

LAPORAN PRAKTIKUM

Algoritma Pemrograman

MODUL [04]

[I/O, TIPE DATA & VARIABEL]



Disusun oleh:

RAYSA RAHMA IRAHIM

109082500167

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

TUGAS PENDAHULUAN

Pemberi soal : Rafli Nurhidayat-109082500152

Deskripsi soal :

1. Buatlah sebuah program dalam bahasa Go yang dapat menentukan digit ke-2 genap atau bukan. Program harus menerima satu masukan berupa bilangan bulat positif ratusan (100-999).

Contoh Input/Output :

Input	Output
154	False
865	True
987	True

Source Code

```
package main

import "fmt"

func main() {
    var angka int
    fmt.Print("Masukkan bilangan (100-999): ")
    fmt.Scan(&angka)

    puluhan := (angka / 10) % 10
    fmt.Println(puluhan%2 == 0)
}
```

Screenshot program

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The main area displays a Go file named `tugaspendahuluan1.go` with the following code:

```
package main
import "fmt"
func main() {
    var angka int
    fmt.Print("Masukkan bilangan (100-999): ")
    fmt.Scan(&angka)

    puluhan := (angka / 10) % 10
    fmt.Println(puluhan)
}
```

The terminal below shows the execution of the program:

```
Masukkan bilangan (100-999): 154
false
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugaspendahuluan1.go"
Masukkan bilangan (100-999): 865
true
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugaspendahuluan1.go"
Masukkan bilangan (100-999): 987
true
PS C:\Users\user\OneDrive\laprak4>
```

The status bar at the bottom right indicates the file is saved (green checkmark), the tab size is 4, and the date and time are 10/11/2025 2:09 PM.

Deskripsi program

- **package main**

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi `main()` sebagai titik awal berjalanannya program. Tanpa `package main`, program tidak bisa dijalankan sebagai aplikasi.

- **import "fmt"**

Berfungsi supaya program bisa membaca input dari keyboard menggunakan `Scan` dan menampilkan output ke layar menggunakan `Print` atau `Println`. Tanpa mengimpor `fmt`, perintah input-output seperti `fmt.Scan` atau `fmt.Println` tidak dapat digunakan. Paket `fmt` merupakan singkatan dari `format`, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa `main()`, program tidak akan bisa dijalankan.

- **Tanda buka dan tutup kurung kurawal { }**

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

- **var angka int**

var berfungsi sebagai wadah untuk membuat variabel baru yang menyimpan data. variabel angka digunakan untuk menampung bilangan yang dimasukkan melalui keyboard. Tipe data int dipilih karena input yang dimasukkan berupa bilangan bulat (integer).

- **fmt.Println("Masukkan bilangan (100-999): ")**

Berfungsi untuk membaca input dari keyboard dan menyimpannya ke dalam variabel angka. Tanda & di depan variabel berarti alamat dari variabel tersebut. Go membutuhkan alamat ini agar data yang diketik bisa langsung disimpan ke variabel yang dituju. Jika & tidak ada, maka program tidak tahu ke mana harus menaruh input.

- **puluhan := (angka / 10) % 10**

Baris ini digunakan untuk mengambil digit puluhan dari bilangan yang dimasukkan. angka / 10 membuang satu angka terakhir (misal 123 → 12) (angka / 10) % 10 mengambil sisa pembagian 10 dari hasil sebelumnya (12 % 10 → 2). Jadi, nilai puluhan akan berisi angka puluhan dari input. Operator := dipakai agar deklarasi dan pengisian nilai bisa dilakukan dalam satu baris.

- **fmt.Println(puluhan % 2 == 0)**

Berfungsi untuk mengecek apakah angka puluhan tersebut genap atau ganjil. puluhan % 2 artinya sisa pembagian puluhan dengan 2. == 0 artinya membandingkan apakah sisa itu nol (jika nol → genap). Hasilnya berupa nilai boolean, yaitu true jika genap dan false jika ganjil. fmt.Println digunakan supaya hasilnya tampil di layar dengan baris baru di akhir output.

2. Buatlah sebuah program dalam bahasa Go yang dapat menentukan apakah bilangan pertama merupakan kelipatan bilangan kedua atau bukan. Program harus menerima 2 masukan berupa bilangan bulat positif.

