

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 3  
I/O, TIPE DATA & VARIABEL**



**Disusun oleh:**

**Anindya Rahadita Yumnaa**

**109082500138**

**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 sisi int

    var volume float64

    fmt.Print("Masukkan panjang sisi kubus: ")

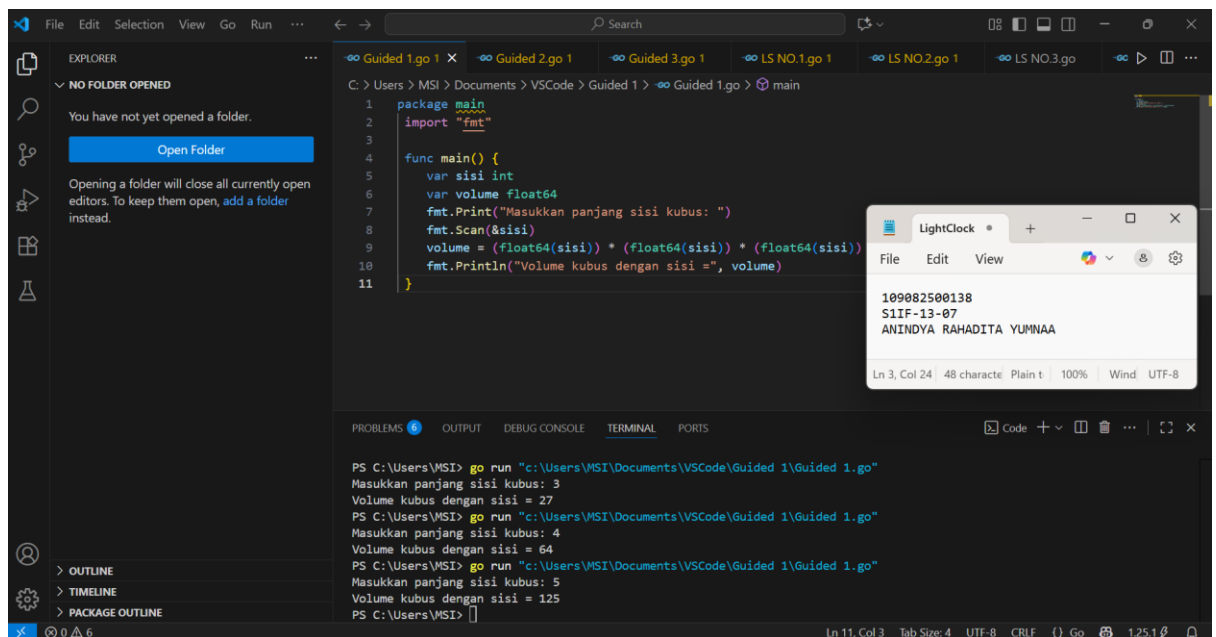
    fmt.Scan(&sisi)

    volume = (float64(sisi)) * (float64(sisi)) *
(float64(sisi))

    fmt.Println("Volume kubus dengan sisi =", volume)

