

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 9**

**IF-THEN**



**Disusun oleh:**

**M MAHDAN ARGYA SYARIF**

**109082500059**

**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 x int

    fmt.Scan(&x)

    if x < 0{

        x *= -1

    }

    fmt.Print(x)

}
```

#### Screenshoot program

```
1 package main
2 import "fmt"
3
4 func main(){ main redeclared in this block (see details)
5     var x int
6     fmt.Scan(&x)
7
8     if x < 0{
9         x *= -1
10    }
11    fmt.Print(x)
12 }
```

Ln 3, Col 16 | 61 character | Plain text | 120% | Window

PROBLEMS 10 OUTPUT TERMINAL DEBUG CONSOLE PORTS

```
• PS C:\GoLang> go run guide1.go
10
10
• PS C:\GoLang> go run guide1.go
-3
3
• PS C:\GoLang> go run guide1.go
5
5
• PS C:\GoLang> go run guide1.go
0
0
• PS C:\GoLang> go run guide1.go
-9999
9999
```

### Deskripsi program

Program ini dirancang untuk memastikan outputnya selalu bilangan positif, tidak peduli apakah inputnya positif atau negatif. Setelah pengguna memasukkan sebuah bilangan integer (x) menggunakan `fmt.Scan`, program tersebut menjalankan satu algoritma kondisional inti: `if n < 0 { x *= -1 }`. Logika ini secara spesifik memeriksa apakah angka yang dimasukkan (x) bernilai negatif; jika ya, angka tersebut akan dikalikan dengan -1 untuk mengubahnya menjadi positif. Jika angka yang dimasukkan sudah positif atau nol, kondisi `if` tidak terpenuhi, dan angka tersebut dicetak apa adanya.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var x int

    fmt.Scan(&x)

    if x > 0{

        fmt.Print("positif")

    } else {

        fmt.Print("bukan positif")

    }

}
```

## Screenshoot program

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

PS C:\GoLang> go run guide2.go  
10  
positif  
PS C:\GoLang> go run guide2.go  
-3  
bukan positif  
PS C:\GoLang> go run guide2.go  
5  
positif  
PS C:\GoLang> go run guide2.go  
0  
bukan positif

## Deskripsi program

Program ini dirancang untuk mengklasifikasikan sebuah angka masukan (n) sebagai "positif" atau "bukan positif" menggunakan struktur if-else. Inti algoritmanya terletak pada kondisi if  $n > 0$ , yang berarti hanya jika nilai yang diinput pengguna tepat lebih besar dari 0, program akan mencetak "positif". Untuk kasus lainnya, termasuk jika pengguna memasukkan 0 atau angka negatif, program akan menjalankan bagian else dan mencetak "bukan positif". Seperti yang Anda catat, 0 secara spesifik tidak dianggap positif dalam logika ini karena kondisi yang digunakan adalah  $>$  (lebih besar dari), bukan  $\geq$  (lebih besar atau sama dengan).

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var x int

    var t bool
```

```

    fmt.Scan(&x)

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

        t = true

    }

    fmt.Print(t)

}

```

### Screenshoot program

The screenshot shows a Go program in a code editor and its execution in a terminal. The code defines a function `main` that reads an integer `x` and a boolean `t`. It checks if `x` is both negative and even using the logical AND operator `&&`. If the condition is true, `t` is set to `true`, and the result is printed using `fmt.Print(t)`.

The terminal shows the program being run multiple times with different inputs. For negative odd numbers (-3, -4), the output is `false`. For a negative even number (-2), the output is `true`. For non-negative numbers (0, 10), the output is `false`.

