

**LAPORAN PRAKTIKUM ALGORITMA
DAN PEMROGRAMAN 1**

**MODUL [No. MODUL]
[NAMA MODUL]**



Disusun oleh:

HAFIZD SAMA'I SYAMSI

109082500183

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 detik, jam, menit int
    fmt.Scan(&detik)
    jam = detik / 3600
    menit = (detik % 3600) / 60
    detik = detik % 60
    fmt.Println(jam, "jam", menit, "menit dan", detik,
        "detik")
}
```

Screenshoot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)
CONTOH TANGKAPAN LAYAR:

```
Welcome modul4.go 1 guided1.go 3 X guided2.go 1 guide
guided1.go > main
1 package main
2 import "fmt"
3 func main() {
4     var detik, jam, menit int
5     fmt.Scan(&detik)
6     jam = detik / 3600
7     menit = (detik % 3600) / 60
8     detik = detik % 60
9     fmt.Println(jam, "jam", menit, "menit dan", detik, "detik")
10 }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\go\praktek4> go run soal1.go
# command-line-arguments
runtime.main_main·f: function main is undeclared in the main package
PS D:\go\praktek4> go run soal1.go
3661
1 jam 1 menit dan 1 detik
PS D:\go\praktek4> go run guided2.go
362
false
PS D:\go\praktek4> 
```

Deskripsi program

membaca input berupa total detik, lalu menghitung berapa jam dengan membagi 3600, menghitung menit dari sisa detik yang tersisa, dan akhirnya menghitung sisa detik terakhir. Hasilnya kemudian ditampilkan dalam format "x jam x menit dan x detik"

2. Guided 2

Source Code

```
package main

import "fmt"

func main() {

    var bilangan, d1, d2, d3 int

    fmt.Scan(&bilangan)

    d1 = bilangan / 100
```

```
        d2 = bilangan % 100 / 10  
        d3 = bilangan % 100 % 10  
        fmt.Println(d1 <= d2 && d2 <= d3)  
  
    }
```

Screenshoot program

```
guided2.go > main
1  package main
2  import "fmt"
3  func main() {
4      var bilangan, d1, d2, d3 int
5      fmt.Scan(&bilangan)
6      d1 = bilangan / 100
7      d2 = bilangan % 100 / 10
8      d3 = bilangan % 100 % 10
9      fmt.Println(d1 <= d2 && d2 <= d3)
10 }
```

PROBLEMS 6 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\go\praktek4> go run guided2.go
fork/exec C:\Users\LENOVO\AppData\Local\go-build\71\714f2476ed5b\
id not complete successfully because the file contains a virus o
PS D:\go\praktek4> go run guided2.go
fork/exec C:\Users\LENOVO\AppData\Local\Temp\go-build1640207383\
contains a virus or potentially unwanted software.
go: unlinkat C:\Users\LENOVO\AppData\Local\Temp\go-build16402073
le contains a virus or potentially unwanted software.
PS D:\go\praktek4> go run guided2.go
362
false
PS D:\go\praktek4> go run guided2.go
256
true
PS D:\go\praktek4> go run guided2.go
189
true
PS D:\go\praktek4> 
```

Deskripsi program

Memeriksa apakah angka-angka dalam sebuah bilangan tiga digit tersusun secara berurutan dari kecil ke besar. Cukup masukan bilangan tiga digit, lalu program akan memisahkan setiap digitnya dan mengecek urutannya jika urutannya dari kecil ke besar adalah true, kalau besar ke kecil adalah false

Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var beratBadan, tinggiBadan, bmi float64

    fmt.Scan(&beratBadan, &tinggiBadan)

    bmi = beratBadan / (tinggiBadan * tinggiBadan)

    fmt.Printf("%.2f", bmi)

}
```

Screenshoot program

```
guided2.go 1 x  ~GO soal1.go 1  ~GO soal3.go 1  ~GO soal5.go 1  ~GO jaribola.go  ~GO  
~GO guided3.go > main  
7 1 package main  
1 2 import "fmt"  
1 3 func main() {  
1 4  
1 5     var beratBadan, tinggiBadan, bmi float64  
1 6     fmt.Scan(&beratBadan, &tinggiBadan)  
1 7     bmi = beratBadan / (tinggiBadan * tinggiBadan)  
1 8     fmt.Printf("%.2f", bmi)  
1 9 }
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\go\praktek4> go run guided3.go  
70. 1.75  
22.86  
PS D:\go\praktek4> go run guided3.go  
60 1.6  
23.44  
PS D:\go\praktek4> go run guided3.go  
80 1.8  
24.69  
PS D:\go\praktek4>
```

Deskripsi program

Menghitung indeks BMI berdasarkan berat dan tinggi badan yang dimasukan. Ketik berat badan dalam kilogram dan tinggi badan dalam meter, kemudian memposes data dalam dan menampilkan hasil perhitungan BMI dengan dua angka dibelakang koma, hasil mengetahui apakah berat badan tergolong ideal, kurang atau berlebih

TUGAS

1. Tugas 1

Source code

