

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**MODUL 09**  
**IF-THEN**



**Disusun oleh:**  
**FAREL JULIYANDRA RESTU HERMAWAN**  
**109082530038**  
**S1IF-13-04**

**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 {

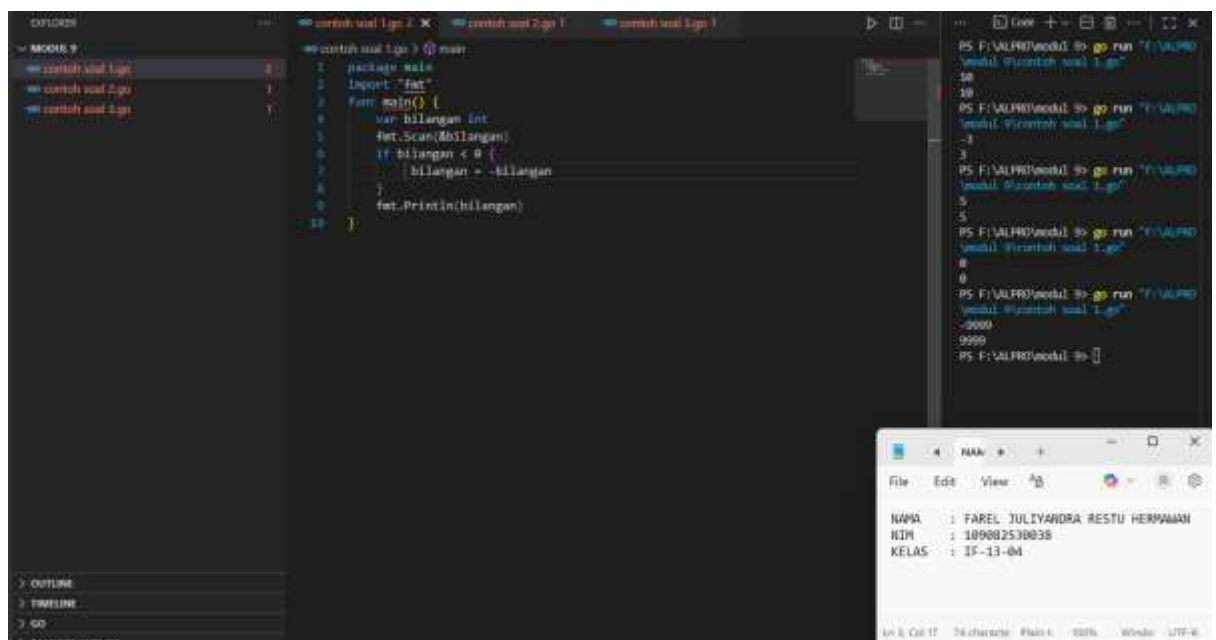
        bilangan = -bilangan

    }

    fmt.Println(bilangan)