```

1 package main
2 import "fmt"
3
4 func main() {
5     var x int
6     var t bool
7
8     fmt.Scan(&x)
9
10    if x < 0 && x % 2 == 0 {
11        t = true
12    }
13    fmt.Print(t)
14 }

```

```

PS C:\GoLang> go run guide3.go
10
false
PS C:\GoLang> go run guide3.go
-3
false
PS C:\GoLang> go run guide3.go
-4
true
PS C:\GoLang> go run guide3.go
0
false
PS C:\GoLang> go run guide3.go
-2
true

```

### Deskripsi program

Program ini dirancang khusus untuk bernilai `true` hanya jika angka yang diinput pengguna adalah bilangan genap negatif. Logikanya menggunakan operator `&&` (DAN), yang mengharuskan kedua kondisi terpenuhi secara bersamaan: pertama, angka tersebut harus negatif ( $n < 0$ ), dan kedua, angka tersebut harus genap ( $n \% 2 == 0$ ). Inilah sebabnya mengapa jika pengguna memasukkan bilangan bulat positif, meskipun angkanya genap, hasilnya akan tetap `false`; karena syarat pertama (negatif) tidak terpenuhi, sehingga operator `&&` secara otomatis gagal.

## TUGAS

### 1. Tugas 1

#### Source code

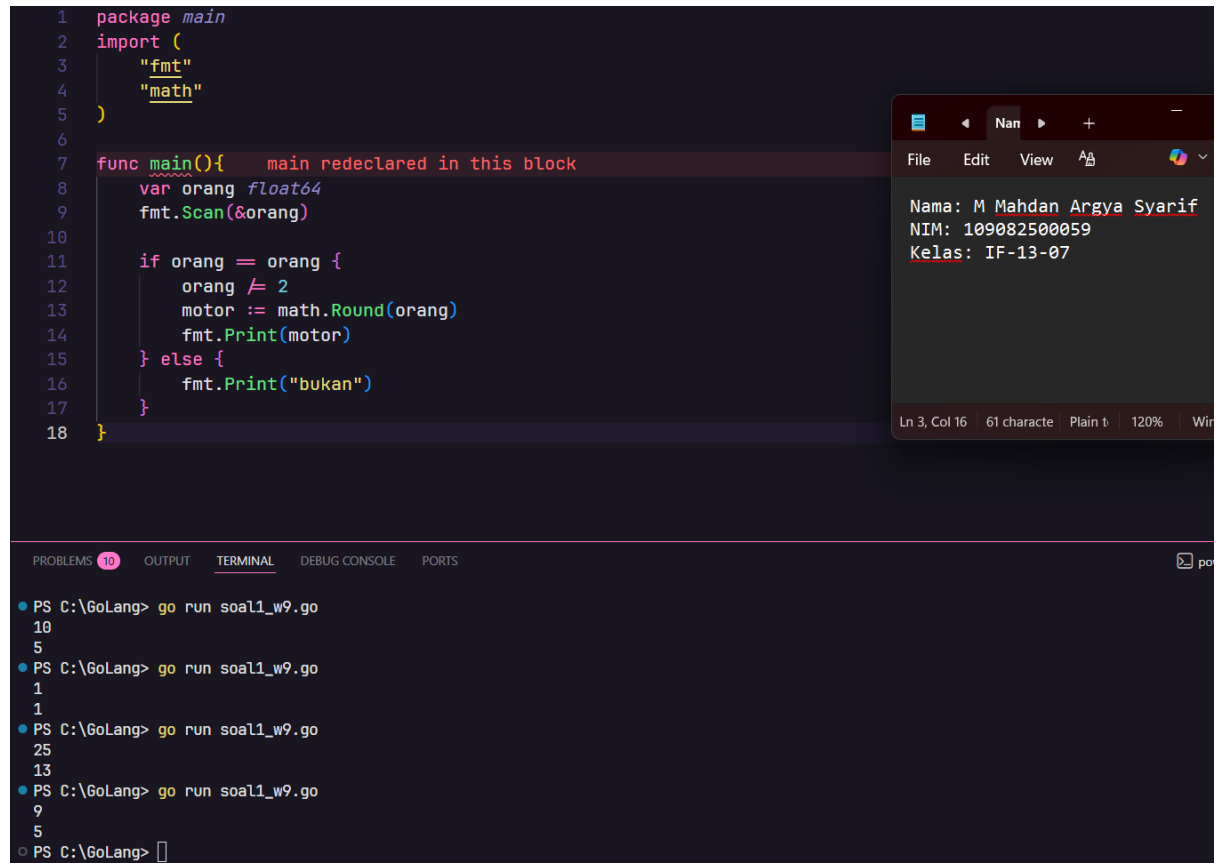
```
package main

import (
    "fmt"
    "math"
)

func main() {
    var orang float64
    fmt.Scan(&orang)

    if orang == orang {
        orang /= 2
        motor := math.Round(orang)
        fmt.Print(motor)
    } else {
        fmt.Print("bukan")
    }
}
```

## Screenshoot program