Contoh Input/Output :

Input	Output
100 10	True
50 25	True
33 5	False

Source Code

```
package main

import "fmt"

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

    fmt.Println(a%b == 0)
}
```

Screenshoot program

The screenshot shows the Visual Studio Code interface. In the top left, the Explorer sidebar displays a folder named 'LAPRAK4' containing several Go files: 'guide1.go', 'guide2.go', 'guide3.go', 'tugas1.go', 'tugas2.go', 'tugas3.go', 'tugaspendahuluan1.go', and 'tugaspendahuluan2.go'. The main editor area shows the content of 'tugaspendahuluan2.go'. The terminal at the bottom shows command-line interactions:

```
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugaspendahuluan2.go"
Masukkan dua bilangan: 100 10
true
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugaspendahuluan2.go"
Masukkan dua bilangan: 50 25
true
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugaspendahuluan2.go"
Masukkan dua bilangan: 33 5
false
PS C:\Users\user\OneDrive\laprak4>
```

The status bar at the bottom right indicates the file is '12.1.0' and the date is '10/11/2025'.

Deskripsi program

- **package main**

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalanannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

- **import "fmt"**

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau.Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

- **Tanda buka dan tutup kurung kurawal { }**

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

- **var a, b int**

Baris ini digunakan untuk mendeklarasikan dua variabel bernama a dan b yang bertipe integer (int). Variabel ini berfungsi sebagai wadah untuk menampung dua bilangan yang akan dimasukkan melalui keyboard. Dengan tipe data int, kedua variabel hanya bisa menyimpan bilangan bulat tanpa desimal.

- **fmt.Println("Masukkan dua bilangan: ")**

Digunakan untuk menampilkan perintah di layar, agar pengguna tahu bahwa mereka harus memasukkan dua bilangan. fmt.Println tidak membuat baris baru, sehingga input pengguna muncul di baris yang sama dengan teks perintah. Tanpa perintah ini, pengguna tidak akan tahu kapan harus mengetik angka.

- **fmt.Scan(&a, &b)**

Berfungsi untuk membaca dua input angka dari keyboard, kemudian menyimpannya ke dalam variabel a dan b. Tanda & berarti alamat dari variabel, agar data yang diketik pengguna bisa langsung masuk ke tempat penyimpanan variabel yang benar. Jika tanda & dihapus, maka program tidak akan tahu ke mana harus meletakkan input pengguna.

- **fmt.Println(a % b == 0)**

Baris ini berfungsi untuk mengecek apakah bilangan pertama (a) habis dibagi bilangan kedua (b). Operator % digunakan untuk mencari sisa pembagian (modulus). Jika hasil a % b sama dengan 0, artinya a habis dibagi b, maka hasilnya true. Jika tidak, hasilnya false. Hasil perbandingan ini langsung ditampilkan ke layar oleh fmt.Println.

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {
    var detik, jam, menit int
    fmt.Scan(&detik)