```
package main  
  
import "fmt"  
  
func main() {
```

```

var totalBelanja, diskon int
var totalAkhir float64

fmt.Scan(&totalBelanja)
fmt.Scan(&diskon)

totalAkhir = float64(totalBelanja) * (1 - float64(diskon)/100)

fmt.Printf ("Total belanja akhir: %.2f\n", totalAkhir)

}

```

Screenshoot program

The screenshot shows a Go IDE with the following tabs: Welcome, modul4.go 1, guided1.go 4, guided2.go 1, soal1.go 1, and soal1.go 2. The editor displays the source code for 'soal1.go' with line numbers 1 through 15. The code defines variables for totalBelanja, diskon, and totalAkhir, reads input from the user, calculates the total after a discount, and prints the result.

```

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

```

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

```

PS D:\go\praktek4> go run soal1.go
100000 10
Total belanja akhir: 90000.00
PS D:\go\praktek4> go run soal1.go
200000 20
Total belanja akhir: 160000.00
PS D:\go\praktek4> go run soal1.go
150000 15
Total belanja akhir: 127500.00
PS D:\go\praktek4>

```


Deskripsi program

menghitung total belanja akhir setelah mendapatkan diskon. cukup memasukkan dua nilai: total belanja dan persentase diskon. Program kemudian menghitung potongan harga berdasarkan diskon tersebut, lalu menampilkan hasil akhir yang harus dibayar dengan dua angka di belakang koma. program ini membantu mengetahui berapa jumlah uang yang perlu dibayar setelah harga dikurangi diskon.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var bmi, tinggiBadan, beratBadan float64

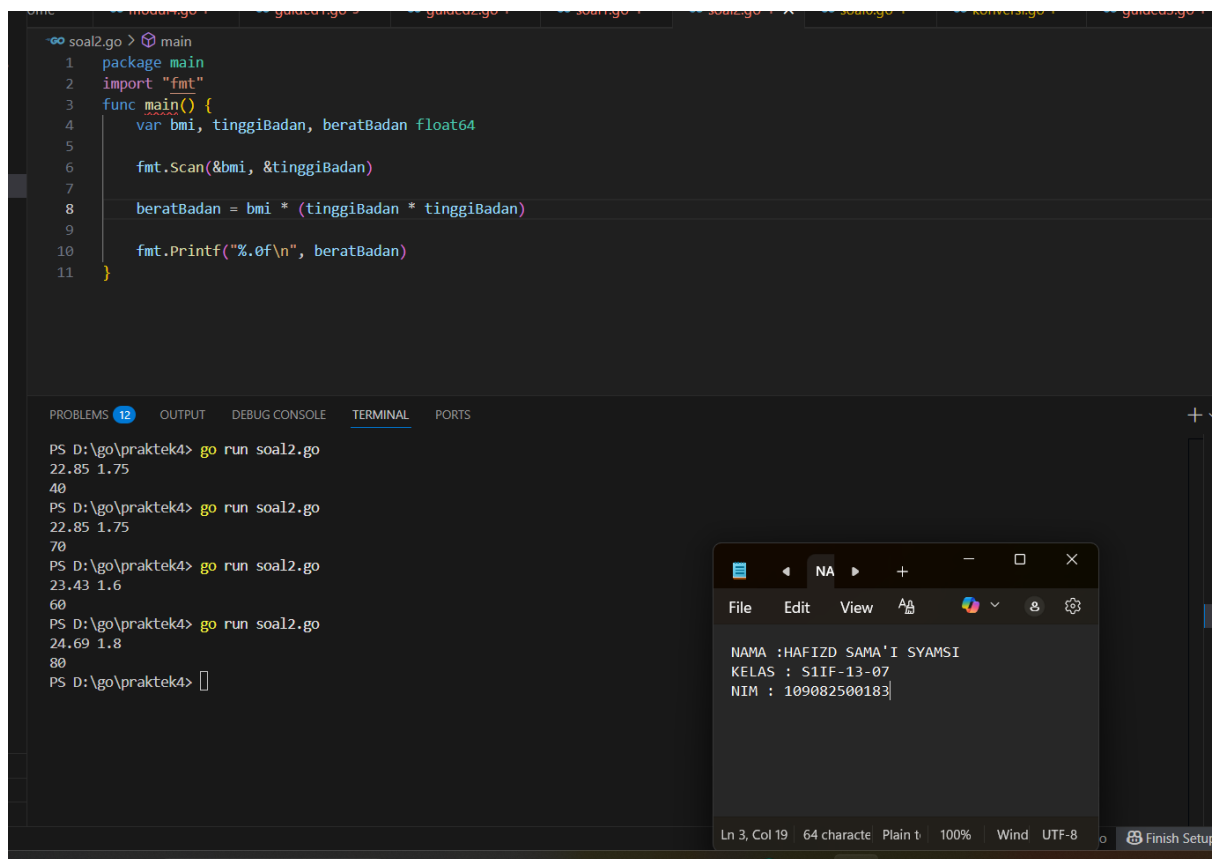
    fmt.Scan(&bmi, &tinggiBadan)

    beratBadan = bmi * (tinggiBadan * tinggiBadan)

    fmt.Printf("%.0f\n", beratBadan)

}
```