}
```

### Screenshoot program



### Deskripsi program

Program Go di atas dimulai dengan mendeklarasikan variabel `bilangan` untuk menampung input dari pengguna. Setelah itu, program membaca nilai yang dimasukkan melalui `fmt.Scan(&bilangan)`. Selanjutnya, program melakukan pengecekan menggunakan `if` bilangan

< 0, yaitu untuk mengetahui apakah bilangan tersebut bernilai negatif. Jika benar negatif, maka bilangan diubah menjadi positif dengan membuatnya bernilai kebalikannya melalui `bilangan = -bilangan`. Jika tidak negatif, nilainya tetap. Terakhir, program mencetak hasil akhir bilangan menggunakan `fmt.Println(bilangan)`. Dengan alur tersebut, program memastikan bahwa output yang ditampilkan selalu dalam bentuk bilangan positif.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var bilangan int

    var teks string

    fmt.Scan(&bilangan)

    teks = "bukan positif"

    if bilangan > 0 {

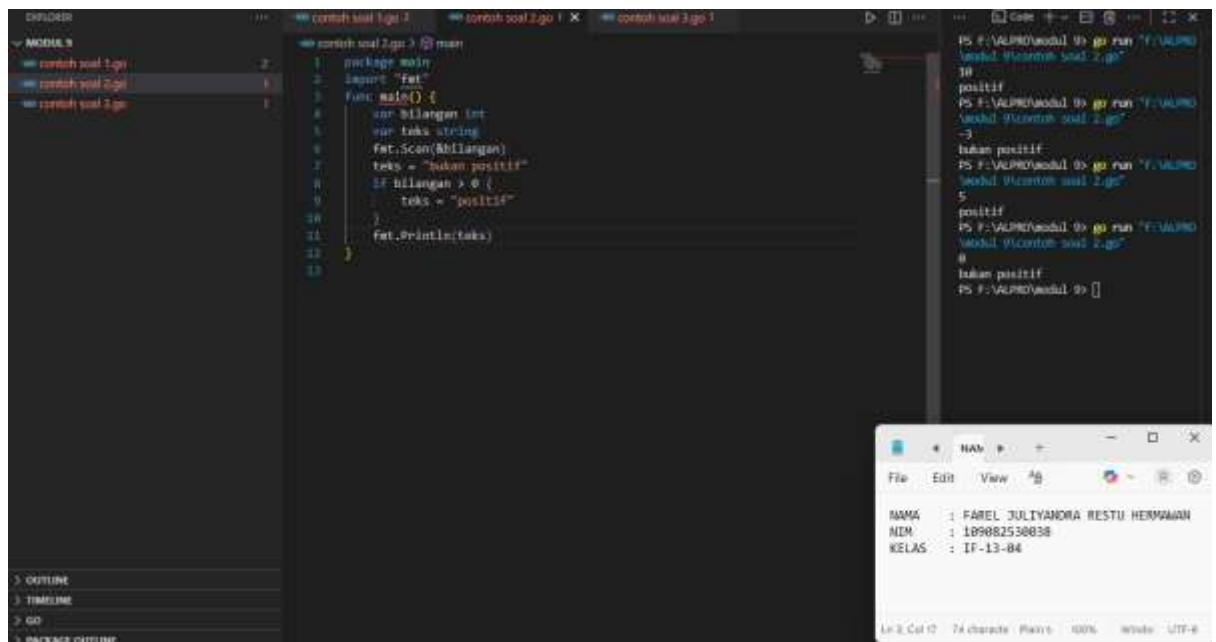
        teks = "positif"

    }

    fmt.Println(teks)

}
```

### Screenshoot program



### Deskripsi program

Program Go di atas dimulai dengan mendeklarasikan variabel bilangan sebagai input angka dan teks sebagai penampung hasil berupa string. Setelah pengguna memasukkan sebuah bilangan melalui `fmt.Scan(&bilangan)`, program langsung memberikan nilai awal untuk variabel teks, yaitu "bukan positif". Nilai awal ini akan menjadi keluaran default jika bilangan tidak lebih dari nol. Program kemudian melakukan pengecekan menggunakan kondisi `if bilangan > 0`; jika syarat ini terpenuhi, artinya bilangan tersebut positif, maka nilai variabel teks diubah menjadi "positif". Terakhir, program menampilkan isi variabel teks ke layar, sehingga output yang muncul sesuai dengan hasil evaluasi bilangan yang dimasukkan pengguna.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var bilangan int