```
1 package main
2 import (
3     "fmt"
4     "math"
5 )
6
7 func main(){  main redeclared in this block
8     var orang float64
9     fmt.Scan(&orang)
10
11     if orang == orang {
12         orang /= 2
13         motor := math.Round(orang)
14         fmt.Print(motor)
15     } else {
16         fmt.Print("bukan")
17     }
18 }
```

File Edit View

Nama: M Mahdan Argya Syarif  
NIM: 109082500059  
Kelas: IF-13-07

Ln 3, Col 16 61 character Plain t 120% Win

PROBLEMS 10 OUTPUT TERMINAL DEBUG CONSOLE PORTS

```
PS C:\GoLang> go run soal1_w9.go
10
5
PS C:\GoLang> go run soal1_w9.go
1
1
PS C:\GoLang> go run soal1_w9.go
25
13
PS C:\GoLang> go run soal1_w9.go
9
5
PS C:\GoLang>
```

## Deskripsi program

Program ini dirancang untuk menghitung jumlah minimum motor yang diperlukan untuk orang yang diinput, di mana 1 motor menampung maksimal 2 orang. Inti algoritmanya adalah mengatasi sisa pembagian: karena 9 orang membutuhkan 5 motor (bukan 4 atau 4.5), program ini menggunakan tipe data float untuk perhitungan ( $\text{jumlahOrang} / 2.0$ ). Setelah mendapatkan hasil desimal (seperti 4.5), program ini kemudian menggunakan fungsi dari *library* math untuk membulatkan hasilnya ke atas (seperti `math.Ceil` atau `math.Round` untuk kasus  $x.5$ ), memastikan bahwa setiap sisa orang tetap dihitung mendapatkan satu motor penuh.

## 2. Tugas 2

### Source code

```
package main

import "fmt"

func main() {

    var x int

    fmt.Scan(&x)

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

        fmt.Print("genap negatif")