Screenshoot program



```
soal2.go > main
1 package main
2 import "fmt"
3 func main() {
4     var bmi, tinggiBadan, beratBadan float64
5
6     fmt.Scan(&bmi, &tinggiBadan)
7
8     beratBadan = bmi * (tinggiBadan * tinggiBadan)
9
10    fmt.Printf("%.0f\n", beratBadan)
11 }
```

```
PS D:\go\praktek4> go run soal2.go
22.85 1.75
40
PS D:\go\praktek4> go run soal2.go
22.85 1.75
70
PS D:\go\praktek4> go run soal2.go
23.43 1.6
60
PS D:\go\praktek4> go run soal2.go
24.69 1.8
80
PS D:\go\praktek4>
```

NAMA : HAFIZD SAMA'I SYAMSI
KELAS : SIIF-13-07
NIM : 109082500183

Deskripsi program

memperkirakan **berat badan seseorang berdasarkan nilai BMI dan tinggi badan yang dimasukkan.**

Ketika dijalankan, program meminta dua input: nilai BMI (indeks massa tubuh) dan tinggi badan dalam meter. Setelah data dimasukkan, program memproses keduanya untuk menentukan berat badan yang sesuai dengan nilai BMI tersebut. Hasil akhirnya ditampilkan dalam bentuk angka bulat tanpa desimal, sehingga bisa langsung mengetahui estimasi berat badannya dengan tampilan yang sederhana

Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var x1, y1, x2, y2, x3, y3 float64
    fmt.Scan(&x1, &y1)
    fmt.Scan(&x2, &y2)
    fmt.Scan(&x3, &y3)
    ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
}
```

```

        bc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-
y2, 2))
        ca := math.Sqrt(math.Pow(x3-x1, 2) + math.Pow(y3-
y1, 2))
        max := ab
        if bc > max {
            max = bc
        }
        if ca > max {
            max = ca
        }
        fmt.Printf("sisi terpanjang = %.2f\n", max)
    }

```

Screenshoot program

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var x1, y1, x2, y2, x3, y3 float64
10    fmt.Scan(&x1, &y1)
11    fmt.Scan(&x2, &y2)
12    fmt.Scan(&x3, &y3)
13    ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
14    bc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
15    ca := math.Sqrt(math.Pow(x3-x1, 2) + math.Pow(y3-y1, 2))
16    max := ab
17    if bc > max {
18        max = bc
19    }
20    if ca > max {
21        max = ca
22    }
23    fmt.Printf("sisi terpanjang = %.2f\n", max)
24 }

```

```

PS D:\go\praktek4> go run soal3.go
1.0 1.0
4.0 1.0
2.0 5.0
sisi terpanjang = 4.47
PS D:\go\praktek4> go run soal3.go
0.0 0.0
3.0 0.0
3.0 4.0
sisi terpanjang = 5.00
PS D:\go\praktek4>

```

NAMA : HAFIZD SAMA'I SYAMSI
KELAS : S1IF-13-07
NIM : 109082500183

Deskripsi program

mencari **sisi terpanjang dari sebuah segitiga** berdasarkan tiga titik koordinat berikan tiga titik dengan nilai x dan y, program akan menghitung jarak antara setiap pasangan titik untuk mendapatkan panjang ketiga sisinya. Dari hasil perhitungan itu, program membandingkan semua sisi dan menampilkan sisi yang paling panjang dengan dua angka di belakang koma.