAKA

<https://blog.ruangdeveloper.com/golang-if-expression/>