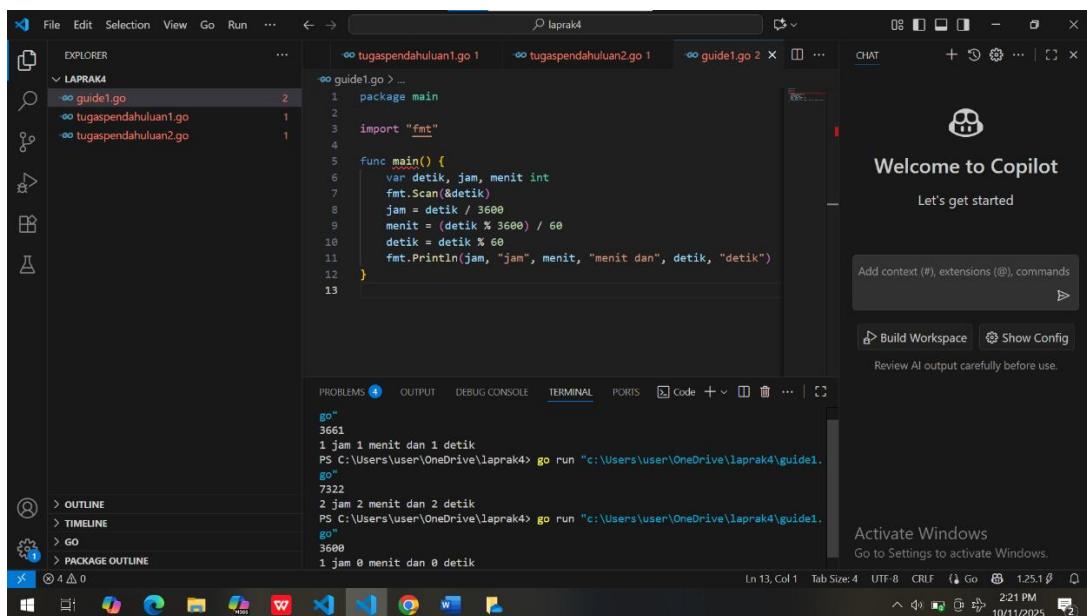
    jam = detik / 3600

    menit = (detik % 3600) / 60

    detik = detik % 60

    fmt.Println(jam, "jam", menit, "menit dan", detik,
    "detik")
}
```

Screenshot program



Deskripsi program

- **package main**

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalanannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

- **import "fmt"**

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

- **Tanda buka dan tutup kurung kurawal { }**

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

- **var detik, jam, menit int**

Berfungsi untuk mendeklarasikan tiga variabel bertipe int (bilangan bulat), yaitu detik, jam, dan menit. detik → digunakan untuk menyimpan jumlah detik yang dimasukkan pengguna. jam dan menit → digunakan untuk menyimpan hasil konversi dari satuan detik. Dengan tipe int, semua variabel ini hanya dapat menyimpan angka bulat tanpa desimal.

- **fmt.Scan(&detik)**

Berfungsi untuk membaca input dari keyboard dan menyimpannya ke variabel detik. Tanda & berarti alamat dari variabel tempat data akan disimpan. Tanpa tanda &, program tidak tahu di mana harus meletakkan hasil input pengguna. Jadi, ketika pengguna mengetik jumlah detik, nilainya akan langsung disimpan di variabel detik. $7 \text{ jam} = \text{detik} / 3600$ Baris ini menghitung jumlah jam dari total detik yang dimasukkan. Karena $1 \text{ jam} = 3600 \text{ detik}$, maka total detik dibagi 3600 untuk mendapatkan jumlah jam. Hasil pembagian otomatis dibulatkan ke bawah karena tipe datanya adalah int.

- **menit = (detik % 3600) / 60**

Digunakan untuk menghitung jumlah menit dari sisa detik yang belum menjadi jam. detik % 3600 menghitung sisa detik setelah jam diambil. hasilnya kemudian dibagi 60 untuk mengetahui berapa menit. Jadi, variabel menit berisi jumlah menit yang tersisa setelah dikonversi dari jam.

- **detik = detik % 60**

Setelah jam dan menit dihitung, baris ini digunakan untuk menghitung sisa detik terakhir yang tidak cukup menjadi 1 menit. Operator % (modulus) berfungsi untuk mengambil sisa pembagian. Jadi nilai detik sekarang menjadi detik yang tersisa setelah dikonversi ke jam dan menit.

- **fmt.Println(jam, "jam", menit, "menit dan", detik, "detik")**

Digunakan untuk menampilkan hasil konversi ke layar. Program akan menuliskan jumlah jam, menit, dan detik hasil perhitungan sebelumnya dalam format kalimat. Fungsi fmt.Println otomatis menambahkan baris baru setelah output, sehingga hasilnya rapi saat ditampilkan di terminal.

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

```
File Edit Selection View Go Run ... ← → 🔍 laprak4 ⌂ CHAT + ⚙️ ... x

EXPLORER LAPRAK4
guide1.go 3
guide2.go 1
tugaspendahuluan1.go 1
tugaspendahuluan2.go 1

guide2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan, d1, d2, d3 int
7     fmt.Scan(&bilangan)
8     d1 = bilangan / 100
9     d2 = bilangan % 100 / 10
10    d3 = bilangan % 100 % 10
11    fmt.Println(d1 <= d2 && d2 <= d3)
12 }
13

PROBLEMS 0 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + ⌂ ⌂ ... | ⌂
362
false
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\guide2.go"
256
true
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\guide2.go"
189
true
PS C:\Users\user\OneDrive\laprak4>

Activate Windows
Go to Settings to activate Windows.