}
```

#### Screenshoot program



## Deskripsi program

### 1. package main

Program ini menandakan bahwa file tersebut adalah **program utama** yang bisa dijalankan (bukan library). Library sendiri itu adalah kode bantu yang dapat berjalan hanya jika dipanggil oleh kode atau program utama, yaitu **package main**. Jika sebuah paket diberi nama main, tool Go akan tahu jika tujuan akhirnya adalah membuat program yang bisa dijalankan.

### 2. import "fmt"

import sendiri berfungsi untuk memanggil kode lain agar fungsinya bisa dipakai di file tersebut. Kemudian, fmt adalah paket standar Go untuk **formatting** dan operasi input atau output sederhana (untuk menulis ke layar dan membaca dari input).

### 3. func main() { ... }

func main() adalah **titik masuk** program, kode di dalam func main() dijalankan pertama kali setelah kode atau program diberi nama. Bentuknya harus **"func main()"** tanpa parameter dan tanpa nilai balik.

### 4. var sisi int

Untuk membuat variabel yang hanya bisa menyimpan angka bilangan bulat.

### 5. var volume float64

Mendeklarasikan dua variabel x dan fx dengan tipe data **float64**. Tipe ini digunakan agar program bisa menangani angka desimal (bilangan pecahan), bukan hanya bilangan bulat.

### 6. fmt.Print("Masukkan panjang sisi kubus: ")

Perintah digunakan untuk **menampilkan teks instruksi** di layar komputer, yaitu "Masukkan panjang sisi kubus: " Teks ini memberi tahu pengguna data apa yang harus dimasukkan.

### 7. fmt.Scan(&sisi)

akan **menunggu pengguna memasukkan angka**. Angka yang diketik dan di-Enter oleh pengguna akan **ditangkap** oleh program dan disimpan ke dalam variabel **sisi**. Simbol (*ampersand*) memastikan data disimpan langsung ke lokasi memori variabel.

### 8. volume = (float64(sisi)) \* (float64(sisi)) \* (float64(sisi))

Baris untuk **melakukan perhitungan** utama. Program menerapkan rumus volume kubus: Sisi Sisi Sisi.

- Nilai dari variabel **sisi** diubah terlebih dahulu menjadi tipe (). Ini dilakukan agar proses perkalian menghasilkan angka desimal yang akurat.
- Hasil perkalian akhir tersebut kemudian **dimasukkan** (disimpan) ke dalam variabel **volume**.

### 9. `fmt.Println("Volume kubus dengan sisi =", volume)`

digunakan untuk **menampilkan hasil perhitungan** di layar. Program akan menampilkan tulisan **"Volume kubus dengan sisi ="** yang diikuti oleh nilai angka yang sudah tersimpan di dalam variabel **volume**.

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var alas float64

    var tinggi float64

    var luas float64

    fmt.Print("Masukkan alas segitiga: ")

    fmt.Scan(&alas)

    fmt.Print("Masukkan tinggi segitiga: ")

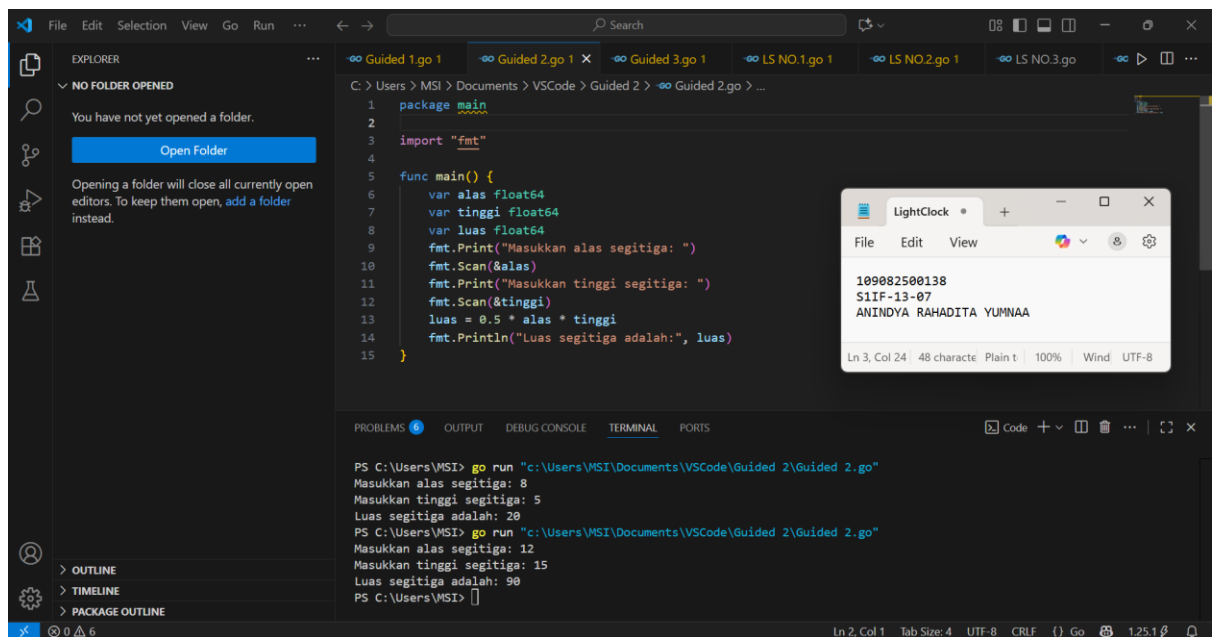
    fmt.Scan(&tinggi)

    luas = 0.5 * alas * tinggi

    fmt.Println("Luas segitiga adalah:", luas)

}
```

### Screenshoot program



## Deskripsi program

### 1. package main

Program ini menandakan bahwa file tersebut adalah **program utama** yang bisa dijalankan (bukan library). Library sendiri itu adalah kode bantu yang dapat berjalan hanya jika dipanggil oleh kode atau program utama, yaitu **package main**. Jika sebuah paket diberi nama main, tool Go akan tahu jika tujuan akhirnya adalah membuat program yang bisa dijalankan.

### 2. import "fmt"

import sendiri berfungsi untuk memanggil kode lain agar fungsinya bisa dipakai di file tersebut. Kemudian, fmt adalah paket standar Go untuk **formatting** dan operasi input atau output sederhana (untuk menulis ke layar dan membaca dari input).

### 3. func main() { ... }

func main() adalah **titik masuk** program, kode di dalam func main() dijalankan pertama kali setelah kode atau program diberi nama. Bentuknya harus **"func main()"** tanpa parameter dan tanpa nilai balik.

### 4. var alas float64

**Mendeklarasikan variabel** bernama **alas** dengan tipe data `float64`. Variabel ini disiapkan untuk menyimpan nilai panjang alas segitiga. Tipe digunakan agar variabel dapat menampung angka desimal (angka berkoma).

### 5. var tinggi float64

Sama seperti baris sebelumnya, perintah ini mendeklarasikan variabel bernama **tinggi** dengan tipe data `float64`. Variabel ini berfungsi untuk menyimpan nilai tinggi segitiga, dan juga disiapkan untuk menampung angka desimal.

### 6. var luas float64

Mendeklarasikan variabel bernama luas dengan tipe data . Variabel ini secara khusus disiapkan untuk menyimpan hasil akhir perhitungan luas segitiga.

#### **7. fmt.Print("Masukkan alas segitiga: ")**

Digunakan untuk **menampilkan instruksi** kepada pengguna di layar. Teks yang muncul adalah "**Masukkan alas segitiga:** ", meminta pengguna memasukkan nilai alas.

#### **8. fmt.Scan(&alas)**

**Menangkap input angka** yang dimasukkan oleh pengguna. Angka tersebut kemudian disimpan ke dalam variabel **alas**. Simbol (*ampersand*) menunjukkan lokasi memori variabel tempat data akan disimpan.