    var hasil bool

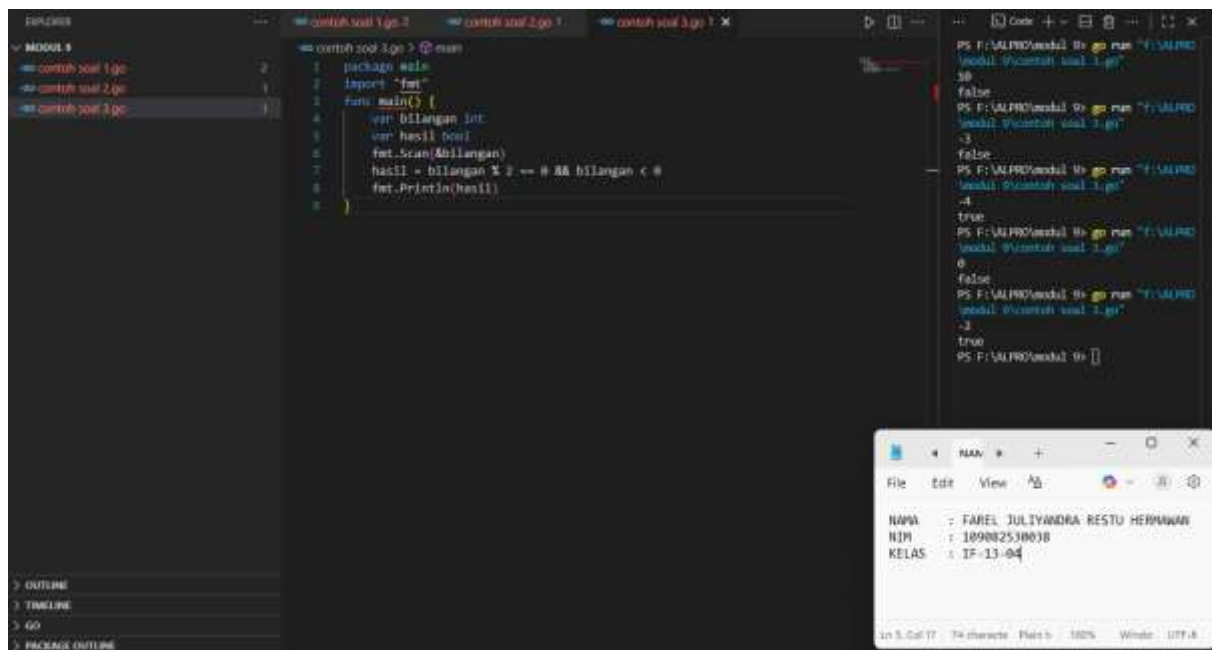
    fmt.Scan(&bilangan)

    hasil = bilangan % 2 == 0 && bilangan < 0

    fmt.Println(hasil)

}
```

## Screenshoot program



### Deskripsi program

Program Go di atas dimulai dengan mendeklarasikan variabel bilangan untuk menyimpan input angka dan hasil untuk menampung nilai boolean. Setelah pengguna memasukkan sebuah bilangan melalui `fmt.Scan(&bilangan)`, program langsung mengevaluasi dua kondisi sekaligus, yaitu apakah bilangan tersebut genap (`bilangan % 2 == 0`) dan apakah bilangan tersebut negatif (`bilangan < 0`). Kedua kondisi ini digabungkan dengan operator logika `&&`, sehingga hasil akan bernilai `true` hanya jika bilangan memenuhi kedua syarat tersebut. Jika salah satu kondisi tidak terpenuhi, maka hasil otomatis bernilai `false`. Pada bagian akhir, program mencetak nilai boolean tersebut ke layar, sehingga pengguna dapat mengetahui apakah bilangan yang dimasukkan merupakan bilangan genap negatif atau tidak.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {

    var n, m, s int

    fmt.Print("Masukan :")
```

```

    fmt.Scan(&n)

    m = n / 2

    s = n % 2

    if s != 0 {

        m++

    }

    fmt.Print("Keluaran : ",m)

}

```

### Screenshoot program

The screenshot shows a Go program in an IDE. The source code is as follows:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var n, m, s int
7     fmt.Print("Masukan : ")
8     fmt.Scan(&n)
9
10    m = n / 2
11    s = n % 2
12
13    if s != 0 {
14        m++
15    }
16    fmt.Print("Keluaran : ",m)
17 }

```

The output window shows the following execution results:

```

PS F:\VALEPRO\modul 9> go run "F:\VALEPRO\modul 9\latihan 1.go"
Masukan : 10
Keluaran : 5
PS F:\VALEPRO\modul 9> go run "F:\VALEPRO\modul 9\latihan 1.go"
Masukan : 11
Keluaran : 6
PS F:\VALEPRO\modul 9> go run "F:\VALEPRO\modul 9\latihan 1.go"
Masukan : 12
Keluaran : 6
PS F:\VALEPRO\modul 9> go run "F:\VALEPRO\modul 9\latihan 1.go"
Masukan : 13
Keluaran : 7
PS F:\VALEPRO\modul 9> go run "F:\VALEPRO\modul 9\latihan 1.go"
Masukan : 14
Keluaran : 7
PS F:\VALEPRO\modul 9> go run "F:\VALEPRO\modul 9\latihan 1.go"
Masukan : 15
Keluaran : 8
PS F:\VALEPRO\modul 9>

```

### Deskripsi program

Program Go di atas digunakan untuk menghitung jumlah motor yang diperlukan berdasarkan jumlah orang yang akan berangkat. Pertama, program mendeklarasikan variabel orang untuk menampung input jumlah orang dari pengguna. Setelah nilai dibaca melalui `fmt.Scan`, program menghitung jumlah motor awal dengan membagi jumlah orang per dua, karena satu motor dapat menampung dua orang. Selanjutnya, program melakukan pengecekan menggunakan kondisi `if orang%2 != 0` untuk mengetahui apakah jumlah orang bernilai ganjil; jika ganjil, berarti masih ada satu orang yang belum mendapat motor sehingga variabel motor ditambah satu. Terakhir, program menampilkan jumlah motor yang harus disiapkan sesuai perhitungan tersebut.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var x int

    var teks string

    fmt.Print("masukan :")

    fmt.Scan(&x)

    teks = "bukan"

    if x < 0 {

        x = x / 2

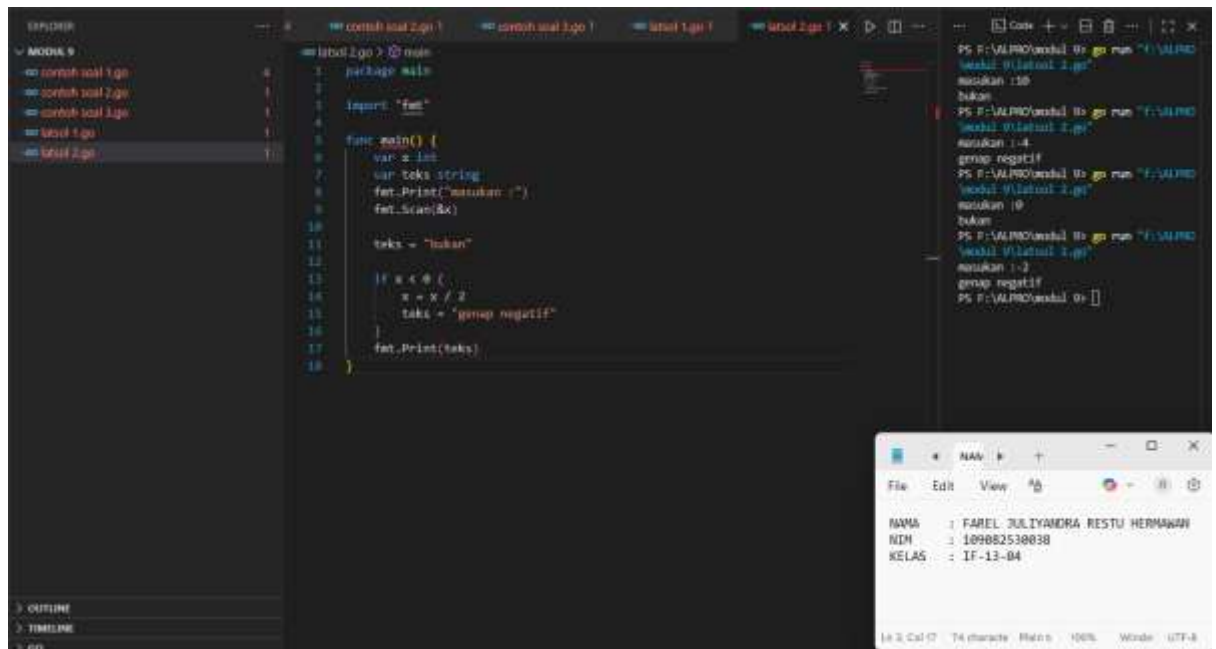
        teks = "genap negatif"

    }

    fmt.Print(teks)

}
```