In 13, Col 1 Tab Size: 4 UTF-8 CRLF ⌂ Go ⌂ 1.25.1 ⌂ 2:24 PM 10/11/2025 ⌂
```

Deskripsi program

- **package main**

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalanannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

- **import "fmt"**

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

- **Tanda buka dan tutup kurung kurawal { }**

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

- **var bilangan, d1, d2, d3 int**

Berfungsi untuk mendeklarasikan empat variabel bertipe int (bilangan bulat), yaitu: bilangan → digunakan untuk menyimpan input bilangan tiga digit dari pengguna. d1, d2, d3 → digunakan untuk menyimpan masing-masing digit ratusan, puluhan, dan satuan dari bilangan tersebut. Dengan tipe data int, keempat variabel ini hanya dapat menyimpan angka bulat tanpa desimal.

- **fmt.Scan(&bilangan)**

Digunakan untuk membaca input dari keyboard dan menyimpannya ke dalam variabel bilangan. Tanda & di depan nama variabel berarti alamat dari variabel tersebut. Tanpa tanda &, Go tidak tahu di mana harus menyimpan nilai yang dimasukkan oleh pengguna.

- **d1 = bilangan / 100**

Digunakan untuk mengambil digit ratusan dari bilangan yang dimasukkan. Karena bilangan dibagi 100, hasil pembagian akan menghilangkan dua digit terakhir dan menyisakan digit pertama.

- **d2 = bilangan % 100 / 10**

Baris ini digunakan untuk mengambil digit puluhan dari bilangan. bilangan % 100 → mengambil sisa dua digit terakhir (misal $345 \% 100 = 45$). / 10 → menghilangkan satu digit terakhir sehingga tersisa digit puluhan.

- **d3 = bilangan % 100 % 10**

Digunakan untuk mengambil digit satuan dari bilangan. bilangan % 100 → mengambil dua digit terakhir (misal $345 \% 100 = 45$). % 10 → mengambil sisa pembagian 10, yaitu digit terakhir dari 45 → 5. Jadi d3 berisi angka satuan dari bilangan input.

- **fmt.Println(d1 <= d2 && d2 <= d3)**

Berfungsi untuk mengecek apakah ketiga digit bilangan berurutan naik (dari kecil ke besar). Operator <= digunakan untuk membandingkan apakah satu angka lebih kecil atau sama dengan angka berikutnya. Operator && berarti dan, artinya kedua kondisi harus benar agar hasil akhir true. Jadi, hasilnya akan true jika urutan digit dari kiri ke kanan semakin besar atau tetap, dan false jika tidak.

3. Guided 3

Source Code

```
package main

import "fmt"

func main () {

    var beratBadan, tinggiBadan, bmi float64

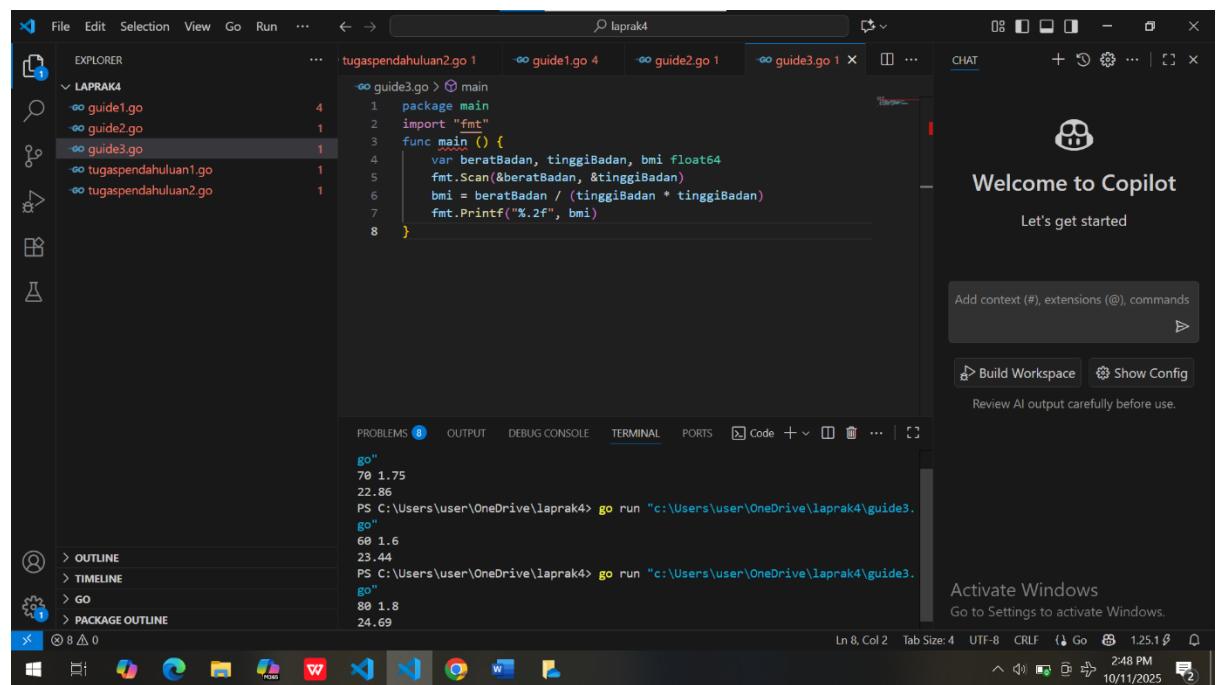
    fmt.Scan(&beratBadan, &tinggiBadan)

    bmi = beratBadan / (tinggiBadan * tinggiBadan)