#### **9. fmt.Print("Masukkan tinggi segitiga: ")**

Perintah kembali digunakan untuk **menampilkan instruksi** lanjutan. Teks yang muncul di layar adalah "**Masukkan tinggi segitiga:** ", meminta pengguna memasukkan nilai tinggi.

#### **10. fmt.Scan(&tinggi)**

Perintah akan **menangkap input angka** yang dimasukkan pengguna untuk tinggi. Angka tersebut kemudian disimpan ke dalam variabel **tinggi**.

#### **11. luas = 0.5 \* alas \* tinggi**

Ini adalah baris untuk **melakukan perhitungan** utama. Program menghitung luas segitiga menggunakan rumus: (setengah dikali alas dikali tinggi). Hasil perhitungan ini kemudian **disimpan** ke dalam variabel **luas**.

#### **12. fmt.Println("Luas segitiga adalah:", luas)**

Terakhir, perintah **menampilkan hasil** perhitungan luas segitiga di layar. Teks "**Luas segitiga adalah:**" akan muncul, diikuti oleh **nilai angka** yang tersimpan dalam variabel **luas**.

### **3. Guided 3**

#### **Source Code**

```

package main

import "fmt"

func main() {

    var idr, usd, kurs int

    fmt.Print("Masukkan jumlah uang dalam IDR: ")

    fmt.Scan(&idr)

    fmt.Print("Masukkan nilai tukar USD ke IDR: ")

    fmt.Scan(&kurs)

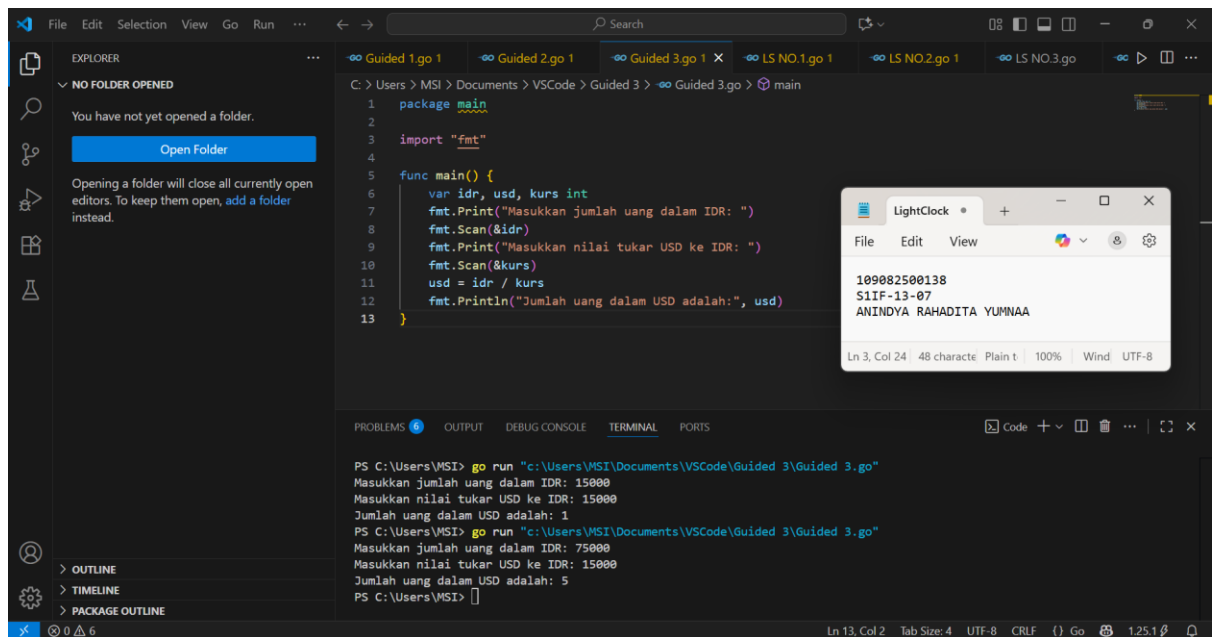
    usd = idr / kurs

    fmt.Println("Jumlah uang dalam USD adalah:", usd)

}

```

## Screenshoot program



## Deskripsi program

### 1. package main

Program ini menandakan bahwa file tersebut adalah **program utama** yang bisa dijalankan (bukan library). Library sendiri itu adalah kode bantu yang dapat berjalan hanya jika dipanggil oleh kode atau program utama, yaitu **package main**. Jika sebuah paket diberi nama main, tool Go akan tahu jika tujuan akhirnya adalah membuat program yang bisa dijalankan.

## 2. import "fmt"

import sendiri berfungsi untuk memanggil kode lain agar fungsinya bisa dipakai di file tersebut. Kemudian, fmt adalah paket standar Go untuk **formatting** dan operasi input atau output sederhana (untuk menulis ke layar dan membaca dari input).

## 3. func main() { ... }

func main() adalah **titik masuk** program, kode di dalam func main() dijalankan pertama kali setelah kode atau program diberi nama. Bentuknya harus **"func main()"** tanpa parameter dan tanpa nilai balik.

## 4. var idr, usd, kurs int

Untuk **mendeklarasikan tiga buah variabel** sekaligus, yaitu **idr**, **usd**, dan **kurs**. Ketiga variabel ini diberikan tipe data (*integer*), yang berarti mereka hanya dapat menyimpan bilangan bulat (angka tanpa koma).

