

LAPORAN PRAKTIKUM ALGORITMA

DAN PEMROGRAMAN 1

MODUL 9

IF-THEN



**Telkom
University
PURWOKERTO**

Disusun oleh:

MUHAMMAD FIRDAUS ARDIANSYAH

109082500126

S1IF-13-07

Asisten Praktikum

Adithana dharma putra

Apri pandu wicaksono

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {
    var a int

    fmt.Scan(&a)

    if a<0 {
        a *= -1
    }

    fmt.Println(a)
}
```

Screenshot program

```

package main

import "fmt"

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

    if a<0 {
        a *= -1
    }
    fmt.Println(a)
}

```

Terminal output:

```

PS D:\coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided1\main.go"
10
10
PS D:\coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided1\main.go"
-3
3
PS D:\coding\Latihan GO 1>

```

Deskripsi program

Program ini bertujuan untuk menghitung nilai absolut dari sebuah bilangan integer yang diinput oleh pengguna. Fungsi utamanya, func main(), dimulai dengan mendeklarasikan sebuah variabel integer, a, melalui var a int. Program kemudian menunggu input dari kita melalui fungsi fmt.Scan(&a). Nilai integer yang dimasukkan oleh kita ini akan disimpan di dalam variabel a. Setelah input diterima, program akan mengecek kondisi menggunakan pernyataan if a < 0.

Jika kondisi a < 0 (nilai yang dimasukkan negatif) terpenuhi, blok kode di dalamnya akan dieksekusi, yaitu a = a * -1. Operasi ini mengubah nilai negatif a menjadi nilai positifnya (nilai absolut). Sebagai contoh, jika a = -3, maka operasi ini akan menghasilkan a = (-3) x (-1) = 3. Jika nilai a adalah nol atau positif, kondisi if akan bernilai false, dan nilai a akan tetap dipertahankan. Akhirnya, fungsi fmt.Println(a) akan mencetak nilai akhir dari variabel a ke konsol. Tampilan eksekusi di terminal (seperti -3 menjadi 3) membuktikan bahwa program ini secara fungsional bekerja sebagai kalkulator nilai absolut sederhana.

2. Guided 2

Source Code

```

package main

import "fmt"

```

```

func main() {

    var a int

    fmt.Scan(&a)

    if a>0 {

        fmt.Println("positif")

    }

    if a<0 {

        fmt.Println("bukan positif")

    }

}

```

Screenshot program

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under "LATHAN GO 1" containing files like BERLATIH, Coba, latihan9, latihan, latihan2, latihan3, main.go, Modul1, Modul2, Modul3, Modul4, Modul5, modul9, guided1, guided2, and soal1 through soalA.
- Code Editor:** Displays the content of main.go:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var a int
7     fmt.Scan(&a)
8     if a>0 {
9         fmt.Println("positif")
10    }
11    if a<0 {
12        fmt.Println("bukan positif")
13    }
14 }

```
- Terminal:** Shows the command line output for running the program:

```

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided2\main.go"
10
positif
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided2\main.go"
-3
bukan positif
PS D:\Coding\Latihan GO 1>

```
- Status Bar:** Shows the current file is main.go, and the terminal encoding is UTF-8.

Deskripsi program

Program Go ini, dirancang untuk menguji dan mengklasifikasikan sebuah bilangan integer yang diinput kita apakah ia "positif" atau "bukan positif". Fungsi utamanya, func main(), memulai dengan mendeklarasikan variabel integer a (var a int). Program kemudian meminta input numerik dari kita menggunakan fmt.Scan(&a).

Nilai input ini selanjutnya akan dievaluasi menggunakan dua blok kondisional if yang terpisah (bukan if-else atau if-else if), yang berarti kedua kondisi akan diperiksa secara independen.

Blok kondisional pertama adalah if $a \geq 0$. Jika nilai yang dimasukkan oleh pengguna lebih besar dari atau sama dengan nol (yaitu nol atau bilangan positif), program akan mencetak string "positif" ke konsol. Blok kondisional kedua adalah if $a < 0$. Jika nilai yang dimasukkan kurang dari nol (negatif), program akan mencetak string "bukan positif" ke konsol. Hasil eksekusi di terminal menunjukkan fungsionalitas ini, ketika input adalah 10 (yang ≥ 0), outputnya adalah "positif"; dan ketika input adalah -3 (yang < 0), outputnya adalah "bukan positif". Perlu dicatat bahwa angka nol (0) akan memicu output "positif" karena memenuhi kondisi $a \geq 0$.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {
    var a int
    var b bool

    fmt.Scan(&a)

    if a<0 && a%2==0 {
        b = true
    }

    if a>0 && a%2!=0 {
        b = false
    }
}
```

```

        fmt.Println(b)

