

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 3
I/O, TIPE DATA & VARIABEL



Disusun oleh:
ALMA BONITA MIA WARDHANA
109082500015
S1IF-13-04

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 sisi, volume float64

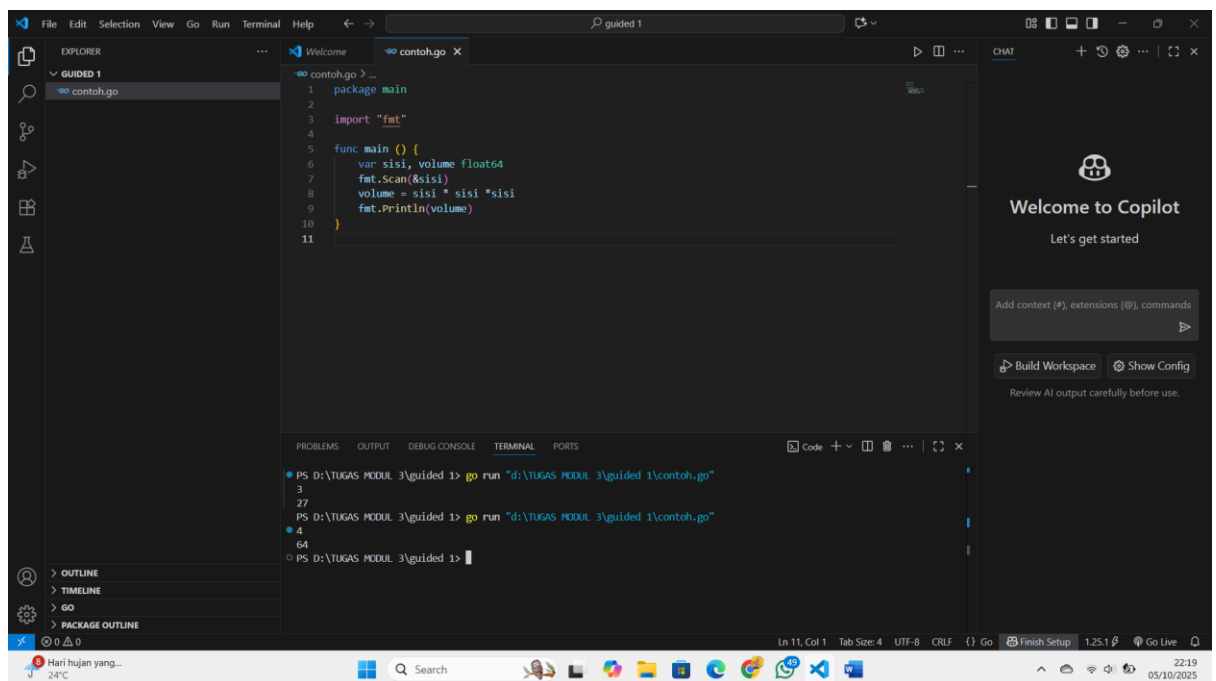
    fmt.Scan(&sisi)

    volume = sisi * sisi *sisi

    fmt.Println(volume)

}
```

Screenshoot program



Deskripsi program

Program di atas ditulis dengan bahasa Go dan berfungsi untuk menghitung volume kubus dari panjang sisi yang dimasukkan pengguna.

Baris package main menunjukkan bahwa program ini adalah program utama yang bisa dijalankan.

import "fmt" digunakan agar program dapat memakai fungsi input dan output seperti fmt.Scan dan fmt.Println.

Di dalam func main(), dua variabel dideklarasikan yaitu sisi dan volume dengan tipe float64, yaitu tipe data untuk angka desimal.

fmt.Scan(&sisi) berfungsi membaca nilai sisi dari pengguna lewat keyboard. Tanda & artinya nilai input disimpan langsung ke variabel sisi.

Setelah itu, $\text{volume} = \text{sisi} * \text{sisi} * \text{sisi}$ menghitung volume kubus dengan rumus sisi^3 .

Terakhir, fmt.Println(volume) menampilkan hasil perhitungannya di layar.

Singkatnya, program ini meminta pengguna memasukkan panjang sisi kubus, menghitung volumenya, lalu menampilkan hasilnya.

2. Guided 2

Source Code