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

}
```

Screenshot program



Deskripsi program

- **package main**

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalanannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

- **import "fmt"**

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

- **Tanda buka dan tutup kurung kurawal { }**

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

- **var beratBadan, tinggiBadan, bmi float64**

Baris ini digunakan untuk mendeklarasikan tiga variabel dengan tipe data float64, yaitu: beratBadan → untuk menyimpan berat badan pengguna (dalam kilogram). tinggiBadan → untuk menyimpan tinggi badan pengguna (dalam meter). bmi → untuk menyimpan hasil perhitungan nilai Body Mass Index (BMI). Tipe float64 dipakai karena data yang digunakan melibatkan bilangan desimal, bukan bilangan bulat.

- **fmt.Scan(&beratBadan, &tinggiBadan)**

Digunakan untuk membaca dua input angka dari keyboard, yaitu berat badan dan tinggi badan. Tanda & di depan nama variabel berarti alamat dari variabel tersebut — agar data input bisa langsung disimpan ke dalam variabel yang benar. Tanpa tanda &, program tidak akan tahu ke mana harus meletakkan data yang diketik oleh pengguna.

- **bmi = beratBadan / (tinggiBadan * tinggiBadan)**

Baris ini berfungsi untuk menghitung nilai BMI (Body Mass Index) dengan rumusnya lalu Hasilnya disimpan di variabel bmi.

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

Berfungsi untuk menampilkan hasil perhitungan BMI ke layar dengan format dua angka di belakang koma. Printf digunakan untuk mencetak output dengan format tertentu. %.2f artinya angka ditampilkan dalam bentuk bilangan desimal (float) dengan dua angka di belakang koma.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var harga, diskon, total float64

    fmt.Print("Masukkan total belanja awal:")
    fmt.Scan(&harga)

    fmt.Print("Masukkan diskon (%):")
    fmt.Scan(&diskon)

    total = harga * (1 - diskon/100)

    fmt.Println("Masukkan harga setelah diskon: ", total)
}
```

Screenshoot program

```
File Edit Selection View Go Run ... ← → ⌂ laprak4
EXPLORER LAPRAK4
guide1.go 7
guide2.go 1
guide3.go 1
tugas1.go 1
tugas2.go 1
tugas3.go 1
tugaspendahuluan1.go 1
tugaspendahuluan2.go 1
guide1.go 7
guide2.go 1
guide3.go 1
tugas1.go > main
2
3 import "fmt"
4
5 func main() {
6     var harga, diskon, total float64
7
8     fmt.Println("Masukkan total belanja awal:")
9     fmt.Scan(&harga)
10    fmt.Println("Masukkan diskon (%):")
11    fmt.Scan(&diskon)
12    total = harga * (1 - diskon/100)
13
14    fmt.Println("harga setelah diskon: ", total)
15 }
16

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Masukkan diskon (%):10
harga setelah diskon: 90000
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugas1.go"
Masukkan total belanja awal:150000
Masukkan diskon (%):20
harga setelah diskon: 120000
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugas1.go"
Masukkan total belanja awal:150000
Masukkan diskon (%):15
harga setelah diskon: 127500
PS C:\Users\user\OneDrive\laprak4>

Activate Windows
Go to Settings to activate Windows.

Ln 14, Col 18 Tab Size: 4 UTF-8 CRLF ⓘ Go ⌂ 5:15 PM 10/12/2025
```

Deskripsi program

- **package main**

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalanannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

- **import "fmt"**

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

- **Tanda buka dan tutup kurung kurawal { }**

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika

kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

- **var harga, diskon, total float64**

Berfungsi untuk mendeklarasikan tiga variabel dengan tipe data float64, yaitu: harga → digunakan untuk menyimpan total belanja awal sebelum diskon. diskon → digunakan untuk menyimpan persentase diskon yang diberikan (contoh: 10 untuk 10%). total → digunakan untuk menyimpan hasil akhir harga setelah diskon diterapkan. Dengan tipe data float64, program bisa mengolah bilangan desimal agar hasil perhitungannya lebih akurat.