    }

```

Screenshot program

```

package main

import "fmt"

func main() {
    var a int
    var b bool

    fmt.Scan(&a)

    if a<0 && a%2==0 {
        b = true
    }
    if a>0 && a%2!=0 {
        b = false
    }
    fmt.Println(b)
}

```

Output from Terminal:

```

PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided3\main.go"
10
false
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided3\main.go"
-3
false
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided3\main.go"
-4
false
PS D:\Coding\Latihan GO 1> go run "d:\Coding\Latihan GO 1\modul9\guided3\main.go"
4
true
PS D:\Coding\Latihan GO 1>

```

Deskripsi program

Program Go ini dirancang untuk menguji sifat keterbagian dan kepositifan dari bilangan integer yang diinput kita (a), menyimpan hasilnya dalam variabel boolean (b), yang nilai default-nya adalah false. Setelah membaca input a, program akan memeriksa kondisi utama if $a > 0 \&& a \% 2 == 0$. Kondisi ini mengharuskan a menjadi bilangan bulat positif dan genap secara bersamaan. Jika kedua syarat ini terpenuhi, variabel b akan diatur nilainya menjadi true.

Di dalam blok kondisional utama, terdapat pernyataan if $a \% 2 != 0$. Blok ini secara logis merupakan kode yang mustahil dieksekusi karena kondisi di luar sudah memastikan a adalah genap, sehingga a tidak mungkin ganjil ($a \% 2 != 0$). Jika blok mustahil ini dieksekusi, b akan disetel kembali ke false. Fungsi utama dari program ini adalah mengidentifikasi dan mencetak true hanya untuk bilangan yang positif dan genap (misalnya 4); untuk input lainnya (negatif, nol, atau ganjil), program akan mencetak false karena kondisinya tidak terpenuhi.

TUGAS

- Tugas 1**
- Source code

```
package main

import "fmt"

func main() {
    var jumlahorang, jumlahmotor int

    fmt.Print("masukkan jumlah orang: ")

    fmt.Scan(&jumlahorang)

    if jumlahorang%2 == 0 {
        jumlahmotor = jumlahorang / 2
    } else {
        jumlahmotor = (jumlahorang / 2) + 1
    }

    fmt.Println("jumlah motor yang dibutuhkan: ", jumlahmotor)
}
```

Screenshot program

The screenshot shows the Visual Studio Code interface with a Go project named "Latihan GO 1". The Explorer sidebar on the left lists files and folders: BERLATIH, Coba, lat, latihan9, latihan, main.go, latihan2, main.go, latihan3, main.go, Modul1, Modul2, modul3, Modul4, modul5, modul9, guided1, guided2, guided3, soal1, and soal2. The terminal at the bottom right shows the following command-line session:

```

PS D:\Coding\Latihan GO 1> go run "d:/Coding/Latihan GO 1/modul9/soal1/main.go"
masukkan jumlah orang: 10
jumlah motor yang dibutuhkan: 5
PS D:\Coding\Latihan GO 1> go run "d:/Coding/Latihan GO 1/modul9/soal1/main.go"
masukkan jumlah orang: 1
jumlah motor yang dibutuhkan: 1
PS D:\Coding\Latihan GO 1>

```

Deskripsi program

Program Go ini, dirancang untuk menghitung jumlah minimum motor yang dibutuhkan untuk mengangkut sejumlah orang. Fungsi `main()` mendeklarasikan dua variabel integer, `jumlahorang` (input) dan `jumlahmotor` (output). Program pertama-tama meminta kita memasukkan jumlah orang menggunakan `fmt.Print()` dan membaca input tersebut menggunakan `fmt.Scan()`. Motor di sini diasumsikan dapat menampung dua orang (satu motor, satu pengendara, satu penumpang). Logika perhitungan didasarkan pada dua kasus menggunakan pernyataan `if-else`.

Kasus pertama, diuji dengan `if jumlahorang % 2 == 0`, memeriksa apakah jumlah orang genap (habis dibagi 2). Jika genap, jumlah motor yang dibutuhkan hanyalah setengah dari jumlah orang tersebut (`jumlahmotor = jumlahorang / 2`). Kasus kedua, ditangani oleh blok `else`, berlaku jika jumlah orang ganjil. Dalam kasus ganjil, perhitungan yang dilakukan adalah membagi jumlah orang dengan 2, lalu hasilnya ditambahkan 1 (`jumlahmotor = (jumlahorang / 2) + 1`). Penambahan 1 ini memastikan bahwa satu motor ekstra disediakan untuk orang terakhir yang tersisa. Terakhir, program mencetak hasil perhitungan tersebut ke konsol. Sebagai contoh, 10 orang membutuhkan 5 motor, sedangkan 1 orang membutuhkan 1 motor.

2. Tugas 2

Source code

