**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN**

**MODUL 3**

**Sebuah gambar berisi logo, teks, simbol, Grafis

Konten yang dihasilkan AI mungkin salah.**

**Disusun oleh:**

**Farrel Muhamad Julianto**

**109082500148**

**S1IF-13-07**

**Asisten Praktikum**

Adithana dharma putra

Apri pandu wicaksono

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

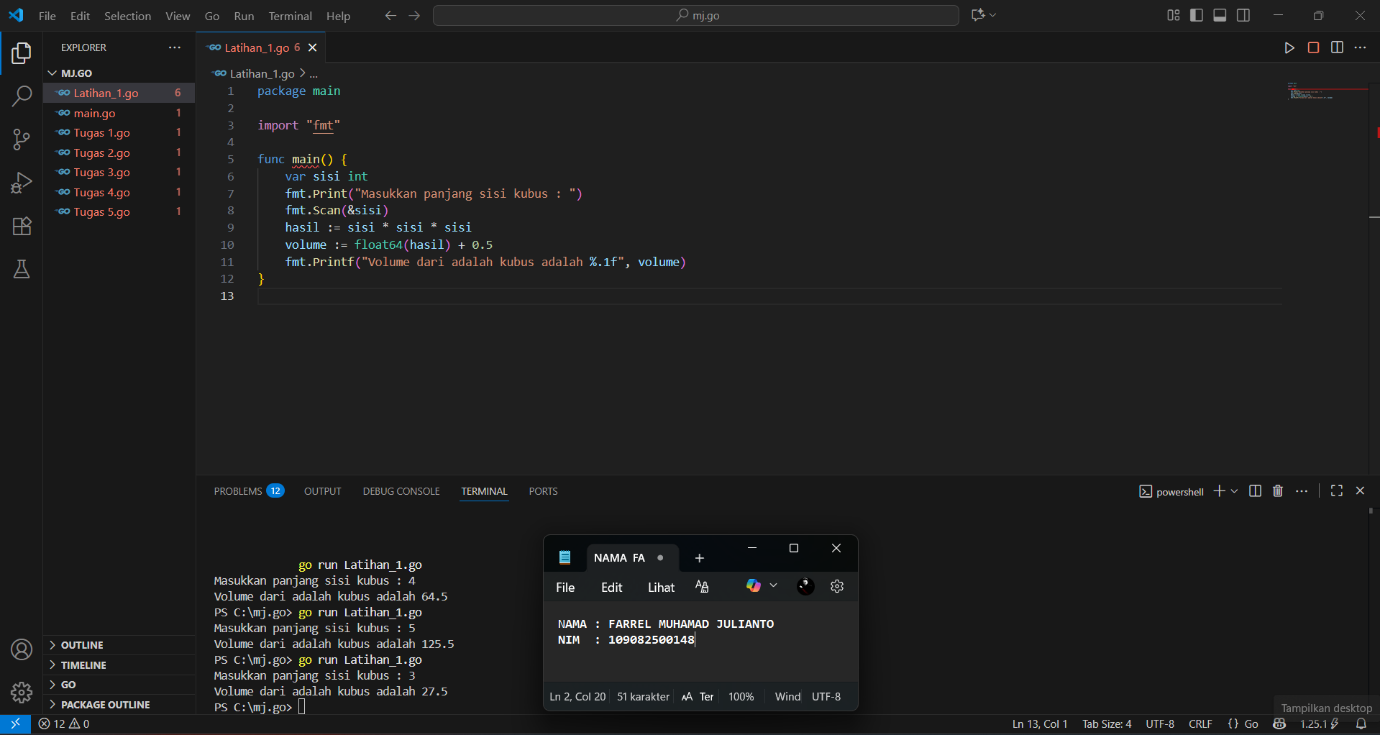
**LATIHAN KELAS – GUIDED**

1. **Guided 1**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var sisi int      fmt.Print("Masukkan panjang sisi kubus : ")      fmt.Scan(&sisi)      hasil := sisi \* sisi \* sisi      volume := float64(hasil) + 0.5      fmt.Printf("Volume dari adalah kubus adalah %.1f", volume)  } |

**Screenshoot program**

****

**Deskripsi program**

Program ini mengambil input berupa bilangan bulat (panjang sisi kubus) dari pengguna, melakukan perhitungan, dan mencetak hasilnya.

**. Input dan Perhitungan Dasar**

**Input Sisi:** Program meminta pengguna untuk memasukkan **panjang sisi kubus** (fmt.Print("Masukkan\npanjang sisi kubus : ")) dan menyimpan nilai bilangan bulat tersebut ke dalam variabel sisi (int).

Perhitungan Volume: Program menghitung volume kubus matematis dengan mengalikan sisi sebanyak tiga kali:

Volume Asli=sisi×sisi×sisi

Hasil perhitungan ini, yang bertipe integer (int), disimpan dalam variabel hasil.

**Modifikasi dan Output Aneh**

Pada bagian ini terjadi penyimpangan dari perhitungan volume kubus standar:

Konversi dan Penambahan Bias: Nilai hasil (volume asli) dikonversi menjadi bilangan desimal (float64) dan kemudian ditambahkan dengan 0.5.

Volume Output=Volume Asli+0.5

Nilai ini disimpan dalam variabel volume.

* + **Contoh:** Jika sisi = 4, maka hasil = 64. Variabel volume akan menjadi.

