

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

MODUL 9

IF-THEN



Disusun oleh:

Yedija Johanan Siregar

109082500075

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 {

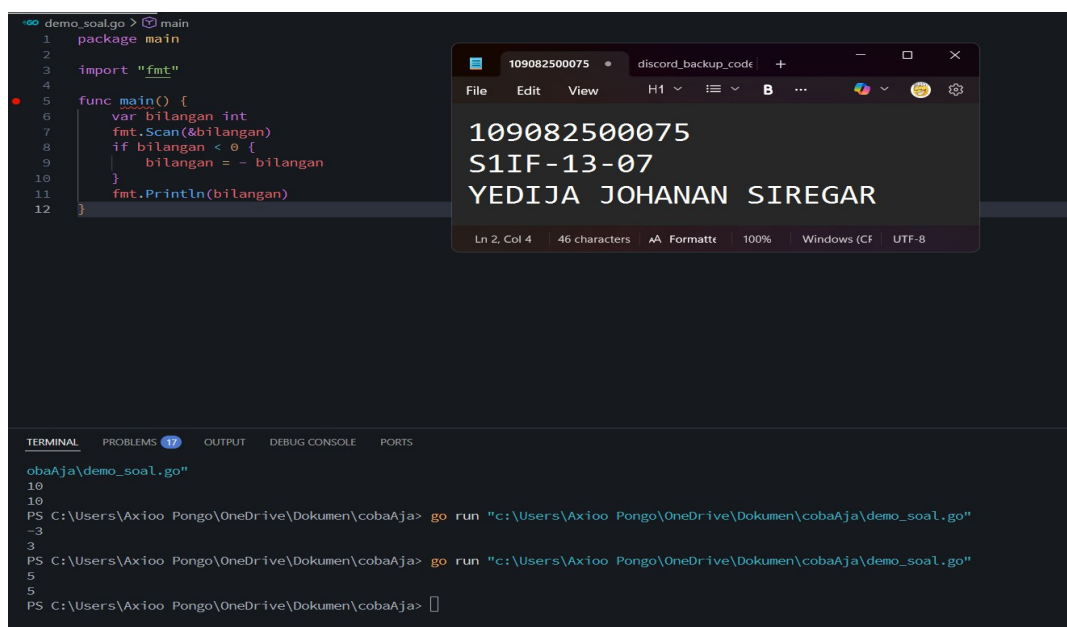
        bilangan = - bilangan

    }

    fmt.Println(bilangan)

}
```

Screenshoot program



Deskripsi program

Program ini berfungsi untuk menampilkan **nilai absolut** dari sebuah bilangan bulat yang dimasukkan oleh pengguna. Saat program dijalankan, pengguna diminta memasukkan satu nilai integer melalui input standar. Setelah nilai diterima, program melakukan pengecekan menggunakan percabangan if. Jika bilangan tersebut bernilai negatif, maka program mengubahnya menjadi positif dengan mengalikan angka tersebut dengan -1. Namun, jika bilangan yang dimasukkan sudah bernilai positif, program langsung mencetaknya tanpa perubahan. Dengan demikian, output yang dihasilkan selalu berupa versi positif dari bilangan yang diberikan pengguna.

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 {

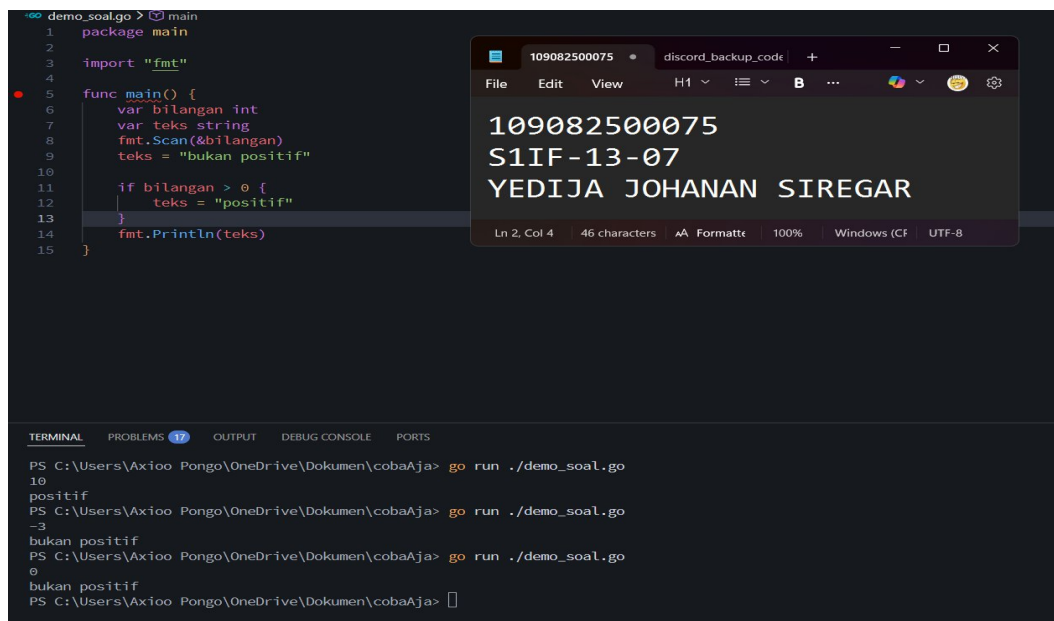
        teks = "positif"

    }

    fmt.Println(teks)

}
```