```
package main

import "fmt"

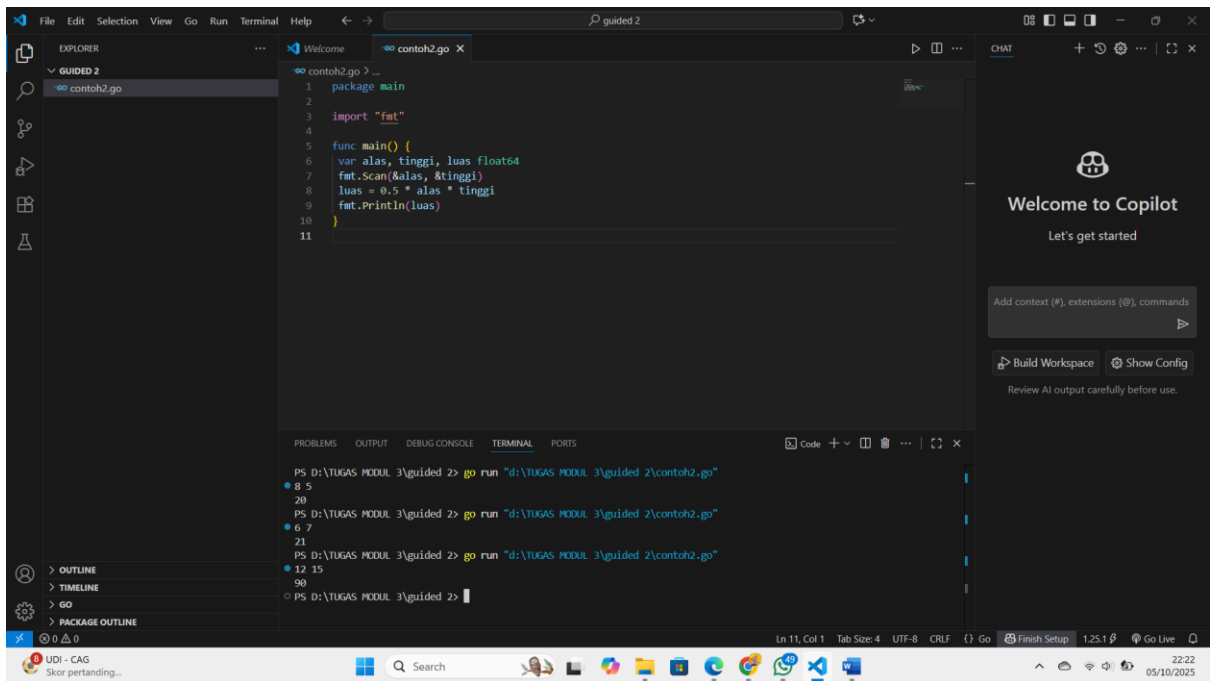
func main() {
    var alas, tinggi, luas float64

    fmt.Scan(&alas, &tinggi)

    luas = 0.5 * alas * tinggi

    fmt.Println(luas)
}
```

Screenshoot program



Deskripsi program

Program di atas ditulis dengan bahasa Go dan berfungsi untuk menghitung luas segitiga berdasarkan nilai alas dan tinggi yang dimasukkan oleh pengguna.

Baris package main menunjukkan bahwa program ini merupakan program utama yang bisa dijalankan, dan Go akan mencari fungsi main() sebagai titik awal eksekusi. import "fmt" digunakan untuk mengimpor paket fmt, yang berisi fungsi input/output seperti Scan, Println, dan Printf.

Di dalam func main(), terdapat tiga variabel alas, tinggi, dan luas yang bertipe float64 untuk menyimpan angka pecahan. Nilai awal variabel ini adalah 0.0.

Fungsi fmt.Scan(&alas, &tinggi) membaca dua nilai dari input pengguna, yaitu alas dan tinggi segitiga, lalu menyimpannya ke variabel yang sesuai.

Setelah itu, luas = 0.5 * alas * tinggi menghitung luas segitiga menggunakan rumus $\frac{1}{2} \times \text{alas} \times \text{tinggi}$.

Terakhir, fmt.Println(luas) menampilkan hasil perhitungannya di layar.

Secara singkat, program ini meminta pengguna memasukkan nilai alas dan tinggi segitiga, lalu menghitung dan menampilkan hasil luasnya.

3. Guided 3

Source Code