- disiapkan untuk menyimpan jumlah uang dalam Rupiah.
- disiapkan untuk menyimpan hasil konversi ke Dolar.
- disiapkan untuk menyimpan nilai tukar (kurs) dari USD ke IDR.

## 5. fmt.Print("Masukkan jumlah uang dalam IDR: ")

Perintah digunakan untuk **menampilkan instruksi** kepada pengguna di layar. Teks yang muncul adalah **"Masukkan jumlah uang dalam IDR: "**, meminta pengguna memasukkan jumlah uang Rupiah yang ingin dikonversi.

## 6. fmt.Scan(&idr)

Perintah akan **menangkap input angka** yang dimasukkan oleh pengguna. Angka tersebut kemudian disimpan ke dalam variabel **idr**. Simbol (*ampersand*) menunjukkan lokasi memori variabel tempat data akan disimpan.

## 7. fmt.Print("Masukkan nilai tukar USD ke IDR: ")

Digunakan untuk **menampilkan instruksi** lanjutan. Teks yang muncul di layar adalah **"Masukkan nilai tukar USD ke IDR: "**, meminta pengguna memasukkan nilai kurs (misalnya, 15000).

## 8. fmt.Scan(&kurs)

Akan **menangkap input angka** nilai tukar (kurs) yang dimasukkan pengguna. Angka tersebut kemudian disimpan ke dalam variabel **kurs**.

## 9. usd = idr / kurs



Untuk **melakukan perhitungan** konversi. Program menghitung jumlah Dolar dengan **membagi** jumlah Rupiah () dengan nilai tukar (). Hasil pembagian ini kemudian **disimpan** ke dalam variabel **usd**. Perlu dicatat, karena variabel bertipe `int`, hasil pembagian ini akan dibulatkan ke bawah (pembagian bilangan bulat).

**10. `fmt.Println("Jumlah uang dalam USD adalah:", usd)`**  
**menampilkan hasil** konversi mata uang di layar. Teks **"Jumlah uang dalam USD adalah:"** akan muncul, diikuti oleh **nilai angka** yang tersimpan dalam variabel **usd**.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import "fmt"

func main() {
    var fx float64

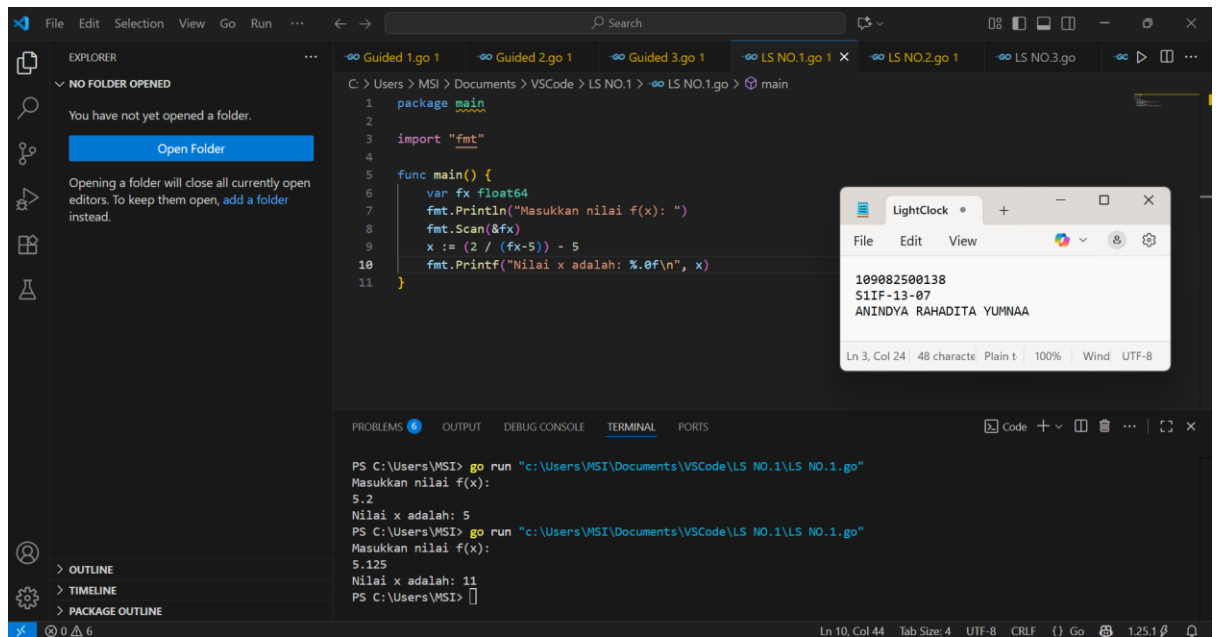
    fmt.Println("Masukkan nilai f(x): ")

    fmt.Scan(&fx)

    x := (2 / (fx-5)) - 5

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

#### Screenshoot program



## Deskripsi program

### 1. package main

Program ini menandakan bahwa file tersebut adalah **program utama** yang bisa dijalankan (bukan library). Library sendiri itu adalah kode bantu yang dapat berjalan hanya jika dipanggil oleh kode atau program utama, yaitu **package main**. Jika sebuah paket diberi nama main, tool Go akan tahu jika tujuan akhirnya adalah membuat program yang bisa dijalankan.

### 2. import "fmt"

import sendiri berfungsi untuk memanggil kode lain agar fungsinya bisa dipakai di file tersebut. Kemudian, fmt adalah paket standar Go untuk **formatting** dan operasi input atau output sederhana (untuk menulis ke layar dan membaca dari input).