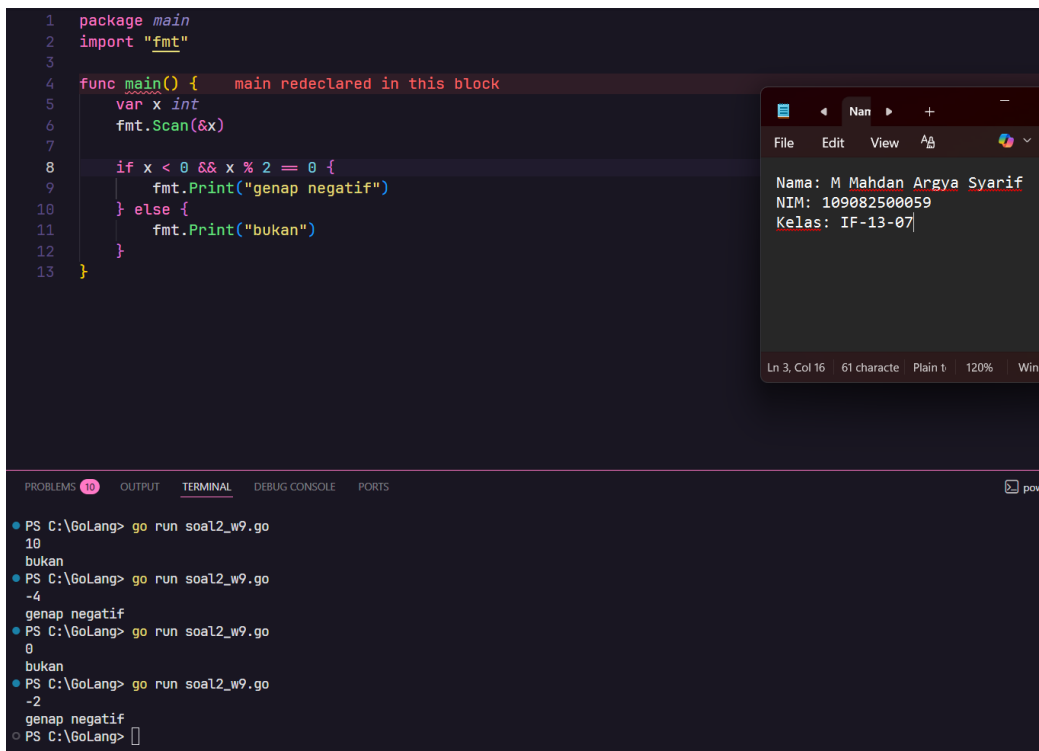
    } else {

        fmt.Print("bukan")

    }

}
```

### Screenshoot program



The screenshot shows a Go program in a code editor and its execution in a terminal. The code defines a `main` function that reads an integer `x` and checks if it is a negative even number. If so, it prints "genap negatif"; otherwise, it prints "bukan".

The terminal shows three test cases:

- Input: 10, Output: bukan
- Input: -4, Output: genap negatif
- Input: 0, Output: bukan

The code editor shows the following code with line numbers 1 through 13:

```
1 package main
2 import "fmt"
3
4 func main() {
5     var x int
6     fmt.Scan(&x)
7
8     if x < 0 && x % 2 == 0 {
9         fmt.Print("genap negatif")
10    } else {
11        fmt.Print("bukan")
12    }
13 }
```



### Deskripsi program

Program ini pada intinya dirancang untuk memvalidasi apakah sebuah angka masukan (n) adalah bilangan genap negatif. Algoritma utamanya menggunakan satu pernyataan if dengan operator logika && (DAN), yaitu `if n < 0 && n % 2 == 0`. Ini berarti sebuah angka hanya akan dianggap "benar" atau lolos jika memenuhi kedua syarat sekaligus: pertama, `n < 0` (angka itu harus negatif), dan kedua, `n % 2 == 0` (angka itu harus genap). Seperti contoh Anda, nilai -3 akan gagal karena meskipun memenuhi syarat pertama (negatif), ia tidak memenuhi syarat kedua (ganjil), sehingga operator && akan menghasilkan false.

### 3. Tugas 3

#### Source code

```
package main
import "fmt"

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

    if y == y {
        t = y % x == 0
        t = bool(t)
    }
    fmt.Println(t)

    if x == x {
        t = x % y == 0
        t = bool(t)
    }
    fmt.Println(t)
}
```

## Screenshoot program

```
1 package main
2 import "fmt"
3
4 func main() {  main redeclared in this block
5     var x, y int
6     var t bool
7     fmt.Scan(&x, &y)
8
9     if y == y {
10         t = y % x == 0
11         t = bool(t)
12     }
13     fmt.Println(t)
14
15     if x == x {
16         t = x % y == 0
17         t = bool(t)
18     }
19     fmt.Println(t)
20 }
```

Ln 3, Col 16 | 61 character | Plain text | 120% | Window

PROBLEMS 10 | OUTPUT | TERMINAL | DEBUG CONSOLE | PORTS

```
PS C:\GoLang> go run soa13_w9.go
10 5
false
true
PS C:\GoLang> go run soa13_w9.go
3 21
true
false
PS C:\GoLang> go run soa13_w9.go
4 4
true
true
```

## Deskripsi program

Program ini adalah alat pengecek faktor sederhana untuk dua angka masukan (x dan y) yang mengandalkan operator modulo (%) untuk menguji sisa bagi. Inti logikanya adalah melakukan dua evaluasi boolean yang terpisah: pertama, ia memeriksa (y % x == 0) untuk melihat apakah x adalah faktor dari y, dan kedua, ia memeriksa (x % y == 0) untuk melihat apakah y adalah faktor dari x. Hasil akhir program, seperti "false & true", hanyalah output gabungan dari kedua tes independen ini, yang sepenuhnya ditentukan oleh nilai yang dimasukkan pengguna.