

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL PRAKTIKUM 9  
IF-THEN**



**Disusun oleh:**

**Didi Hermawanto**

**109082500088**

**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 bilangan int

    fmt.Scan(&bilangan)


    if bilangan < 0 {

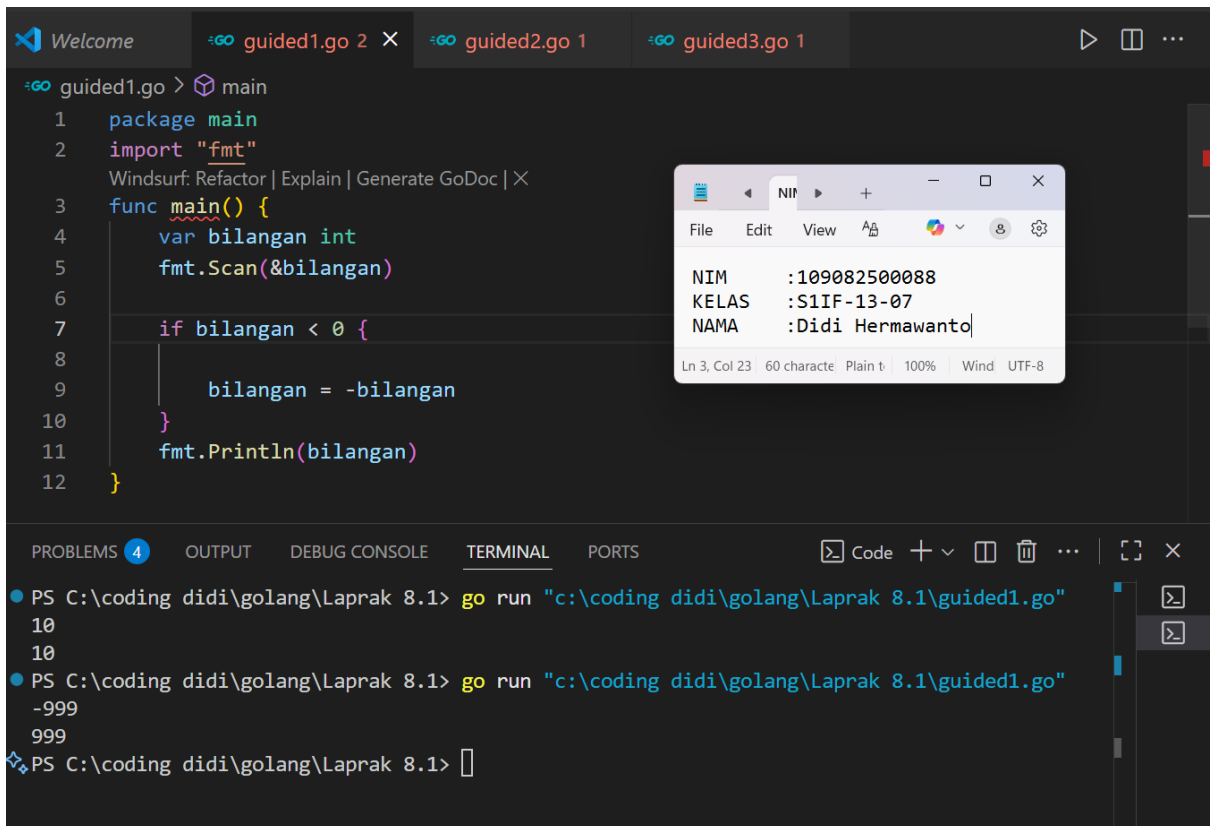
        bilangan = -bilangan

    }

    fmt.Println(bilangan)

}
```

#### Screenshoot program



```

Welcome
-go guided1.go 2 X -go guided2.go 1 -go guided3.go 1
-go guided1.go > main
1 package main
2 import "fmt"
3 func main() {
4     var bilangan int
5     fmt.Scan(&bilangan)
6
7     if bilangan < 0 {
8
9         bilangan = -bilangan
10    }
11    fmt.Println(bilangan)
12 }

NIP
File Edit View A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
NIM :109082500088
KELAS :SIIF-13-07
NAMA :Didi Hermawanto
Ln 3, Col 23 60 character Plain text 100% Wind UTF-8