### 3. func main() { ... }

func main() adalah **titik masuk** program, kode di dalam func main() dijalankan pertama kali setelah kode atau program diberi nama. Bentuknya harus **"func main()"** tanpa parameter dan tanpa nilai balik.

### 4. var fx float64

Untuk **mendeklarasikan sebuah variabel** bernama **fx**. Variabel ini disiapkan dengan tipe data , yang memungkinkan variabel menyimpan angka desimal (angka berkoma). Variabel digunakan untuk menampung nilai dari fungsi yang akan dimasukkan oleh pengguna.

### 5. fmt.Println("Masukkan nilai f(x): ")

Digunakan untuk **menampilkan instruksi** kepada pengguna di layar. Teks yang muncul adalah **"Masukkan nilai f(x): "**, meminta pengguna memasukkan nilai . Penggunaan membuat kursor berpindah ke baris baru setelah teks ditampilkan.

### 6. fmt.Scan(&fx)

Akan **menangkap input angka** yang dimasukkan oleh pengguna. Angka tersebut kemudian disimpan ke dalam variabel **fx**. Simbol (*ampersand*) menunjukkan lokasi memori variabel tempat data akan disimpan.

#### 7. $x := (2 / (fx-5)) - 5$

Baris untuk **melakukan perhitungan** dan **mendeklarasikan variabel baru** secara singkat.

- Operator digunakan untuk mendeklarasikan dan memberikan nilai awal kepada variabel secara instan. Tipe data akan otomatis menjadi karena hasil perhitungannya adalah desimal.
- Perhitungan yang dilakukan adalah sesuai dengan rumus untuk mencari nilai , yaitu: .
- Hasil perhitungan ini kemudian **disimpan** ke dalam variabel **x**.

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

Digunakan untuk **menampilkan hasil** nilai di layar dengan format yang spesifik.

- Perintah digunakan untuk menampilkan output dengan format khusus.
- adalah **penentu format** yang memberitahu program untuk menampilkan nilai **x** sebagai angka desimal (.) tetapi tanpa angka di belakang koma (0). Artinya, nilai akan dibulatkan ke bilangan bulat terdekat.
- adalah kode untuk **membuat baris baru** setelah output selesai ditampilkan.

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var r int

    fmt.Print("Jari jari = ")
```

```

    fmt.Scan(&r)

    volume := (4.0/3.0) * math.Pi *
math.Pow(float64(r), 3)

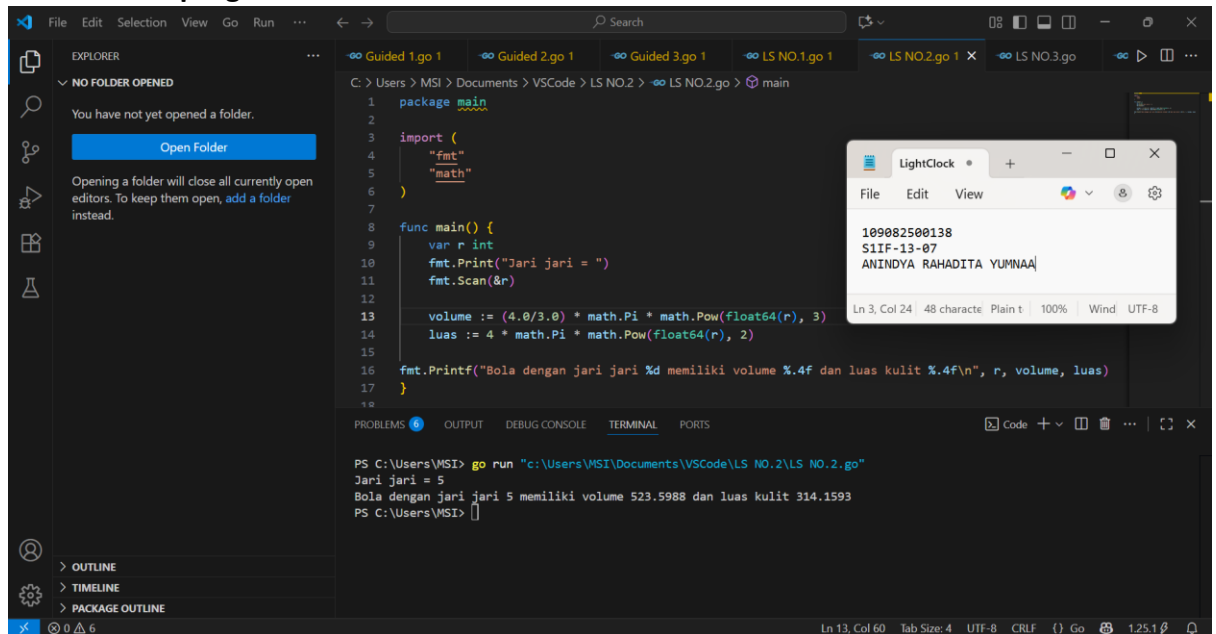
    luas := 4 * math.Pi * math.Pow(float64(r), 2)