### Screenshoot program



### Deskripsi program

Program Go di atas berfungsi untuk mengecek apakah sebuah bilangan termasuk kategori “genap negatif”. Program dimulai dengan mendeklarasikan variabel *n* untuk menampung input dari pengguna. Setelah nilai bilangan dimasukkan melalui `fmt.Scan(&n)`, program masuk ke proses pengecekan menggunakan kondisi `if n < 0 && n%2 == 0`. Kondisi ini memeriksa dua hal sekaligus: apakah bilangan tersebut bernilai negatif ( $n < 0$ ) dan apakah bilangan tersebut genap ( $n\%2 == 0$ ). Jika kedua syarat terpenuhi, program mencetak “genap negatif”. Namun, jika salah satu atau kedua kondisi tidak terpenuhi, bagian `else` akan dijalankan dan program mencetak “bukan”. Dengan cara ini, program memastikan output sesuai dengan karakteristik bilangan yang dimasukkan pengguna.

### 3. Tugas 3

#### Source code

```

package main

import "fmt"

func main(){
    var x, y, a,b int
    fmt.Print("Masukan :")
    fmt.Scan(&x,&y)

    a = y % x
    b = x % y

```



```

        if a == 0{
            fmt.Println("true")
        }else {
            fmt.Println("false")
        }

        if b == 0 {
            fmt.Println("true")
        }else {
            fmt.Println("false")
        }
    }
}

```

### Screenshoot program

The screenshot shows a Go program in an IDE. The code defines a `main` function that reads two integers `x` and `y` from the command line. It then checks if `y` is divisible by `x` (`y % x == 0`) and prints `true` or `false`. It then checks if `x` is divisible by `y` (`x % y == 0`) and prints `true` or `false`. The terminal output shows the program being run with inputs `10 5` and `5 10`, resulting in `false true` and `true true` respectively.

```

package main

import "fmt"

func main() {
    var x, y, a, b int
    fmt.Print("Masukan :")
    fmt.Scan(&x, &y)

    a = y % x
    b = x % y

    if a == 0 {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }

    if b == 0 {
        fmt.Println("true")
    } else {
        fmt.Println("false")
    }
}

```

```

PS F:\ALFMD\modul 3> go run "F:\ALFMD\modul 3\latasol 3.go"
Masukan :10 5
false
true
PS F:\ALFMD\modul 3> go run "F:\ALFMD\modul 3\latasol 3.go"
Masukan :5 10
true
true
PS F:\ALFMD\modul 3>

```

### Deskripsi program

Program Go di atas digunakan untuk mengecek hubungan faktor antara dua bilangan, yaitu `x` dan `y`, yang dimasukkan oleh pengguna. Setelah kedua nilai dibaca melalui `fmt.Scan(&x, &y)`, program melakukan pengecekan pertama dengan kondisi `if y%x == 0` untuk menentukan apakah `x` merupakan faktor dari `y`; jika sisa pembagian `y` oleh `x` adalah nol, program mencetak `true`, jika tidak maka mencetak `false`. Selanjutnya, program melakukan pengecekan kebalikan menggunakan kondisi `if x%y == 0` untuk mengetahui apakah `y` merupakan faktor dari `x`. Sama seperti sebelumnya, jika hasil pembagian tidak bersisa maka program menampilkan `true`, dan jika bersisa maka menampilkan `false`. Dengan dua pengecekan ini, program memberikan informasi apakah kedua angka tersebut saling menjadi faktor satu sama lain atau tidak.