Screenshoot program



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 `bilangan` and a string `teks`. It checks if `bilangan` is greater than 0; if so, it sets `teks` to "positif", otherwise it remains "bukan positif". The program then prints `teks`. The terminal shows the program being run three times with inputs 10, -3, and 0, resulting in outputs "positif", "bukan positif", and "bukan positif" respectively. An inset window shows the formatted output of the first run: "109082500075", "S1IF-13-07", and "YEDIJA JOHANAN SIREGAR".

```
demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     var teks string
8     fmt.Scan(&bilangan)
9     teks = "bukan positif"
10
11     if bilangan > 0 {
12         teks = "positif"
13     }
14     fmt.Println(teks)
15 }
```

```
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
10
positif
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
-3
bukan positif
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
0
bukan positif
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>
```

109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR

Deskripsi program

Program ini digunakan untuk menentukan apakah sebuah bilangan yang dimasukkan pengguna bernilai **positif** atau **bukan positif**. Ketika program dijalankan, pengguna diminta memasukkan sebuah angka bertipe integer. Setelah nilai diterima, program terlebih dahulu mengisi variabel teks dengan nilai awal "**bukan positif**" sebagai anggapan default. Kemudian dilakukan pengecekan menggunakan percabangan if: apabila bilangan yang dimasukkan lebih besar dari 0, maka nilai teks diubah menjadi "**positif**". Pada akhir program, nilai teks yang telah disesuaikan akan dicetak ke layar, sehingga pengguna dapat mengetahui apakah bilangan tersebut tergolong positif atau tidak.

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

The screenshot shows a Go program being executed in a terminal. The program's logic is to check if an input number is both even and negative. The terminal output shows three test cases: input 10 results in 'false', input -3 results in 'false', and input 0 results in 'false'. In the background, a Discord backup code window is visible, displaying the code '109082500075', the identifier 'S1IF-13-07', and the name 'YEDIJA JOHANAN SIREGAR'.

```

demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan int
7     var hasil bool
8     fmt.Scan(&bilangan)
9
10    hasil = bilangan % 2 == 0 && bilangan < 0
11    fmt.Println(hasil)
12 }

```

109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR

Ln 2, Col 4 46 characters AA Formatte 100% Windows (CF UTF-8

TERMINAL PROBLEMS 17 OUTPUT DEBUG CONSOLE PORTS

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
10
false
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
-3
false
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
0
false
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

Deskripsi program

Program ini memeriksa apakah bilangan yang dimasukkan pengguna merupakan **bilangan genap** dan sekaligus **negatif**. Setelah membaca input, program mengevaluasi dua kondisi: bilangan habis dibagi 2 dan nilainya kurang dari 0. Jika

kedua syarat terpenuhi, variabel hasil bernilai true; jika tidak, nilainya false. Hasil pemeriksaan tersebut kemudian dicetak ke layar.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var n int

    fmt.Print("Masukkan bilangan bulat positif n: ")