```

package main

import "fmt"

```

```

func main() {

    var bilangan int

    var teks string

    fmt.Scan(&bilangan)

    if bilangan%2 == 0 && bilangan < 0 {

        teks = "genap negatif"

    } else {

        teks = "bukan"

    }

    fmt.Println(teks)

}

```

Screenshot program

The screenshot shows a Go development environment with the following components:

- EXPLORER**: Shows a file tree for "LATHIAN GO 1" containing various Go files (main.go, modul1, modul2, modul3, modul4, modul5, modul9, guided1, guided2, guided3, soal1, soal2, soal3, soalA) and a "code.png" image.
- CODE EDITOR**: Displays the source code of "main.go" with syntax highlighting for keywords like `func`, `var`, `if`, and `fmt`. The code is identical to the one shown in the text block above.
- TERMINAL**: Shows command-line output from running the program. It includes the command `go run` followed by the path to the main.go file in the "soal2" directory. The output shows the program's logic: it asks for input, checks if it's even and less than zero, and prints "genap negatif".

Deskripsi program

Program Go ini, bertujuan untuk menguji apakah bilangan integer yang diinput adalah bilangan genap negatif dan memberikan *label* teks yang sesuai. Fungsi main() mendeklarasikan variabel integer bilangan (untuk input) dan variabel *string* teks (untuk output, dengan nilai *default* "bukan"). Program kemudian meminta dan membaca input integer menggunakan fmt.Scan(&bilangan). Logika inti program menggunakan pernyataan if-else untuk mengevaluasi input berdasarkan kondisi majemuk: if bilangan % 2 == 0 && bilangan < 0.

Kondisi majemuk ini memeriksa apakah input bilangan adalah genap (bilangan % 2 == 0) dan negatif (bilangan < 0) secara bersamaan. Jika kedua syarat ini terpenuhi, variabel teks akan diatur menjadi "genap negatif". Jika kondisi tersebut tidak terpenuhi (misalnya, jika bilangan positif, nol, atau ganjil negatif), blok else akan dieksekusi, dan variabel teks akan diatur ke nilai "bukan". Terakhir, program mencetak nilai akhir dari variabel teks menggunakan fmt.Println(teks). Contoh eksekuksi menunjukkan: input -4 (genap dan negatif) menghasilkan "genap negatif", sedangkan input 10 (genap tapi positif) menghasilkan "bukan".

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var x, y int

    fmt.Print("masukkan x:")
    fmt.Scan(&x)
    fmt.Print("masukkan y:")
    fmt.Scan(&y)

    if y%x == 0 {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }

    if x%y == 0 {
        fmt.Println("true")
    } else {
        fmt.Println("false")
```

```
}
```

Screenshot program

The screenshot shows a code editor interface with multiple tabs open. The active tab is 'main.go' from the 'Latihan GO 1' folder. The code in 'main.go' is:

```
package main
import "fmt"
func main() {
    var x, y int
    fmt.Println("masukkan x:")
    fmt.Scan(&x)
    fmt.Println("masukkan y:")
    fmt.Scan(&y)
    if y/x == 0 {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}
```

The terminal tab shows the command 'go run "d:\Coding\Latihan GO 1\modul9\soal3\main.go"' being run twice. The first run prompts for 'x' and 'y', and prints 'false'. The second run prompts for 'x' and 'y', and prints 'true'.

Deskripsi program

Program Go ini, dirancang untuk menguji properti keterbagian (modulus) dari dua bilangan integer, x dan y , yang diinput kita. Fungsi `main()` mendeklarasikan dua variabel integer, x dan y . Program kemudian meminta pengguna untuk memasukkan nilai x dan y secara berurutan menggunakan `fmt.Println()` dan `fmt.Scan()`. Logika utama program menggunakan pernyataan `if-else` untuk mengevaluasi apakah x habis dibagi y , dan sebaliknya.

Pernyataan kondisional pertama adalah `if y % x == 0`. Kondisi ini memeriksa apakah y habis dibagi x (sisa pembagian y oleh x adalah nol). Jika kondisi ini benar, program akan mencetak "true". Jika kondisi ini salah, program akan masuk ke blok `else`. Di dalam blok `else` ini terdapat pemeriksaan kondisional kedua, `if x % y == 0`, yang memeriksa apakah x habis dibagi y . Jika x habis dibagi y , program mencetak "true". Jika kedua kondisi tersebut salah (yaitu y tidak habis dibagi x , dan x juga tidak habis dibagi y), program akan mencetak "false" (yang berada di blok `else` dari kondisional kedua). Dengan kata lain, program ini mencetak "true" jika salah satu bilangan merupakan kelipatan dari bilangan lainnya.