```
package main

import "fmt"
```

```
func main() {

    var rupiah, dolar int

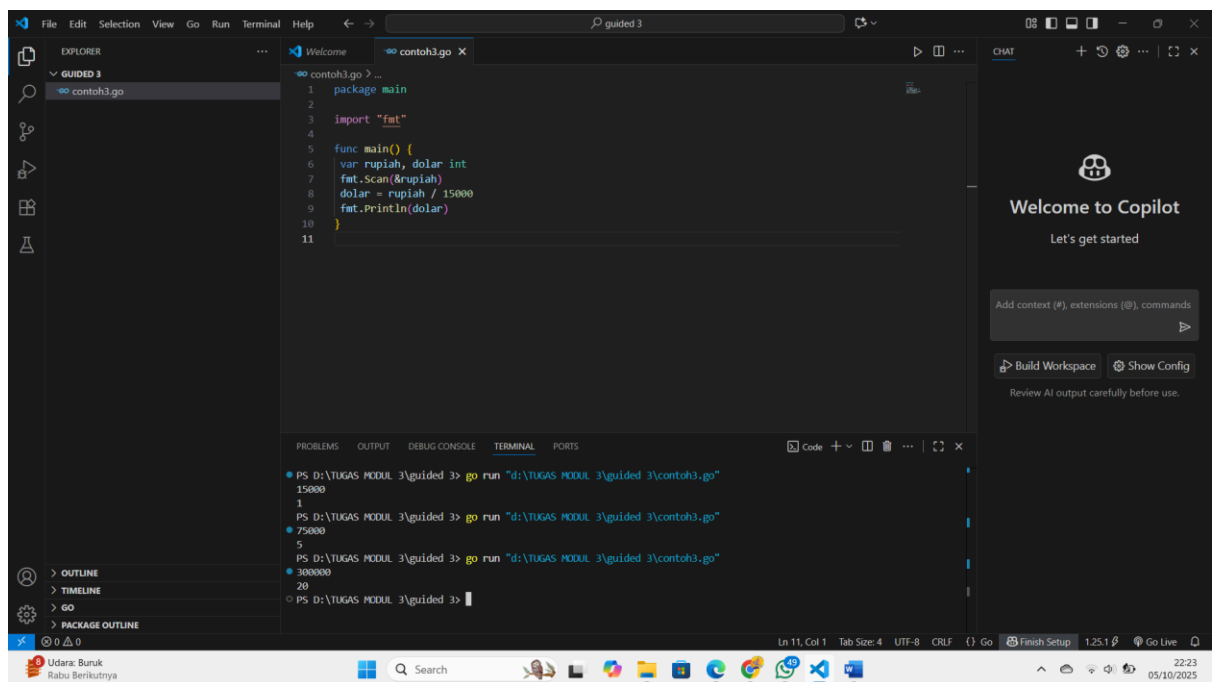
    fmt.Scan(&rupiah)

    dolar = rupiah / 15000

    fmt.Println(dolar)