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

}

```

## Screenshoot program



## Deskripsi program

### 1. package main

Program ini menandakan bahwa file tersebut adalah **program utama** yang bisa dijalankan (bukan library). Library sendiri itu adalah kode bantu yang dapat berjalan hanya jika dipanggil oleh kode atau program utama, yaitu **package main**. Jika sebuah paket diberi nama main, tool Go akan tahu jika tujuan akhirnya adalah membuat program yang bisa dijalankan.

### 2. import "fmt"

Berfungsi untuk **memasukkan (mengimpor)** dua paket yang berisi fungsi-fungsi siap pakai:

- : Diimpor agar program bisa **berkomunikasi** dengan pengguna, yaitu untuk **menampilkan teks di layar** (, ) dan **menerima input** dari pengguna ().
- : Diimpor agar program dapat **melakukan perhitungan matematika lanjutan**, seperti menggunakan nilai () atau menghitung pangkat ().

### 3. func main() { ... }

func main() adalah **titik masuk** program, kode di dalam func main() dijalankan pertama kali setelah kode atau program diberi nama. Bentuknya harus **“func main()”** tanpa parameter dan tanpa nilai balik.

### 4. var r int

**Mendeklarasikan variabel** bernama **r**. Variabel ini disiapkan dengan tipe data (*integer*), yang berarti hanya dapat menyimpan bilangan bulat (angka tanpa koma). Variabel **r** akan digunakan untuk menampung nilai jari-jari bola.

### 5. fmt.Print("Jari jari = ")

Digunakan untuk **menampilkan instruksi** kepada pengguna di layar. Teks yang muncul adalah **"Jari jari = "**, meminta pengguna memasukkan nilai jari-jari.

### 6. fmt.Scan(&r)

Akan **menangkap input angka** yang dimasukkan oleh pengguna. Angka tersebut kemudian disimpan ke dalam variabel **r**. Simbol (*ampersand*) menunjukkan lokasi memori variabel tempat data akan disimpan.

### 7. volume := (4.0/3.0) \* math.Pi \* math.Pow(float64(r), 3)

**Menghitung volume bola dan mendeklarasikan variabel volume.**

- Operator digunakan untuk mendeklarasikan dan memberikan nilai awal kepada variabel **volume**. Tipenya akan otomatis menjadi karena hasil perhitungannya adalah desimal.
- Perhitungan ini mengikuti rumus volume bola: .
  - memastikan pembagian menghasilkan nilai desimal (bukan bilangan bulat).
  - mengambil nilai konstanta .
  - menghitung (jari-jari pangkat 3). Variabel diubah ke agar fungsi dapat bekerja.
- Hasil perhitungan kemudian **disimpan** ke dalam variabel **volume**.

### 8. luas := 4 \* math.Pi \* math.Pow(float64(r), 2)

**Menghitung luas permukaan bola dan mendeklarasikan variabel luas.**

- Operator mendeklarasikan variabel **luas** (dengan tipe ) dan memberikan nilainya.
- Perhitungan ini mengikuti rumus luas permukaan bola: .
  - menghitung (jari-jari pangkat 2).
- Hasil perhitungan kemudian **disimpan** ke dalam variabel **luas**.

### 9. fmt.Printf("Bola dengan jari jari %d memiliki volume %.4f dan luas kulit %.4f\n", r, volume, luas)

Perintah **menampilkan hasil** di layar dengan format yang spesifik.

- adalah penentu format untuk menampilkan nilai variabel **r** sebagai bilangan bulat.
- adalah penentu format untuk menampilkan nilai variabel **volume** dan **luas** sebagai angka desimal (.) dengan **empat** angka di belakang koma (.).
- adalah kode untuk **membuat baris baru** setelah output selesai.
- Secara berurutan, program menampilkan **r**, **volume** dan **luas**.

### 3. Tugas 3

#### Source code

```
package main

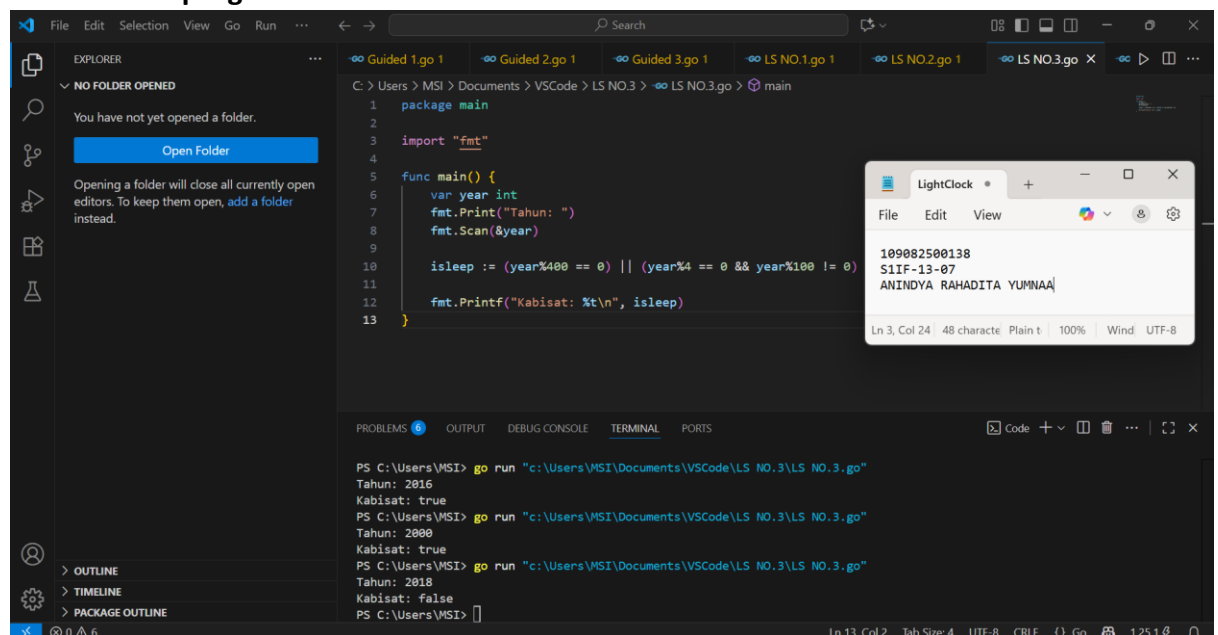
import "fmt"

