

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 9**

**IF-THEN**



**Disusun oleh:**

**ALIN KARISA HIZANNAH**

**109082500010**

**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.Print("Input :")

    fmt.Scan(&a)

    if a < 0 {

        a = a * -1

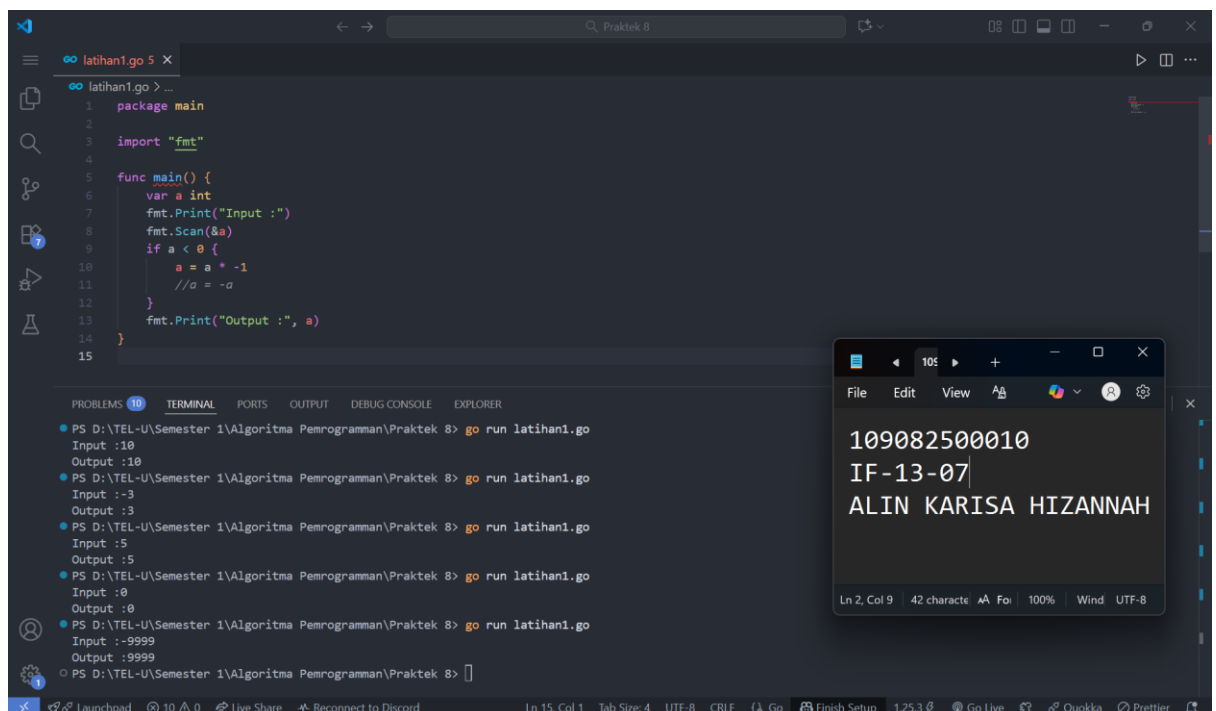
        //a = -a

    }

    fmt.Print("Output :", a)

}
```

#### Screenshoot program



### Deskripsi program

Program ini digunakan untuk menampilkan nilai absolut dari sebuah bilangan yang dimasukkan pengguna. Saat pengguna memasukkan sebuah angka, program akan mengecek apakah angka tersebut bernilai negatif. Jika ternyata negatif, angka tersebut akan dikalikan dengan  $-1$  sehingga berubah menjadi positif.

Setelah proses pengecekan dan pengubahan selesai, program langsung menampilkan hasil akhirnya. Intinya, apa pun angka yang dimasukkan—baik itu positif atau negatif—program akan memastikan bahwa output yang keluar adalah bentuk positifnya.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    //Cara 1 If-Else-Then

    // var a int

    // fmt.Print("Input :")

    // fmt.Scan(&a)

    // if a > 0 {

    //     fmt.Print("Positif")

    // } else {

    //     fmt.Print("Bukan Positif")

    // }

    //Cara 2 If-Then

    var a int

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