}
```

Screenshoot program



Deskripsi program

baris package main menandakan file ini bagian dari program utama yang bisa dijalankan; Go akan mulai mengeksekusi dari fungsi main(). import "fmt" mengimpor paket untuk input/output sederhana (mis. fmt.Scan dan fmt.Println). Di func main() ada deklarasi var rupiah, dolar int yang membuat dua variabel bertipe int (bilangan bulat). fmt.Scan(&rupiah) membaca satu nilai dari input (keyboard) dan menyimpan ke variabel rupiah lewat alamatnya (& menunjukkan pointer). Baris dolar = rupiah / 15000 melakukan pembagian antara rupiah dan konstanta 15000; karena kedua operand bertipe int, pembagian ini adalah pembagian bulat sehingga hasilnya dibulatkan ke bawah (truncated) mis. jika rupiah = 16000 maka dolar akan bernilai 1, bukan 1.066.... Terakhir fmt.Println(dolar) mencetak nilai dolar ke layar. Beberapa hal penting: nilai awal rupiah/dolar adalah 0 kalau belum diisi; kode ini tidak memeriksa apakah fmt.Scan berhasil (tidak ada pengecekan error), dan kurs 15000 ditulis

langsung (hard-coded) padahal kurs sesungguhnya berubah-ubah. Karena tipe int digunakan, hasil konversi tidak akurat untuk nilai rupiah yang tidak kelipatan 15000 — jika mau hasil desimal akurat, gunakan tipe float64 dan format output (mis. 2 desimal).

TUGAS

1. Tugas 1

Source code

```
package main

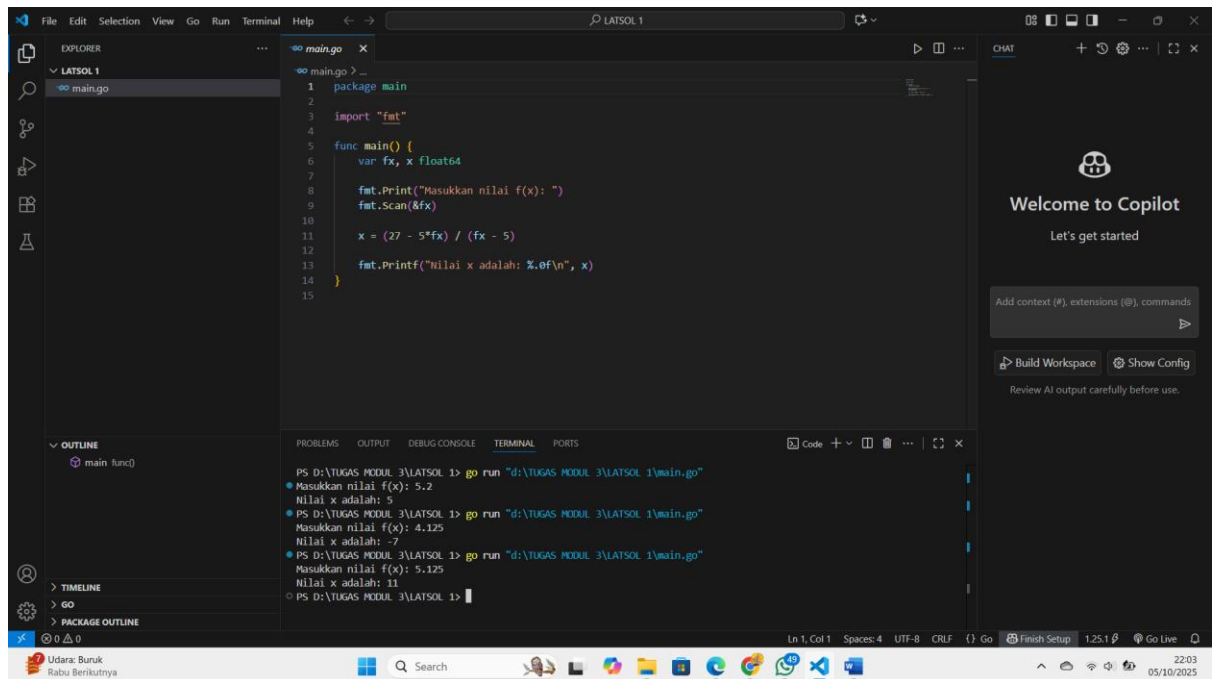
import "fmt"

func main() {
    var fx, x float64

    fmt.Print("Masukkan nilai f(x): ")
    fmt.Scan(&fx)