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\guided1.go"
10
10
PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\guided1.go"
-999
999
PS C:\coding didi\golang\Laprak 8.1> 
```

#### Deskripsi program

Program Go tersebut berfungsi untuk mengambil sebuah input bilangan bulat dari pengguna, kemudian memeriksa apakah bilangan tersebut bernilai negatif, dan jika iya maka bilangan itu diubah menjadi positif, sehingga output yang dihasilkan selalu berupa nilai absolut dari angka yang dimasukkan. Program dimulai dengan mengimpor paket `fmt` untuk keperluan input-output, lalu mendeklarasikan variabel bilangan sebagai penampung data yang dimasukkan pengguna melalui `fmt.Scan`. Setelah nilai diterima, program menjalankan percabangan `if` untuk mengecek apakah `bilangan < 0`; apabila kondisi ini benar, maka nilai tersebut dikalikan dengan `-1` melalui operasi `bilangan = -bilangan` sehingga berubah menjadi positif. Jika bilangan yang dimasukkan sudah positif atau nol, program melewati blok kondisi tanpa perubahan. Terakhir, program mencetak nilai bilangan yang sudah dipastikan positif. Secara algoritmik, alurnya dapat disederhanakan menjadi: mulai → baca input bilangan → periksa apakah bilangan negatif → jika negatif, ubah menjadi positif → tampilkan hasil → selesai. Program ini merupakan implementasi manual dari cara kerja nilai absolut pada bilangan bulat.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {
    var bilangan int
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&bilangan)

    if bilangan > 0 {
        fmt.Println("Bilangan positif")
    } else if bilangan < 0 {
        fmt.Println("Bilangan negatif")
    } else if bilangan == 0 {
        fmt.Println("bukan positif")
    }
}
```

## Screenshoot program

```
10     if bilangan > 0 {
11         fmt.Println("Bilangan positif")
12     } else if bilangan < 0 {
13         fmt.Println("Bilangan negatif")
14     } else if bilangan == 0 {
15         fmt.Println("bukan positif")
16     }
17 }
18
```

NIP

NIM	:109082500088
KELAS	:S1IF-13-07
NAMA	:Didi Hermawanto

Ln 3, Col 23 60 character Plain t 100% Wind UTF-8

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\guided2.go"  
Masukkan bilangan: 10  
Bilangan positif
- PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\guided2.go"  
Masukkan bilangan: -3  
Bilangan negatif
- PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\guided2.go"  
Masukkan bilangan: 5  
Bilangan positif

Ln 17, Col 2 Tab Size: 4 UTF-8 CRLF Go 1.25.1 Go Live Windsurf: Login Prettier

## Deskripsi program :

Program Go di atas berfungsi untuk membaca sebuah bilangan bulat dari input pengguna, kemudian mengidentifikasi apakah bilangan tersebut termasuk kategori **positif**, **negatif**, atau **nol**. Program dimulai dengan memanggil paket `fmt` sebagai sarana untuk menampilkan teks dan menerima input. Setelah variabel `bilangan` dideklarasikan, program meminta pengguna memasukkan sebuah angka melalui `fmt.Scan`. Selanjutnya, program menjalankan percabangan berantai menggunakan `if`, `else if`, dan pemeriksaan nilai. Jika bilangan yang dimasukkan lebih besar dari nol, program menampilkan pesan bahwa bilangan tersebut adalah bilangan positif. Jika nilai yang dimasukkan kurang dari nol, program menampilkan bahwa bilangan tersebut adalah bilangan negatif. Namun apabila nilai yang diberikan tepat sama dengan nol, program menampilkan pesan "bukan positif", yang menunjukkan bahwa nol tidak termasuk ke dalam kategori bilangan positif maupun negatif. Secara algoritma, alur programnya kurang lebih seperti ini: mulai → terima input bilangan → cek apakah bilangan lebih dari nol → jika ya, tampilkan "positif" → jika tidak, cek apakah bilangan kurang dari nol → jika ya, tampilkan "negatif" → jika tidak, berarti bilangan bernilai nol dan tampilkan pesan yang sesuai → selesai. Program ini memberikan cara sederhana untuk memahami konsep klasifikasi bilangan berdasarkan tanda.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var angka int