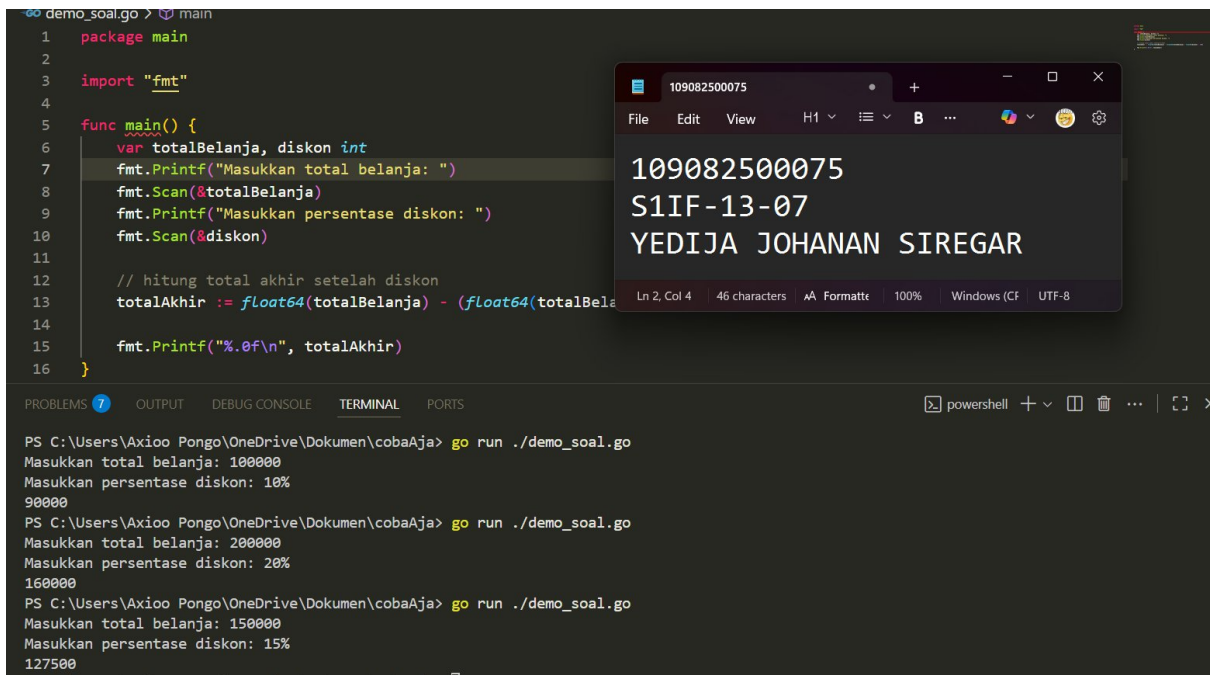
    fmt.Scan(&n)

    jumlah := 0

    for i := 1; i <= n; i++ {
        jumlah += i
    }

    fmt.Println("Hasil penjumlahan dari 1 sampai", n, "adalah",
jumlah)
}
```

Screenshoot program



The screenshot shows a Go IDE with a code editor on the left and a terminal on the right. The code editor displays the following Go code:

```
demo_soal.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var totalBelanja, diskon int
7     fmt.Printf("Masukkan total belanja: ")
8     fmt.Scan(&totalBelanja)
9     fmt.Printf("Masukkan persentase diskon: ")
10    fmt.Scan(&diskon)
11
12    // hitung total akhir setelah diskon
13    totalAkhir := float64(totalBelanja) - (float64(totalBelanja) * float64(diskon) / 100)
14
15    fmt.Printf("%.0f\n", totalAkhir)
16 }
```

The terminal shows the execution of the program with three test cases:

```
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan total belanja: 100000
Masukkan persentase diskon: 10%
90000
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan total belanja: 200000
Masukkan persentase diskon: 20%
160000
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan total belanja: 150000
Masukkan persentase diskon: 15%
127500
```

Deskripsi program

Program ini digunakan untuk menghitung hasil penjumlahan dari bilangan 1 sampai dengan n. Ketika dijalankan, program akan meminta pengguna memasukkan sebuah bilangan bulat positif dan menyimpannya ke dalam variabel n. Selanjutnya, program menginisialisasi variabel jumlah dengan nilai 0, kemudian menggunakan perulangan for untuk menambahkan setiap bilangan dari 1 hingga n ke dalam variabel tersebut. Setelah proses perulangan selesai, program menampilkan hasil akhir penjumlahan di layar sebagai keluaran. Dengan cara ini, pengguna dapat mengetahui total penjumlahan dari deret bilangan 1 hingga n sesuai nilai yang dimasukkan.

2. Tugas 2

Source code

```
package main

import (
    "fmt"
    "math"
```

```

)

func main() {

    var n int

    fmt.Print("Masukkan jumlah kerucut: ")

    fmt.Scan(&n)

    fmt.Println("Masukkan jari-jari dan tinggi setiap
kerucut:")

    for i := 1; i <= n; i++ {

        var r, t float64

        fmt.Printf("Kerucut ke-%d (r t): ", i)

        fmt.Scan(&r, &t)

        volume := (1.0 / 3.0) * math.Pi * r * r * t

        fmt.Printf("Volume kerucut ke-%d = %.14f\n", i,
volume)