```

    fmt.Scan(&a)

    Output := "Bukan Positif"

    if a > 0 {

        Output = "Positif"

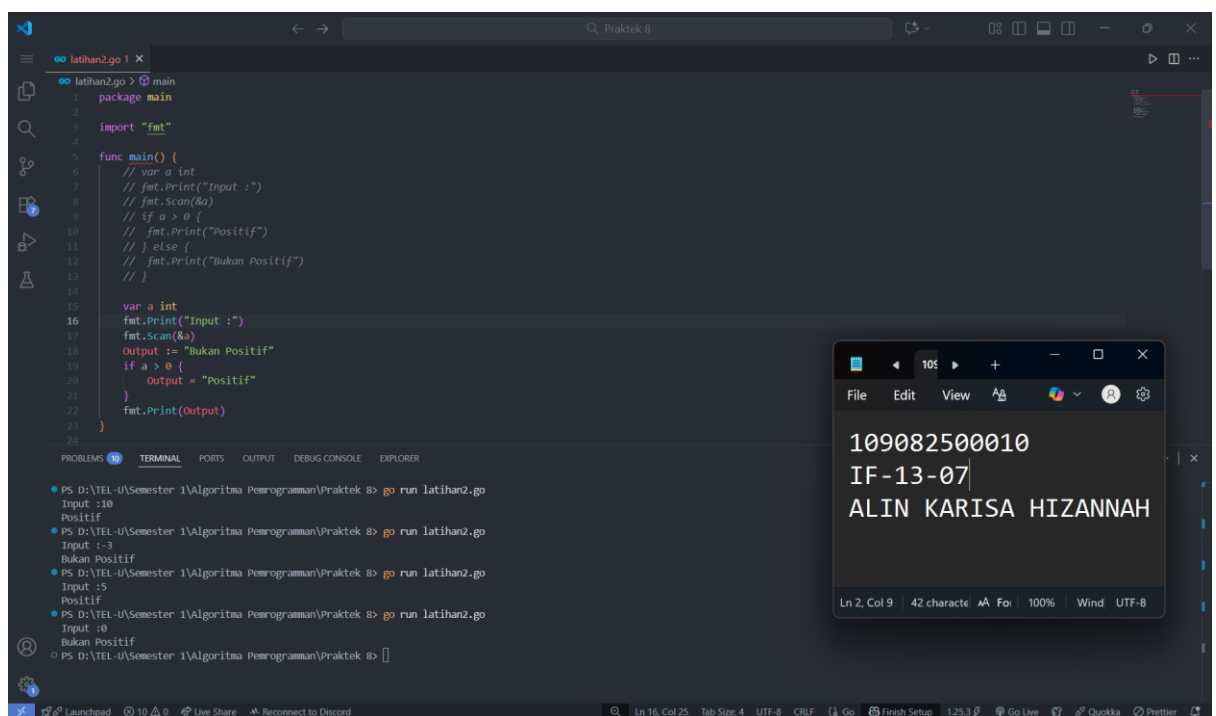
    }

    fmt.Print(Output)