    x = (27 - 5*fx) / (fx - 5)

    fmt.Printf("Nilai x adalah: %.0f\n", x)
}
```



Deskripsi program

`package main` — menyatakan bahwa file ini bagian dari program utama yang dapat dijalankan; Go akan mencari fungsi `main()` sebagai titik masuk. `import "fmt"` — mengimpor paket `fmt` untuk fungsi input/output (seperti `Print`, `Scan`, `Printf`). `func main() { ... }` — deklarasi fungsi `main` tempat eksekusi dimulai. `var fx, x float64` — mendeklarasikan dua variabel bertipe `float64` (angka pecahan): `fx` untuk menampung input `f(x)` dan `x` untuk menyimpan hasil perhitungan; nilai awal keduanya adalah `0.0`. `fmt.Print("Masukkan nilai f(x): ")` — menampilkan prompt agar pengguna tahu harus memasukkan nilai. `fmt.Scan(&fx)` — membaca satu nilai dari keyboard dan menyimpannya ke `fx`; `&` berarti memberi alamat variabel (pointer) supaya `Scan` bisa menulis nilainya langsung. Catatan: `fmt.Scan` sebenarnya mengembalikan dua nilai (`n` int, `err` error) tetapi kode ini tidak memeriksa return tersebut, jadi input yang salah tidak ditangani. Baris `x = (27 - 5*fx) / (fx - 5)` melakukan perhitungan: pertama `5*fx` (perkalian) dieksekusi, lalu `27 - (5*fx)` memberi pembilang, sedangkan `fx - 5` adalah penyebut; kemudian pembilang dibagi penyebut. Perlu diperhatikan kasus khusus `fx == 5` — untuk tipe `float64` pembagian dengan nol menghasilkan `+Inf`, `-Inf` atau `NaN` sesuai konteks (bukan panic seperti pembagian integer), namun itu bukan hasil yang berguna; jadi sebaiknya hindari atau periksa `fx == 5`/kedekatan dengan nol sebelum membagi. Terakhir `fmt.Printf("Nilai x adalah: %.0f\n", x)` mencetak hasil `x` dengan format `%.0f` (nol angka di belakang koma), artinya nilai akan dibulatkan ke bilangan bulat terdekat saat ditampilkan; `\n` menambah newline.

2. Tugas 2

Source code

```
package main
```

```
import "fmt"

func main() {

    var r float64

    var pi float64

    var volume float64

    var luas float64


    fmt.Print("Jejari = ")

    fmt.Scan(&r)