- **fmt.Println("Masukkan total belanja awal:")**

Berfungsi untuk menampilkan teks ke layar agar pengguna tahu bahwa program sedang meminta nilai total belanja awal. Fungsi fmt.Println tidak membuat baris baru, sehingga input pengguna akan diketik di baris yang sama dengan teks tersebut. Tanpa perintah ini, pengguna tidak tahu apa yang harus dimasukkan.

- **fmt.Scan(&harga)**

Berfungsi untuk membaca input dari keyboard dan menyimpannya ke variabel harga. Tanda & berarti alamat variabel, tempat nilai input disimpan. Tanpa tanda &, nilai dari input tidak akan bisa disimpan di dalam variabel.

- **fmt.Println("Masukkan diskon (%):")**

Menampilkan teks agar pengguna tahu bahwa program sedang meminta persentase diskon. Tulisan ini berfungsi sebagai panduan agar pengguna memasukkan angka diskon yang benar (misalnya: 10, 25, atau 50). Fungsi ini juga memakai fmt.Println supaya input tetap di baris yang sama.

- **fmt.Scan(&diskon)**

Digunakan untuk membaca nilai diskon dari keyboard dan menyimpannya ke variabel diskon. Seperti sebelumnya, tanda & penting karena menunjukkan tempat menyimpan data input. Setelah baris ini dijalankan, nilai diskon akan siap digunakan dalam perhitungan total harga.

- **total = harga * (1 - diskon/100)**

Baris ini merupakan rumus utama untuk menghitung harga setelah diskon. diskon / 100 digunakan untuk mengubah nilai persen menjadi bentuk desimal (contoh: 20 → 0.2). 1 - diskon/100 menghasilkan persentase harga yang harus dibayar (contoh: 1 - 0.2 = 0.8). Kemudian dikalikan dengan harga untuk mendapatkan total harga akhir setelah potongan diskon diterapkan.

- **fmt.Println("harga setelah diskon: ", total)**

Digunakan untuk menampilkan hasil akhir ke layar. Fungsi fmt.Println mencetak teks dan nilai total di layar dengan jarak spasi otomatis, serta menambahkan baris baru di akhir output agar tampilan rapi.

2. Tugas 2

Source code

```
package main

import "fmt"

func main() {
    var tinggiBadan, bmi, beratBadan float64

    fmt.Print("Masukkan nilai bmi: ")
    fmt.Scan(&bmi)

    fmt.Print("Masukkan tinggi badan: ")
    fmt.Scan(&tinggiBadan)

    beratBadan = bmi * (tinggiBadan * tinggiBadan)

    fmt.Printf("Berat badan: %0.f kg\n", beratBadan)
}
```

Screenshoot program

```
File Edit Selection View Go Run ... ← → ⌂ laprak4
EXPLORER LAPRAK4
guide1.go 6
guide2.go 1
guide3.go 1
tugas1.go 1
tugas2.go > ...
3 import "fmt"
4
5 func main() {
6     var tinggiBadan, bmi, beratBadan float64
7
8     fmt.Print("Masukkan nilai bmi: ")
9     fmt.Scan(&bmi)
10    fmt.Print("Masukkan tinggi badan: ")
11    fmt.Scan(&tinggiBadan)
12
13    beratBadan = bmi * (tinggiBadan * tinggiBadan)
14
15    fmt.Printf("Berat badan: %0.f kg\n", beratBadan)
16
17 }

PROBLEMS 12 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + ... | ⌂
Masukkan nilai bmi: 22.86
Masukkan tinggi badan: 1.75
Berat badan: 70 kg
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugas2.go"
Masukkan nilai bmi: 23.43
Masukkan tinggi badan: 1.6
Berat badan: 60 kg
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugas2.go"
Masukkan nilai bmi: 24.69

Activate Windows
Go to Settings to activate Windows.

Ln 17, Col 1 Tab Size: 4 UTF-8 CRLF ⌂ Go ⌂ 12.5.1 ⌂ 3:45 PM 10/11/2025 ⌂
```

Deskripsi program

- **package main**

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalanannya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

- **import "fmt"**

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan.

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

- **Tanda buka dan tutup kurung kurawal { }**

Artinya kode utama program dimulai dan diakhiri di dalam kurung kurawal tersebut. Go menggunakan kurung kurawal untuk menandai batas awal dan akhir blok kode. Jika

kurung ini tidak ada, maka program tidak tahu batas instruksi mana saja yang termasuk dalam fungsi main.