    var hasil bool

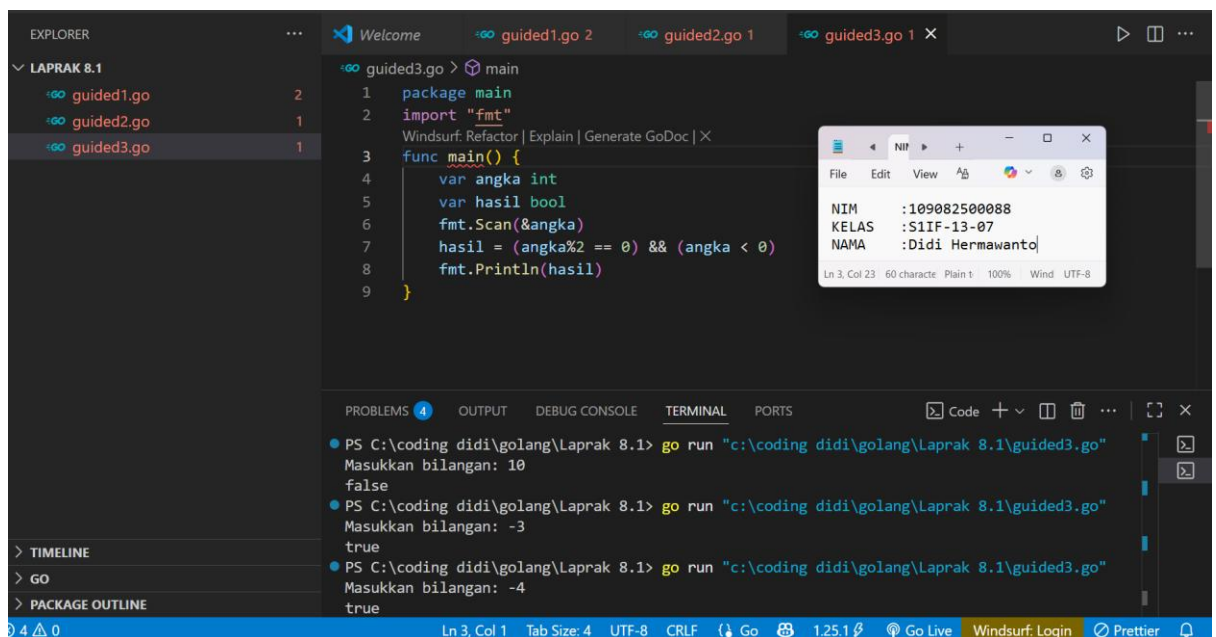
    fmt.Scan(&angka)

    hasil = (angka%2 == 0) && (angka < 0)

    fmt.Println(hasil)

}
```

#### Screenshoot program



#### Deskripsi program

Program Go tersebut digunakan untuk menentukan apakah sebuah bilangan memenuhi dua kondisi sekaligus, yaitu **genap** dan **bernilai negatif**. Program dimulai dengan mengimpor paket `fmt`, kemudian mendeklarasikan dua variabel: `angka` sebagai bilangan bulat yang akan diterima dari input pengguna, dan `hasil` sebagai variabel bertipe boolean untuk menyimpan nilai benar atau salah. Setelah pengguna memasukkan sebuah bilangan melalui `fmt.Scan`, program melakukan pengecekan

logika menggunakan ekspresi `(angka%2 == 0) && (angka < 0)`. Bagian `angka%2 == 0` digunakan untuk mengecek apakah angka tersebut genap, sedangkan `angka < 0` mengecek apakah bilangan tersebut negatif. Operator `&&` memastikan bahwa hasil hanya akan bernilai true jika **kedua syarat** terpenuhi secara bersamaan; jika salah satunya tidak terpenuhi, maka hasil akan menjadi false. Terakhir, program mencetak nilai boolean tersebut sebagai output. Secara algoritma, alurnya dapat ditulis seperti ini: mulai → baca input bilangan → periksa apakah bilangan genap → periksa apakah bilangan negatif → kombinasikan kedua kondisi dengan operator AND → tampilkan hasil → selesai. Program ini membantu memahami bagaimana operator logika dan operasi modulo bekerja untuk menentukan karakteristik sebuah bilangan.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var orang int

    fmt.Print("Masukkan jumlah orang: ")

    fmt.Scan(&orang)

    motor := orang / 2

    if orang%2 != 0 {
        motor++
    }

    fmt.Println("Jumlah motor yang diperlukan:", motor)
}
```