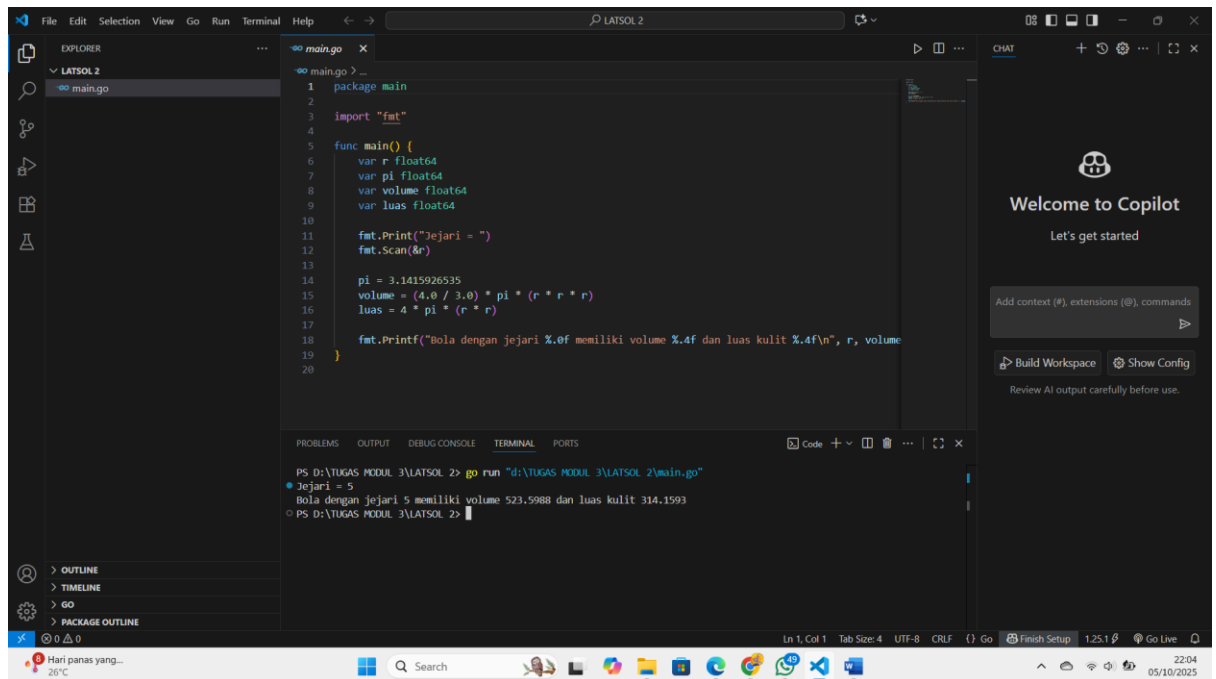

    pi = 3.1415926535

    volume = (4.0 / 3.0) * pi * (r * r * r)

    luas = 4 * pi * (r * r)


    fmt.Printf("Bola dengan jejari %.0f memiliki volume  
%.4f dan luas kulit %.4f\n", r, volume, luas)

}
```

Deskripsi program

Kode ini ditulis dalam bahasa Go dan fungsinya membaca jari-jari dari pengguna lalu menyiapkan variabel untuk menghitung luas/volume: package main menandakan program utama; import "fmt" mengimpor paket input/output; func main() { adalah titik masuk program. Di dalamnya dideklarasikan empat variabel bertipe float64: r (jari-jari), pi, volume, dan luas — semua berisi nilai nol awal. fmt.Print("Jari-jari = ") menampilkan prompt, lalu fmt.Scan(&r) membaca nilai jari-jari dari keyboard (& berarti alamat/pointer supaya nilai disimpan di r). Baris pi = 3.1415926535 menetapkan konstanta π secara manual. Variabel luas dan volume belum dipakai di potongan ini, tetapi biasanya selanjutnya akan dihitung dengan rumus umum (mis. luas = $\pi * r * r$ untuk luas lingkaran, atau volume = $\frac{4.0}{3.0} * \pi * r * r * r$ untuk volume bola). Catatan penting: sebaiknya periksa return value dari fmt.Scan untuk menangani input tidak valid, gunakan math.Pi daripada menulis π sendiri, dan hindari deklarasi variabel yang tidak dipakai (atau gabungkan var r, pi, volume, luas float64) agar kode lebih aman dan rapi; juga gunakan fmt.Printf jika ingin mengatur format output (mis. jumlah desimal).

3. Tugas 3

Source code

```

package main

import "fmt"

func main() {
    var tahun int
    var kabisat bool

```

```

    fmt.Print("Tahun: ")
    fmt.Scan(&tahun)

    if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100
    != 0) {
        kabisat = true
    } else {
        kabisat = false
    }