}

```

### Screenshoot program



### Deskripsi program

Program ini digunakan untuk mengecek apakah suatu bilangan termasuk bilangan positif atau bukan. Ketika pengguna memasukkan sebuah angka, program akan membandingkannya dengan angka nol. Jika nilai yang dimasukkan lebih besar dari nol, maka program akan memberi output "Positif". Sebaliknya, jika angka tersebut nol atau negatif, output yang ditampilkan adalah "Bukan Positif".

Pada bagian akhir kode digunakan pendekatan yang lebih ringkas, yaitu dengan menyimpan hasil pengecekan ke dalam variabel Output. Variabel ini awalnya diisi dengan teks "Bukan Positif", kemudian diubah menjadi "Positif" jika kondisi  $a > 0$  terpenuhi. Cara ini membuat program lebih sederhana dan mudah dibaca, sekaligus tetap memberikan hasil pemeriksaan yang akurat.

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    //Cara 1

    // var a int

    // fmt.Print("Input :")

    // fmt.Scan(&a)

    // Output := false

    // if a < 0 && a%2 == 0 {

    //   Output = true

    // }

    // fmt.Print(Output)


    //Cara 2

    var bilangan int

    var hasil bool

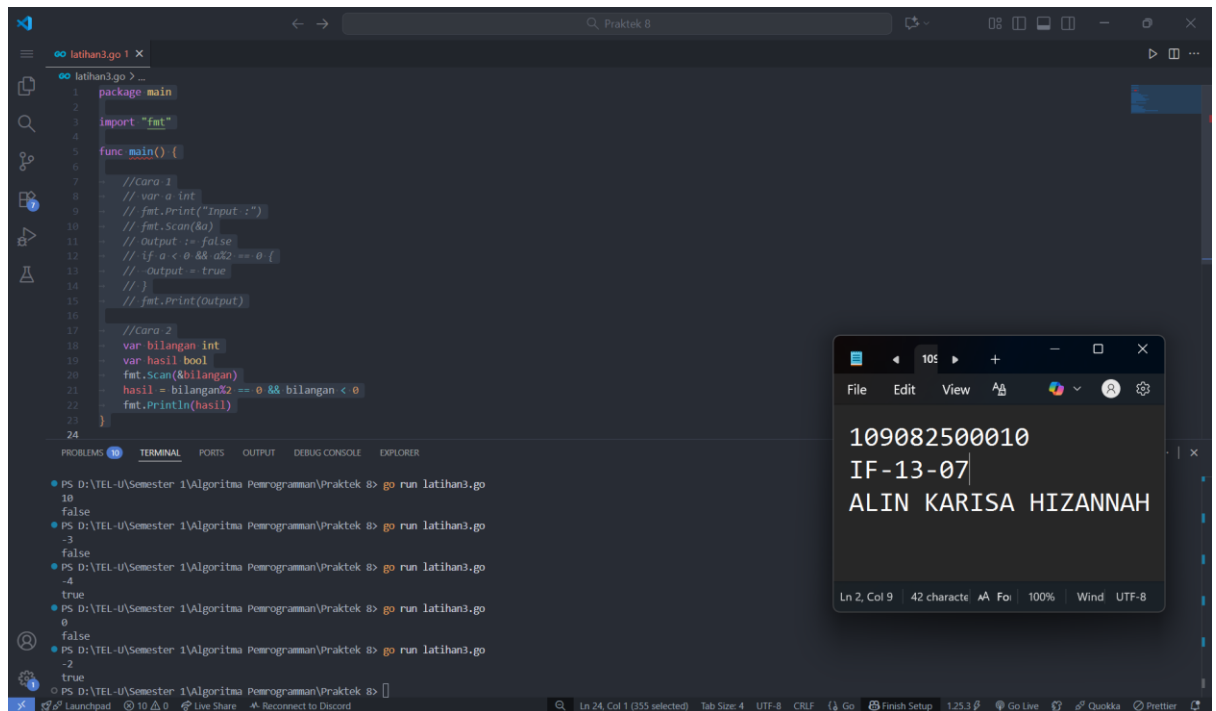
    fmt.Scan(&bilangan)

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

    fmt.Println(hasil)

}
```

**Screenshoot program**



### Deskripsi program

Program ini digunakan untuk menentukan apakah sebuah bilangan termasuk bilangan genap negatif atau tidak. Ketika pengguna memasukkan sebuah angka, program akan mengecek dua kondisi sekaligus: apakah bilangan tersebut habis dibagi dua (genap) dan apakah nilainya kurang dari nol (negatif). Jika kedua syarat ini terpenuhi, hasilnya adalah true. Jika salah satu syarat saja tidak terpenuhi, maka hasilnya false.