#### Screenshoot program

```
func main() {
    var orang int
    fmt.Print("Masukkan jumlah orang: ")
    fmt.Scan(&orang)

    motor := orang / 2
    if orang%2 != 0 {
        motor++
    }

    fmt.Println("Jumlah motor yang diperlukan:", motor)
}
```

```
PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\tugas1.go"
Masukkan jumlah orang: 10
Jumlah motor yang diperlukan: 5
PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\tugas1.go"
Masukkan jumlah orang: 5
Jumlah motor yang diperlukan: 3
PS C:\coding didi\golang\Laprak 8.1> go run "c:\coding didi\golang\Laprak 8.1\tugas1.go"
Masukkan jumlah orang: 25
Jumlah motor yang diperlukan: 13
```

### Deskripsi program :

Program Go ini menghitung berapa banyak motor yang diperlukan untuk sebuah touring dengan aturan satu motor menampung maksimal dua orang (satu pengemudi + satu penumpang). Setelah mengimpor paket fmt, program membaca input berupa bilangan bulat orang dari pengguna menggunakan fmt.Scan; nilai ini diasumsikan jumlah peserta (biasanya nol atau positif). Untuk menentukan jumlah motor program melakukan pembagian integer  $\text{motor} := \text{orang} / 2$  sehingga setiap dua orang dianggap mengisi satu motor penuh; kemudian dicek sisa dengan `if orang % 2 != 0 { motor++ }` — jika ada satu orang tersisa (jumlah ganjil) ditambahkan satu motor lagi agar orang tersebut punya tempat. Dengan begitu contoh-contoh yang diberikan menghasilkan:  $10 \rightarrow 10/2 = 5$  motor,  $1 \rightarrow 1/2 = 0$  ditambah 1 = 1 motor,  $25 \rightarrow 25/2 = 12$  ditambah 1 = 13 motor, dan  $9 \rightarrow 9/2 = 4$  ditambah 1 = 5 motor.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {
    var angka int

    fmt.Print("Masukkan bilangan: ")
}
```

```

    fmt.Scan(&angka)

    if angka%2 == 0 && angka < 0 {
        fmt.Println("genap negatif")
    } else {
        fmt.Println("bukan")
    }
}

```

### Screenshoot program

The screenshot shows a Go IDE with the following components:

- EXPLORER:** Lists files under 'LAPRAK 8.1': guided1.go, guided2.go, guided3.go, tugas1.go, and tugas2.go.
- EDITOR:** Displays the source code for `tugas2.go`. The code imports `fmt`, declares `angka` as an `int`, and uses `fmt.Scan` to read input. It then checks the condition `angka%2 == 0 && angka < 0` to print either "genap negatif" or "bukan".
- TERMINAL:** Shows the execution of the program. It prompts "Masukkan bilangan: " and shows three runs:
  - Input: -4 → Output: genap negatif
  - Input: 0 → Output: bukan
  - Input: -2 → Output: bukan
- STATUS BAR:** Shows the current file is `tugas2.go` at line 9, column 1, with a tab size of 4 and UTF-8 encoding.

### Deskripsi program :

Program Go ini menerima satu bilangan bulat dari pengguna lalu menentukan apakah bilangan itu tergolong **genap negatif** atau **bukan**; caranya sederhana: setelah membaca input ke variabel `angka`, program mengecek dua kondisi sekaligus—apakah `angka % 2 == 0` (artinya habis dibagi 2, sehingga genap) dan apakah `angka < 0` (artinya bernilai negatif)—dengan operator logika `&&` yang memastikan kedua syarat harus benar supaya hasilnya genap negatif, jika tidak terpenuhi maka program mencetak bukan. Secara langkah-langkah: mulai → baca satu bilangan bulat → hitung sisa bagi `angka % 2` untuk menentukan genap/ganjil → bandingkan angka dengan nol untuk menentukan tanda → jika kedua kondisi benar keluarkan "genap negatif", selain itu keluarkan "bukan" → selesai. Perilaku khusus yang perlu dicatat: nol (0) adalah genap tetapi bukan negatif sehingga akan menghasilkan "bukan"; bilangan positif (mis. 10)



juga menghasilkan “bukan”; bilangan seperti -4 dan -2 memenuhi kedua syarat sehingga akan menampilkan “genap negatif”.

### Tugas 3

#### Source code