    fmt.Printf("Kabisat: %t\n", kabisat)
}

```

The screenshot shows a Go IDE with the following components:

- Explorer:** Shows the project structure with a file named `main.go`.
- Editor:** Displays the Go code for `main.go`, which is identical to the code block above.
- Terminal:** Shows the output of running the program three times:
 - Input: 2016, Output: Kabisat: true
 - Input: 2000, Output: Kabisat: true
 - Input: 2018, Output: Kabisat: false
- Chat:** A sidebar on the right with a "Welcome to Copilot" message and options to "Build Workspace" or "Show Config".
- Bottom Bar:** Shows the status bar with "Ln 1, Col 1", "Tab Size: 4", "UTF-8", "CRLF", and "Go".

Deskripsi program

Program ini ditulis dalam Go untuk mengecek apakah sebuah tahun adalah tahun kabisat. package main dan import "fmt" seperti biasa menandakan program utama dan paket I/O yang dipakai. Di func main() dideklarasikan var tahun int (penyimpan tahun yang diinput) dan var kabisat bool (menyimpan hasil — true jika kabisat, false jika bukan; default false). fmt.Print("Tahun: ") menampilkan prompt lalu fmt.Scan(&tahun) membaca angka tahun dari keyboard (alamat &tahun dipakai agar nilai disimpan ke variabel).

Artinya sebuah tahun kabisat jika habis dibagi 400 (mis. 2000), atau habis dibagi 4 tetapi tidak habis dibagi 100 (mis. 2024 → true, 1900 → false). Operator % adalah

modulus (sisa bagi). Akhirnya `fmt.Printf("Kabisat: %t\n", kabisat)` mencetak hasil boolean (%t menampilkan true/false). Catatan singkat: sebaiknya periksa return value `fmt.Scan` untuk menangani input tidak valid, Anda bisa menyederhanakan dengan `kabisat = (tahun%400==0) || (tahun%4==0 && tahun%100!=0)`, dan ingat bahwa untuk tahun negatif atau input non-angka perlu penanganan tambahan.

4. Tugas 3

Source code

```
package main

import "fmt"

func main() {

    var celsius float64

    var reamur float64

    var fahrenheit float64

    var kelvin float64


    fmt.Print("Temperatur Celsius: ")

    fmt.Scan(&celsius)


    reamur = (4.0 / 5.0) * celsius

    fahrenheit = (9.0/5.0)*celsius + 32

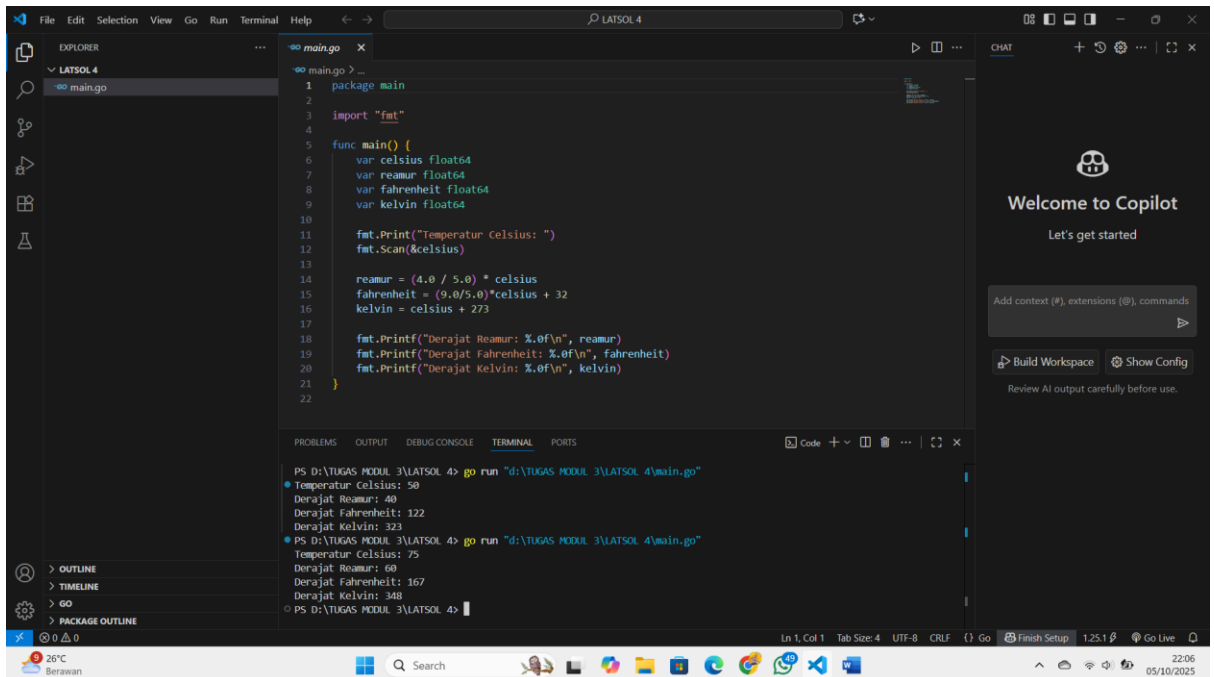
    kelvin = celsius + 273


    fmt.Printf("Derajat Reamur: %.0f\n", reamur)