- **var tinggiBadan, bmi, beratBadan float64**

Berfungsi untuk mendeklarasikan tiga variabel dengan tipe data float64, yaitu: tinggiBadan → digunakan untuk menyimpan nilai tinggi badan pengguna (dalam meter). bmi → digunakan untuk menyimpan nilai indeks massa tubuh (Body Mass Index). beratBadan → digunakan untuk menyimpan hasil perhitungan berat badan berdasarkan BMI dan tinggi badan. Tipe float64 dipilih karena hasil perhitungannya bisa berupa bilangan desimal.

- **fmt.Println("Masukkan nilai bmi: ")**

Digunakan untuk menampilkan teks ke layar agar pengguna tahu bahwa mereka harus memasukkan nilai BMI terlebih dahulu. fmt.Println menampilkan teks tanpa berpindah baris, sehingga input pengguna diketik di baris yang sama dengan teks petunjuk tersebut.

- **fmt.Scan(&bmi)**

Berfungsi untuk membaca nilai BMI yang diketik oleh pengguna dan menyimpannya ke variabel bmi. Tanda & menunjukkan alamat variabel, agar data hasil input bisa langsung disimpan ke dalam variabel yang dituju. Tanpa tanda &, program tidak tahu ke mana hasil input harus disimpan.

- **fmt.Println("Masukkan tinggi badan: ")**

Menampilkan teks petunjuk agar pengguna memasukkan nilai tinggi badan dalam meter. Teks ini muncul di layar untuk membantu pengguna memahami urutan input yang harus diberikan.

- **fmt.Scan(&tinggiBadan)**

Digunakan untuk membaca input tinggi badan dari keyboard dan menyimpannya ke variabel tinggiBadan. Seperti sebelumnya, tanda & digunakan agar hasil input langsung masuk ke dalam variabel tersebut. Nilai tinggi badan ini akan digunakan dalam rumus perhitungan berat badan.

- **beratBadan = bmi * (tinggiBadan * tinggiBadan)**

Baris ini adalah rumus utama untuk menghitung berat badan berdasarkan nilai BMI (Body Mass Index).

- **fmt.Printf("Berat badan: %0.f kg\n", beratBadan)**

Berfungsi untuk menampilkan hasil perhitungan ke layar dalam format yang rapi. fmt.Printf digunakan agar hasil bisa diformat sesuai keinginan. %0.f berarti hasil angka akan dibulatkan menjadi bilangan bulat tanpa desimal. \n berfungsi untuk membuat baris baru setelah hasil ditampilkan.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var ax, ay, bx, by, cx, cy float64

    fmt.Scan(&ax, &ay)
    fmt.Scan(&bx, &by)
    fmt.Scan(&cx, &cy)

    AB := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))
    BC := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))
    CA := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))

    terpanjang := math.Max(AB, math.Max(BC, CA))
    fmt.Printf("%.2f\n", terpanjang)
}
```

Screenshot program

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File Explorer:** Shows a tree view of files under the folder "LAPRAK4". The file "tugas3.go" is selected.
- Code Editor:** Displays the content of "tugas3.go". The code calculates the distances between three points (A, B, C) and prints the maximum distance.

```
guide2.go 1 | guide3.go 1 | tugas1.go 1 | tugas2.go 1 | tugas3.go 1 | package main Untitled-1
```

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var ax, ay, bx, by, cx, cy float64

    fmt.Scan(&ax, &ay)
    fmt.Scan(&bx, &by)
    fmt.Scan(&cx, &cy)

    AB := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))
    BC := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))
    CA := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))

    terpanjang := math.Max(AB, math.Max(BC, CA))
    fmt.Printf("%.2f\n", terpanjang)
```

- Terminal:** Shows the output of running the code in the terminal.

```
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugas3.go"
1.0 1.0
4.0 1.0
1.0 5.0
5.00
PS C:\Users\user\OneDrive\laprak4> go run "c:\Users\user\OneDrive\laprak4\tugas3.go"
0.0 0.0
3.0 0.0
3.0 4.0
5.00
PS C:\Users\user\OneDrive\laprak4>
```

- Status Bar:** Shows the current file is "tugas3.go", line 19, column 49, with tab size 4, UTF-8 encoding, and CRLF line endings.
- Bottom Icons:** Includes icons for file operations like Open, Save, and Close, as well as other standard OS icons.