func main() {
    var year int
    fmt.Print("Tahun: ")
    fmt.Scan(&year)

    isleap := (year%400 == 0) || (year%4 == 0 &&
    year%100 != 0)

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

#### Screenshoot program



## Deskripsi program

### 1. package main

Program ini menandakan bahwa file tersebut adalah **program utama** yang bisa dijalankan (bukan library). Library sendiri itu adalah kode bantu yang dapat berjalan hanya jika dipanggil oleh kode atau program utama, yaitu **package main**. Jika sebuah paket diberi nama main, tool Go akan tahu jika tujuan akhirnya adalah membuat program yang bisa dijalankan.

### 2. import "fmt"

import sendiri berfungsi untuk memanggil kode lain agar fungsinya bisa dipakai di file tersebut. Kemudian, fmt adalah paket standar Go untuk **formatting** dan operasi input atau output sederhana (untuk menulis ke layar dan membaca dari input).

### 3. func main() { ... }

func main() adalah **titik masuk** program, kode di dalam func main() dijalankan pertama kali setelah kode atau program diberi nama. Bentuknya harus **"func main()"** tanpa parameter dan tanpa nilai balik.

### 4. var year int

Perintah ini **mendeklarasikan variabel** bernama **year** (tahun) dengan tipe data (*integer*). Variabel ini disiapkan untuk menyimpan angka tahun yang akan dicek. Tipe digunakan karena tahun selalu berupa bilangan bulat.

### 5. fmt.Print("Tahun: ")

Perintah digunakan untuk **menampilkan instruksi** kepada pengguna di layar. Teks yang muncul adalah **"Tahun: "**, meminta pengguna memasukkan angka tahun.

### 6. fmt.Scan(&year)

Perintah akan **menangkap input angka** tahun yang dimasukkan oleh pengguna. Angka tersebut kemudian disimpan ke dalam variabel **year**. Simbol (*ampersand*) menunjukkan lokasi memori variabel tempat data akan disimpan.

### 7. isleap := (year%400 == 0) || (year%4 == 0 && year%100 != 0)

Ini adalah baris utama yang **melakukan pengecekan** dan **mendeklarasikan variabel isleap**.

- Operator digunakan untuk mendeklarasikan variabel **isleap** dan memberikan nilainya. Tipe datanya akan otomatis menjadi (*boolean*) karena hasilnya berupa Benar () atau Salah ().
- Baris ini menerapkan **aturan tahun kabisat**:
  - : Benar jika tahun habis dibagi 400.
  - : Benar jika tahun habis dibagi 4 **DAN** tidak habis dibagi 100.
- Operator () menyatakan bahwa tahun adalah kabisat jika **salah satu** dari dua kondisi besar di atas terpenuhi.

- Hasil dari seluruh ekspresi logika ini (True atau False) kemudian **disimpan** ke dalam variabel **isleep**.

#### 8. `fmt.Printf("Kabisat: %t\n", isleep)`

**Menampilkan hasil** pengecekan kabisat di layar.

- adalah penentu format yang menampilkan nilai variabel **isleep** sebagai nilai *boolean* (atau).
- adalah kode untuk **membuat baris baru** setelah output selesai. Output yang dihasilkan akan berupa: atau .

### 4. Tugas 4

#### Source code

```
package main

import "fmt"

func main () {
    var celcius float64

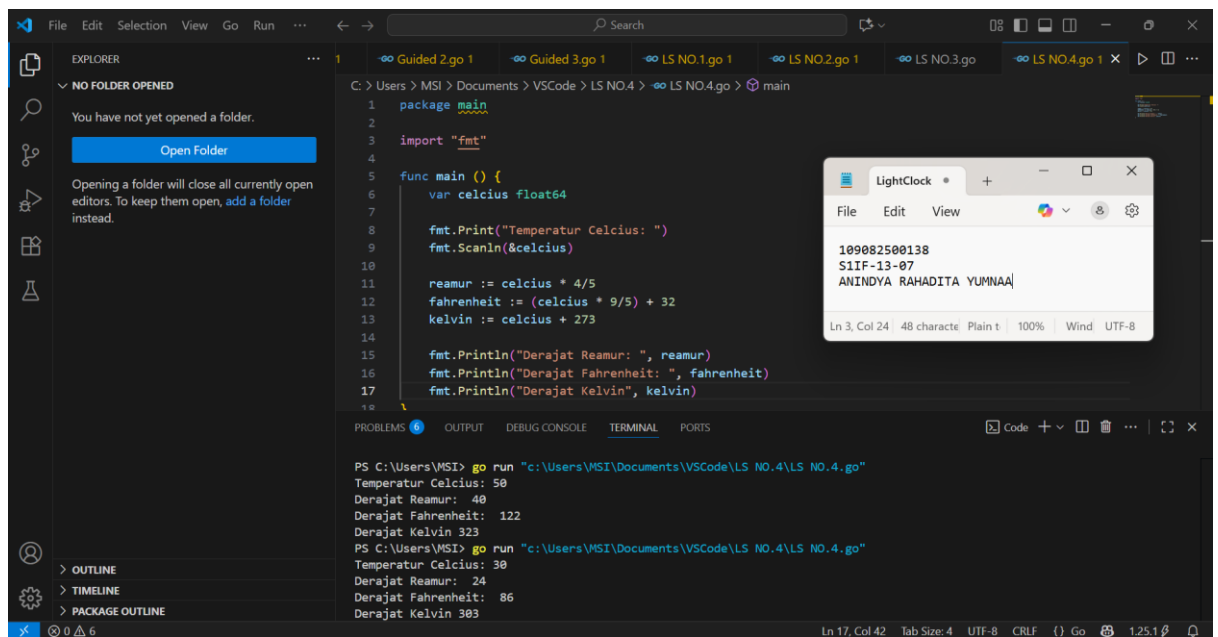
    fmt.Print("Temperatur Celcius: ")
    fmt.Scanln(&celcius)

    reamur := celcius * 4/5
    fahrenheit := (celcius * 9/5) + 32
    kelvin := celcius + 273

    fmt.Println("Derajat Reamur: ", reamur)
    fmt.Println("Derajat Fahrenheit: ", fahrenheit)
    fmt.Println("Derajat Kelvin", kelvin)
}
```