Pada bagian yang aktif digunakan, pengecekan dibuat lebih ringkas dengan langsung menaruh ekspresi logika ke dalam variabel hasil. Ekspresi `bilangan%2 == 0 && bilangan < 0` secara otomatis akan menghasilkan nilai true atau false tanpa perlu menulis struktur if. Cara ini membuat kode lebih efisien, mudah dibaca, dan tetap memberikan hasil yang akurat.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {

    var orang int
```

```

    fmt.Print("Masukkan jumlah orang yang akan touring:
    ")

    fmt.Scan(&orang)

    motor := orang / 2

    if orang%2 != 0 {

        motor++

    }

    fmt.Printf("Jumlah motor yang diperlukan: %d\n",
    motor)

}

```

### Screenshoot program

The screenshot shows the Go source code in VS Code and its execution in the terminal. The code calculates the number of motorcycles needed based on the number of people, rounding up. The terminal shows three test runs with inputs 10, 1, and 25, resulting in 5, 1, and 13 motorcycles respectively. A separate text box displays the input '109082500010', the date 'IF-13-07', and the name 'ALIN KARISA HIZANNAH'.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var orang int
7
8     fmt.Print("Masukkan jumlah orang yang akan touring: ")
9     fmt.Scan(&orang)
10
11     motor := orang / 2
12
13     if orang%2 != 0 {
14         motor++
15     }
16
17     fmt.Printf("Jumlah motor yang diperlukan: %d\n", motor)
18 }

```

```

PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8> go run soal1.go
Masukkan jumlah orang yang akan touring: 10
Jumlah motor yang diperlukan: 5
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8> go run soal1.go
Masukkan jumlah orang yang akan touring: 1
Jumlah motor yang diperlukan: 1
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8> go run soal1.go
Masukkan jumlah orang yang akan touring: 25
Jumlah motor yang diperlukan: 13
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8> go run soal1.go
Masukkan jumlah orang yang akan touring: 9
Jumlah motor yang diperlukan: 5
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8>

```

109082500010  
IF-13-07  
ALIN KARISA HIZANNAH

### Deskripsi program

Program ini bertujuan untuk menghitung jumlah motor yang dibutuhkan untuk kegiatan touring berdasarkan jumlah orang yang ikut. Setiap motor dianggap dapat menampung dua orang, sehingga langkah pertama adalah membagi jumlah orang

dengan 2. Hasil pembagian ini menunjukkan berapa motor minimal yang bisa digunakan jika semuanya genap.

Namun, jika jumlah orang yang ikut ternyata ganjil, maka akan ada satu orang yang tidak mendapatkan pasangan. Untuk mengakomodasi hal ini, program menambahkan satu motor tambahan agar semua peserta tetap mendapat tempat. Setelah seluruh perhitungan dilakukan, jumlah motor yang diperlukan ditampilkan ke layar dengan format yang jelas dan mudah dipahami.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var bilangan int

    fmt.Print("Masukkan bilangan: ")

    fmt.Scan(&bilangan)

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

        fmt.Println("genap negatif")