```
package main

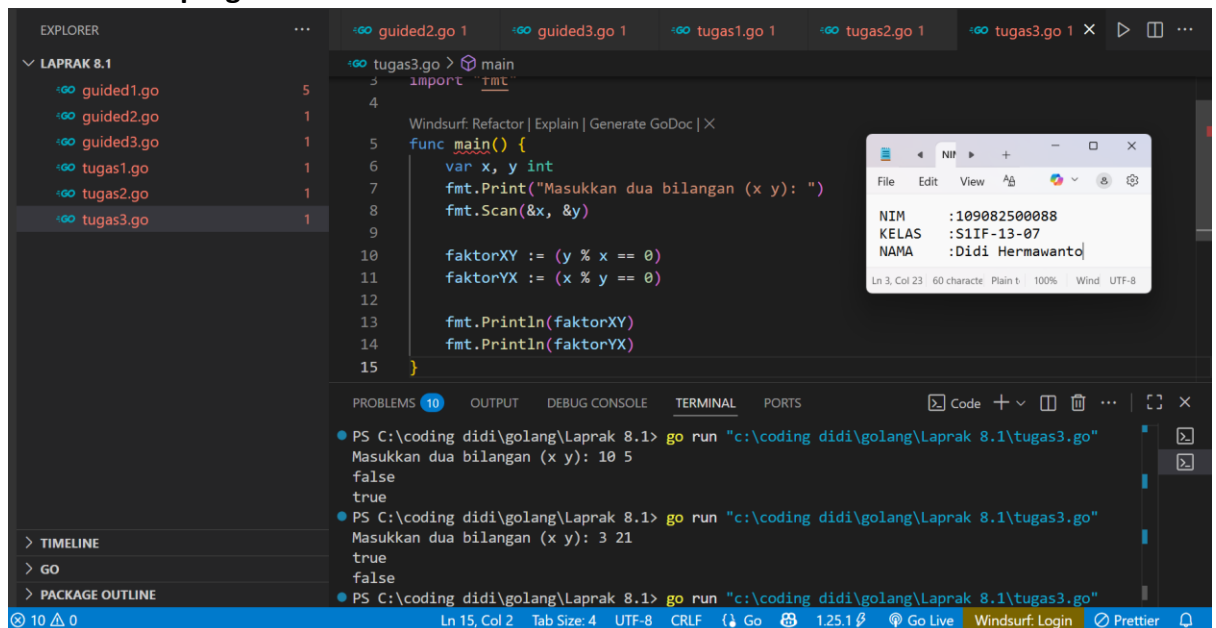
import "fmt"

func main() {
    var x, y int
    fmt.Print("Masukkan dua bilangan (x y): ")
    fmt.Scan(&x, &y)

    faktorXY := (y % x == 0)
    faktorYX := (x % y == 0)

    fmt.Println(faktorXY)
    fmt.Println(faktorYX)
}
```

#### Screenshoot program



#### Deskripsi program

Program Go ini dirancang untuk mengevaluasi hubungan faktor antara dua bilangan bulat positif, yaitu apakah **x merupakan faktor dari y** dan apakah **y merupakan faktor dari x**. Program diawali dengan mengimpor paket fmt sebagai alat untuk menerima input dan menampilkan output. Setelah itu, dua variabel x dan y disiapkan untuk menampung bilangan yang dimasukkan oleh pengguna. Ketika pengguna memasukkan dua angka, program langsung melakukan dua pemeriksaan

menggunakan operasi modulo. Pemeriksaan pertama adalah  $y \% x == 0$ , yang berarti program mengecek apakah sisa pembagian  $y$  oleh  $x$  bernilai nol—hal ini menandakan bahwa  $x$  membagi  $y$  secara sempurna, sehingga  $x$  adalah faktor dari  $y$ . Hasil dari pengecekan ini disimpan pada variabel boolean `faktorXY`. Pemeriksaan kedua dilakukan dengan cara yang sama namun arah pembagiannya dibalik, yaitu  $x \% y == 0$ , untuk menentukan apakah  $y$  adalah faktor dari  $x$ ; hasilnya disimpan dalam `faktorYX`. Kedua hasil boolean tersebut kemudian dicetak secara terpisah pada dua baris, sesuai format keluaran yang ditetapkan. Program ini dapat menangani berbagai kondisi, seperti ketika kedua bilangan sama (yang otomatis membuat keduanya saling menjadi faktor), atau ketika salah satu bilangan lebih kecil tetapi tidak membagi bilangan lainnya. Alur lengkapnya adalah: mulai → baca  $x$  dan  $y$  → cek  $x$  sebagai faktor  $y$  → cek  $y$  sebagai faktor  $x$  → tampilkan hasil → selesai.