#### Screenshoot program





## Deskripsi program

### 1. package main

Program ini menandakan bahwa file tersebut adalah **program utama** yang bisa dijalankan (bukan library). Library sendiri itu adalah kode bantu yang dapat berjalan hanya jika dipanggil oleh kode atau program utama, yaitu **package main**. Jika sebuah paket diberi nama main, tool Go akan tahu jika tujuan akhirnya adalah membuat program yang bisa dijalankan.

### 2. import "fmt"

import sendiri berfungsi untuk memanggil kode lain agar fungsinya bisa dipakai di file tersebut. Kemudian, fmt adalah paket standar Go untuk **formatting** dan operasi input atau output sederhana (untuk menulis ke layar dan membaca dari input).

### 3. func main() { ... }

func main() adalah **titik masuk** program, kode di dalam func main() dijalankan pertama kali setelah kode atau program diberi nama. Bentuknya harus **"func main()"** tanpa parameter dan tanpa nilai balik.

### 4. var celcius float64

**Mendeklarasikan sebuah variabel** bernama **celcius**. Variabel ini disiapkan dengan tipe data, yang memungkinkannya menyimpan angka suhu desimal (angka berkoma).

### 5. fmt.Print("Temperatur Celcius: ")

Digunakan untuk **menampilkan instruksi** kepada pengguna di layar. Teks yang muncul adalah **"Temperatur Celcius: "**, meminta pengguna memasukkan nilai suhu dalam Celcius.

### 6. fmt.Scanln(&celcius)

**Menangkap input angka** suhu yang dimasukkan oleh pengguna. Angka tersebut kemudian disimpan ke dalam variabel **celcius**. Simbol (*ampersand*) menunjukkan lokasi memori variabel tempat data akan disimpan.

**7. reamur := celcius \* 4/5**

Baris ini **menghitung suhu dalam Reamur** dan **mendeklarasikan variabel reamur**.

- Operator mendeklarasikan variabel **reamur** (dengan tipe ) dan memberikan nilainya.
- Perhitungan ini mengikuti rumus konversi Celcius ke Reamur: **Celcius** .
- Hasil perhitungan kemudian **disimpan** ke dalam variabel **reamur**.

**8. fahrenheit := (celcius \* 9/5) + 32**

**Menghitung suhu dalam Fahrenheit** dan **mendeklarasikan variabel fahrenheit**.

- Operator mendeklarasikan variabel **fahrenheit** (dengan tipe ) dan memberikan nilainya.
- Perhitungan ini mengikuti rumus konversi Celcius ke Fahrenheit: **()** .
- Hasil perhitungan kemudian **disimpan** ke dalam variabel **fahrenheit**.

**9. kelvin := celcius + 273**

**menghitung suhu dalam Kelvin** dan **mendeklarasikan variabel kelvin**.

- Operator mendeklarasikan variabel **kelvin** (dengan tipe ) dan memberikan nilainya.
- Perhitungan ini mengikuti rumus konversi Celcius ke Kelvin: **Celcius** (nilai 273 adalah pembulatan dari 273.15).
- Hasil perhitungan kemudian **disimpan** ke dalam variabel **kelvin**.

**10. fmt.Println("Derajat Reamur: ", reamur)**

**Menampilkan hasil** konversi suhu Reamur. Teks "**Derajat Reamur:**" muncul, diikuti oleh nilai yang tersimpan dalam variabel **reamur**.

**11. fmt.Println("Derajat Fahrenheit: ", fahrenheit)**

**Menampilkan hasil** konversi suhu Fahrenheit. Teks "**Derajat Fahrenheit:**" muncul, diikuti oleh nilai yang tersimpan dalam variabel **fahrenheit**.

**12. fmt.Println("Derajat Kelvin", kelvin)**

**Menampilkan hasil** konversi suhu Kelvin. Teks "**Derajat Kelvin**" muncul, diikuti oleh nilai yang tersimpan dalam variabel **kelvin**.