Deskripsi program

- package main

Biasanya digunakan untuk menandai bahwa file tersebut adalah program utama, bagian dari kode yang pertama kali dijalankan untuk menjalankan program Go. Package ini digunakan bersamaan dengan fungsi main() sebagai titik awal berjalananya program. Tanpa package main, program tidak bisa dijalankan sebagai aplikasi.

- import "fmt" "math"

Berfungsi supaya program bisa membaca input dari keyboard menggunakan Scan dan menampilkan output ke layar menggunakan Print atau Println. Tanpa mengimpor fmt, perintah input-output seperti fmt.Scan atau fmt.Println tidak dapat digunakan. Paket fmt merupakan singkatan dari format, karena digunakan untuk memformat teks, angka, dan data yang ditampilkan. math → digunakan untuk perhitungan matematika yang lebih kompleks seperti akar kuadrat (math.Sqrt) dan pangkat (math.Pow).

- **func main()**

Sebagai panggung utama dari program Go. Semua proses utama dijalankan di dalam fungsi ini. Berfungsi untuk mengatur alur kerja program, yaitu mulai dari membaca input, memproses data, hingga menampilkan hasil ke layer. Tanpa main(), program tidak akan bisa dijalankan.

- **var ax, ay, bx, by, cx, cy float64**

Digunakan untuk mendeklarasikan enam variabel bertipe float64, yaitu: ax, ay → menyimpan koordinat titik A. bx, by → menyimpan koordinat titik B. cx, cy → menyimpan koordinat titik C. Tipe data float64 digunakan karena koordinat bisa berupa angka desimal (misalnya 3.5 atau -2.7). Semua variabel ini nantinya dipakai untuk menghitung jarak antar titik.

- **fmt.Scan(&ax, &ay)**

Digunakan untuk membaca dua nilai input pertama dari pengguna, yaitu koordinat titik A (x dan y). Tanda & menunjukkan bahwa nilai input akan disimpan langsung ke alamat variabel ax dan ay. Tanpa tanda &, input tidak akan bisa tersimpan.

- **fmt.Scan(&bx, &by)**

Digunakan untuk membaca koordinat titik B dari input pengguna. Input kedua ini berisi dua nilai: bx (sumbu x) dan by (sumbu y). Nilai yang dimasukkan akan digunakan untuk menghitung jarak dari A ke B dan dari B ke C.

- **fmt.Scan(&cx, &cy)**

Digunakan untuk membaca koordinat titik C dari pengguna. Input ketiga berisi dua nilai: cx dan cy. Dengan tiga titik (A, B, dan C) yang sudah dimasukkan, program dapat menghitung panjang semua sisi segitiga.

- **AB := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))**

Baris ini menghitung panjang sisi AB antara titik A dan titik B menggunakan rumus jarak Euclidean. (bx - ax) menghitung jarak horizontal antara A dan B. (by - ay) menghitung jarak vertikal antara A dan B. math.Pow(..., 2) digunakan untuk memangkatkan nilai dengan 2. math.Sqrt(...) mengambil akar kuadrat dari hasil penjumlahan tersebut untuk mendapatkan jarak sebenarnya.

- **BC := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))**

Baris ini menghitung panjang sisi BC dengan cara yang sama seperti sisi AB, namun titik yang digunakan adalah B dan C. Rumusnya juga menggunakan teorema Pythagoras untuk mencari jarak antar dua titik.

- **CA := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))**

Baris ini menghitung panjang sisi CA, yaitu jarak antara titik C dan titik A. Dengan begitu, sekarang program sudah memiliki tiga sisi segitiga lengkap: AB, BC, dan CA.

- **terpanjang := math.Max(AB, math.Max(BC, CA))**

Digunakan untuk menentukan sisi yang paling panjang di antara ketiga sisi segitiga. Fungsi math.Max() akan membandingkan dua nilai dan mengembalikan nilai yang lebih besar. Pertama, math.Max(BC, CA) mencari sisi terpanjang antara BC dan CA. Lalu hasilnya dibandingkan lagi dengan AB menggunakan math.Max(AB, ...). Hasil akhirnya disimpan dalam variabel terpanjang.

- **fmt.Printf("%f\n", terpanjang)**

Berfungsi untuk menampilkan hasil akhir (sisi terpanjang) ke layar. fmt.Printf digunakan untuk mengatur format tampilan angka. "%f" berarti hasil akan ditampilkan dengan dua angka di belakang koma. \n berfungsi untuk membuat baris baru setelah hasil ditampilkan.