    }

}

```

Screenshoot program


```
demo_soal.go > main
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var n int
10    fmt.Print("Masukkan jumlah kerucut: ")
11    fmt.Scan(&n)
12
13    fmt.Println("Masukkan jari-jari dan tinggi setiap kerucut:")
14
15    for i := 1; i <= n; i++ {
16        var r, t float64
17        fmt.Printf("Kerucut ke-%d (r t): ", i)
18        fmt.Scan(&r, &t)
19
20        volume := (1.0 / 3.0) * math.Pi * r * r * t
21        fmt.Printf("Volume kerucut ke-%d = %.14f\n", i, volume)
22    }
23 }
```

109082500075
S1IF-13-07
YEDIJA JOHANAN SIREGAR

Ln 2, Col 4 46 characters xA Formatt 100% Windows (CF UTF-8

PROBLEMS OUTPUT DEBUG CONSOLE PORTS

TERMINAL

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan jumlah kerucut: 1
Masukkan jari-jari dan tinggi setiap kerucut:
Kerucut ke-1 (r t): 3
4
Volume kerucut ke-1 = 37.6991184387752
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

Deskripsi program

Program ini digunakan untuk menghitung volume beberapa kerucut berdasarkan jari-jari dan tinggi yang dimasukkan pengguna. Pertama, pengguna diminta memasukkan jumlah kerucut (n), kemudian memasukkan pasangan nilai jari-jari (r) dan tinggi (t) sebanyak n kali. Setiap kali nilai dimasukkan, program menghitung volume menggunakan rumus, $V = \frac{1}{3}\pi r^2 t$, lalu menampilkan hasilnya di layar untuk setiap kerucut.

3. Tugas 3

Source code

```
package main

import "fmt"

func main() {
    var a, b int
    fmt.Print("Masukkan dua bilangan bulat positif (a b): ")
    fmt.Scan(&a, &b)

    hasil := 1
    for i := 1; i <= b; i++ {
```

```

        hasil *= a
    }

    fmt.Println("Hasil pemangkatan:", hasil)
}

```

Screenshoot program

```

demo_soal.go > main
1  package main
2
3  import "fmt"
4
5  func main() {
6      var a, b int
7      fmt.Print("Masukkan dua bilangan bulat positif (a b): ")
8      fmt.Scan(&a, &b)
9
10     hasil := 1
11     for i := 1; i <= b; i++ {
12         hasil *= a
13     }
14
15     fmt.Println("Hasil pemangkatan:", hasil)
16 }

```

109082500075
SIF-13-07
YEDIJA JOHANAN SIREGAR

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan dua bilangan bulat positif (a b): 4 2
Hasil pemangkatan: 16
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan dua bilangan bulat positif (a b): 2 10
Hasil pemangkatan: 1024
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan dua bilangan bulat positif (a b): 10 3
Hasil pemangkatan: 1000
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

Deskripsi program

Program ini digunakan untuk menghitung hasil pemangkatan dari dua bilangan bulat positif dengan menggunakan operasi perkalian dan perulangan. Ketika dijalankan, pengguna diminta memasukkan dua bilangan, di mana bilangan pertama (a) merupakan angka yang akan dipangkatkan, dan bilangan kedua (b) adalah pangkatnya. Program kemudian melakukan perkalian berulang sebanyak b kali menggunakan struktur perulangan for. Setiap iterasi mengalikan nilai hasil dengan a hingga perulangan selesai. Setelah itu, program menampilkan hasil akhir pemangkatan ke layar.

4. Tugas 4

Source code

```
package main
```

```

import "fmt"

func main() {
    var n int

    fmt.Print("Masukkan bilangan bulat: ")
    fmt.Scan(&n)

    faktorial := 1
    for i := 1; i <= n; i++ {
        faktorial *= i
    }

    fmt.Println("Hasil faktorial dari", n, "adalah",
faktorial)
}

```

Screenshoot program

The screenshot shows a Go IDE with the following code in `demo_soal.go`:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7     fmt.Print("Masukkan bilangan bulat: ")
8     fmt.Scan(&n)
9
10    faktorial := 1
11    for i := 1; i <= n; i++ {
12        faktorial *= i
13    }
14
15    fmt.Println("Hasil faktorial dari", n, "adalah", faktorial)
16 }

```

The terminal output shows the program being run three times with different inputs:

```

PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan bilangan bulat: 0
Hasil faktorial dari 0 adalah 1
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan bilangan bulat: 1
Hasil faktorial dari 1 adalah 1
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja> go run ./demo_soal.go
Masukkan bilangan bulat: 10
Hasil faktorial dari 10 adalah 3628800
PS C:\Users\Axioo Pongo\OneDrive\Dokumen\cobaAja>

```

Deskripsi program

Program ini digunakan untuk menghitung hasil faktorial dari suatu bilangan bulat non-negatif. Ketika dijalankan, program akan meminta pengguna memasukkan

sebuah bilangan n . Kemudian program menginisialisasi variabel faktorial dengan nilai 1, karena nilai awal faktorial selalu dimulai dari 1. Setelah itu, program menggunakan perulangan `for` dari 1 sampai n untuk mengalikan setiap bilangan secara berurutan dengan variabel faktorial. Setelah perulangan selesai, hasil akhir perhitungan ditampilkan di layar sebagai nilai faktorial dari n .