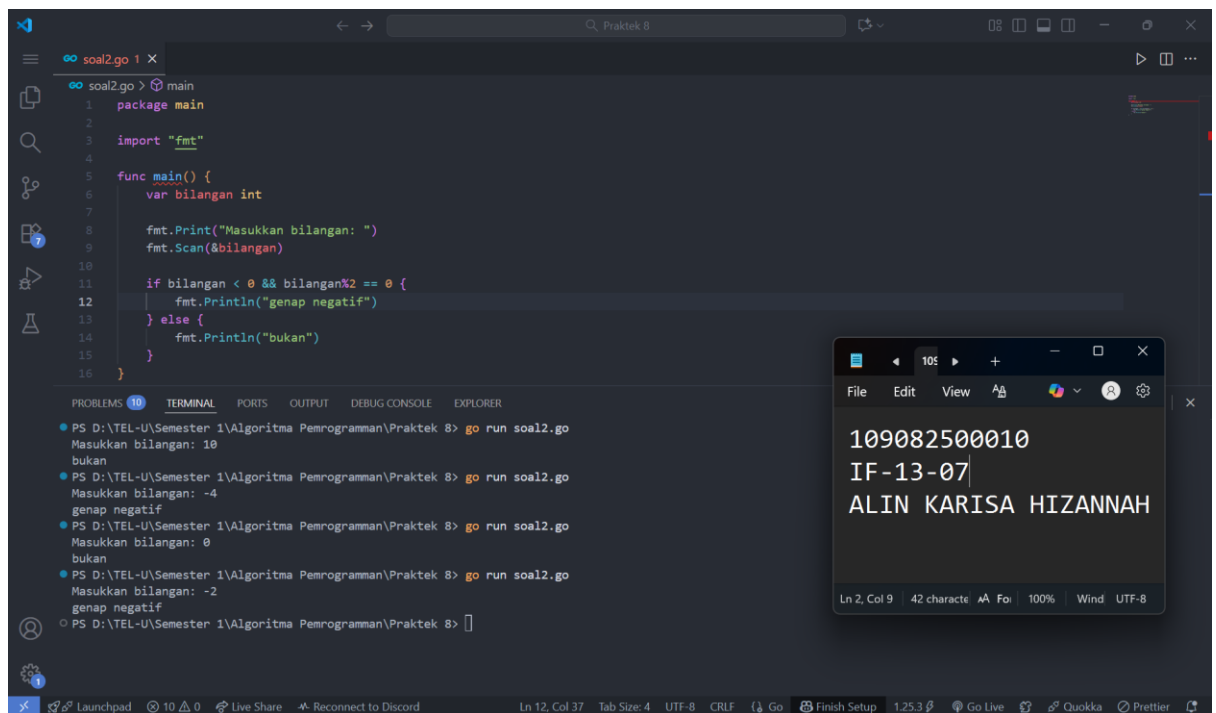
    } else {

        fmt.Println("bukan")

    }

}
```

### Screenshoot program



## Deskripsi program

### 3. 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)

    fmt.Println(y%x == 0)

    fmt.Println(x%y == 0)
}
```

#### Screenshoot program

The screenshot shows a Go program in a VS Code editor. The program is named `soal3.go` and is located in a package named `main`. It imports the `fmt` package and defines a `main` function. Inside the `main` function, it declares two integer variables `x` and `y`. It then prompts the user to enter two numbers, reads the input, and checks if `y` is a multiple of `x` using the expression `y%x == 0`. If true, it prints `109082500010`; if false, it prints `IF-13-07`. Finally, it checks if `x` is a multiple of `y` using the expression `x%y == 0`. If true, it prints `ALIN KARISA HIZANNAH`; if false, it prints `109082500010`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var x, y int
7
8     fmt.Print("Masukkan dua bilangan (x y): ")
9     fmt.Scan(&x, &y)
10
11     fmt.Println(y%x == 0)
12
13     fmt.Println(x%y == 0)
14 }
15
```

The terminal window shows the output of the program for three different inputs:

```
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8> go run soal3.go
Masukkan dua bilangan (x y): 10 5
false
true
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8> go run soal3.go
Masukkan dua bilangan (x y): 3 21
true
false
PS D:\TEL-U\Semester 1\Algoritma Pemrograman\Praktek 8> go run soal3.go
Masukkan dua bilangan (x y): 4 4
true
true
```

## Deskripsi program

Program ini digunakan untuk mengecek hubungan kelipatan antara dua bilangan yang dimasukkan oleh pengguna. Setelah pengguna memasukkan dua angka, yaitu `x` dan `y`, program akan memeriksa apakah `y` merupakan kelipatan dari `x` dengan mengevaluasi ekspresi `y % x == 0`. Jika hasilnya `true`, berarti `y` habis dibagi `x`; jika `false`, berarti tidak.

Setelah itu, program melakukan pengecekan sebaliknya, yaitu apakah `x` merupakan kelipatan dari `y` dengan mengevaluasi `x % y == 0`. Kedua hasil pemeriksaan langsung ditampilkan secara berurutan. Dengan cara ini, pengguna dapat mengetahui hubungan pembagian antara dua bilangan dengan cepat dan sederhana tanpa penjelasan tambahan.