    fmt.Printf("Derajat Fahrenheit: %.0f\n",
fahrenheit)

    fmt.Printf("Derajat Kelvin: %.0f\n", kelvin)
```

```
}
```



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var celsius float64
7     var reamur float64
8     var fahrenheit float64
9     var kelvin float64
10
11     fmt.Print("Temperatur Celsius: ")
12     fmt.Scan(&celsius)
13
14     reamur = (4.0 / 5.0) * celsius
15     fahrenheit = (9.0/5.0)*celsius + 32
16     kelvin = celsius + 273
17
18     fmt.Printf("Derajat Reamur: %.0f\n", reamur)
19     fmt.Printf("Derajat Fahrenheit: %.0f\n", fahrenheit)
20     fmt.Printf("Derajat Kelvin: %.0f\n", kelvin)
21 }
22
```

Terminal Output:

```
PS D:\TUGAS MODUL 3\LATSOL 4> go run "d:\TUGAS MODUL 3\LATSOL 4\main.go"
Temperatur Celsius: 50
Derajat Reamur: 40
Derajat Fahrenheit: 122
Derajat Kelvin: 323
PS D:\TUGAS MODUL 3\LATSOL 4> go run "d:\TUGAS MODUL 3\LATSOL 4\main.go"
Temperatur Celsius: 75
Derajat Reamur: 60
Derajat Fahrenheit: 167
Derajat Kelvin: 348
PS D:\TUGAS MODUL 3\LATSOL 4>
```

Deskripsi program

Program ini ditulis dalam Go untuk mengonversi suhu dari Celsius ke tiga satuan lain: Reamur, Fahrenheit, dan Kelvin. Secara garis besar: `package main` dan `import "fmt"` menandakan program utama dan paket I/O yang dipakai; `func main()` adalah titik masuk. Di dalamnya dideklarasikan empat variabel `float64`: `celsius` (input), `reamur`, `fahrenheit`, dan `kelvin` (hasil konversi). `fmt.Print("Temperatur Celsius: ")` menampilkan prompt, lalu `fmt.Scan(&celsius)` membaca nilai Celsius dari keyboard (nilai disimpan melalui alamat `&celsius`). Rumus yang dipakai: $\text{reamur} = (4.0/5.0) * \text{celsius}$ ($R = 4/5 \cdot C$), $\text{fahrenheit} = (9.0/5.0) * \text{celsius} + 32$ ($F = 9/5 \cdot C + 32$), dan $\text{kelvin} = \text{celsius} + 273$ ($K = C + 273$ — catatan: nilai lebih akurat biasanya 273.15). Hasil dicetak dengan `fmt.Printf(..., "%.0f")` yang menampilkan angka dibulatkan tanpa desimal (mis. input 25 → output Derajat Reamur: 20, Fahrenheit: 77, Kelvin: 298 menurut kode ini). Hal teknis penting: tipe `float64` memungkinkan input desimal; `fmt.Scan` sebenarnya mengembalikan (`n`, `err`) yang sebaiknya diperiksa untuk menangani input tidak valid; dan `%.0f` membulatkan tampilan—jika Anda ingin presisi, gunakan mis. `%.2f`. Saran perbaikan singkat: gunakan `const` untuk konstanta (mis. `const kelvinOffset = 273.15`), periksa error `fmt.Scan`, serta pakai `fmt.Printf("...%.2f\n", value)` agar hasil desimal tampil rapi.