**Pencetakan Hasil:** Program mencetak nilai volume yang telah dimodifikasi.

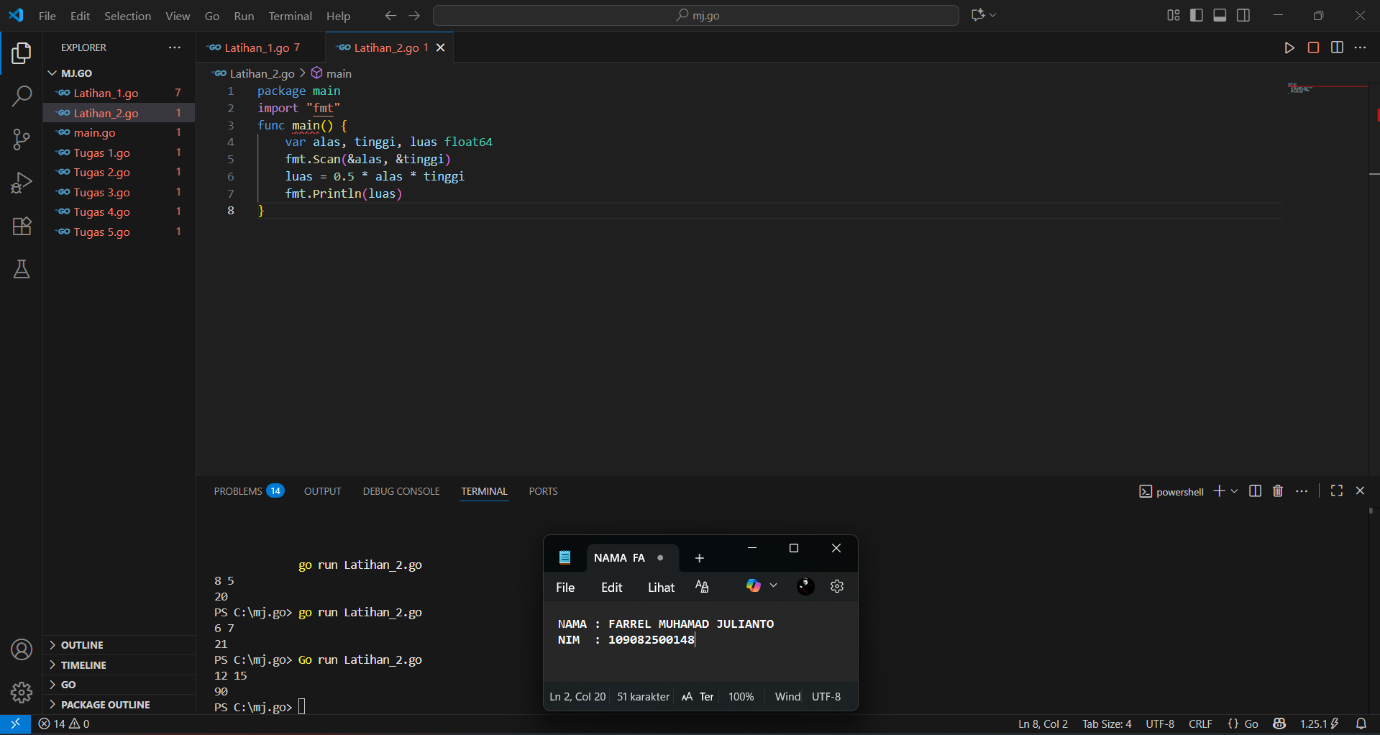
* + fmt.Printf("Volume dari\nadalah kubus adalah %.1f", volume)
  + Format %.1f memastikan bahwa output dicetak sebagai bilangan desimal dengan **satu angka di belakang koma**.

1. **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 ini sangat ringkas dan menjalankan satu tugas inti: menerapkan rumus luas segitiga**

**1. Struktur Dasar**

* **package main: Menunjukkan bahwa ini adalah program yang dapat dieksekusi secara mandiri.**
* **import "fmt": Mengimpor *package* standar Go untuk operasi *input/output* (I/O), yang digunakan di sini untuk menerima masukan dan mencetak keluaran.**
* **func main(): Fungsi utama tempat eksekusi program dimulai.**

**2. Alur Eksekusi**

**Deklarasi Variabel:**

* + **Tiga variabel (alas, tinggi, luas) dideklarasikan, semuanya bertipe float64 (bilangan pecahan). Ini memastikan bahwa input alas dan tinggi dapat berupa bilangan desimal, dan hasil luas juga akan akurat secara desimal.**

**Input Pengguna:**

* + **fmt.Scan(&alas, &tinggi): Program menunggu pengguna untuk memasukkan dua nilai dari *keyboard*.**
  + **Nilai pertama yang dimasukkan akan disimpan di variabel alas.**
  + **Nilai kedua yang dimasukkan akan disimpan di variabel tinggi.**
  + ***Catatan: Program ini tidak mencetak pesan permintaan input (prompt), sehingga pengguna hanya akan melihat kursor berkedip menunggu dua angka dimasukkan, dipisahkan oleh spasi atau baris baru (misalnya: 10 5).***

**Perhitungan Luas:**

* + **luas = 0.5 \* alas \* tinggi: Program menerapkan rumus luas segitiga: Nilai disimpan dalam variabel luas.**

**Output Hasil:**

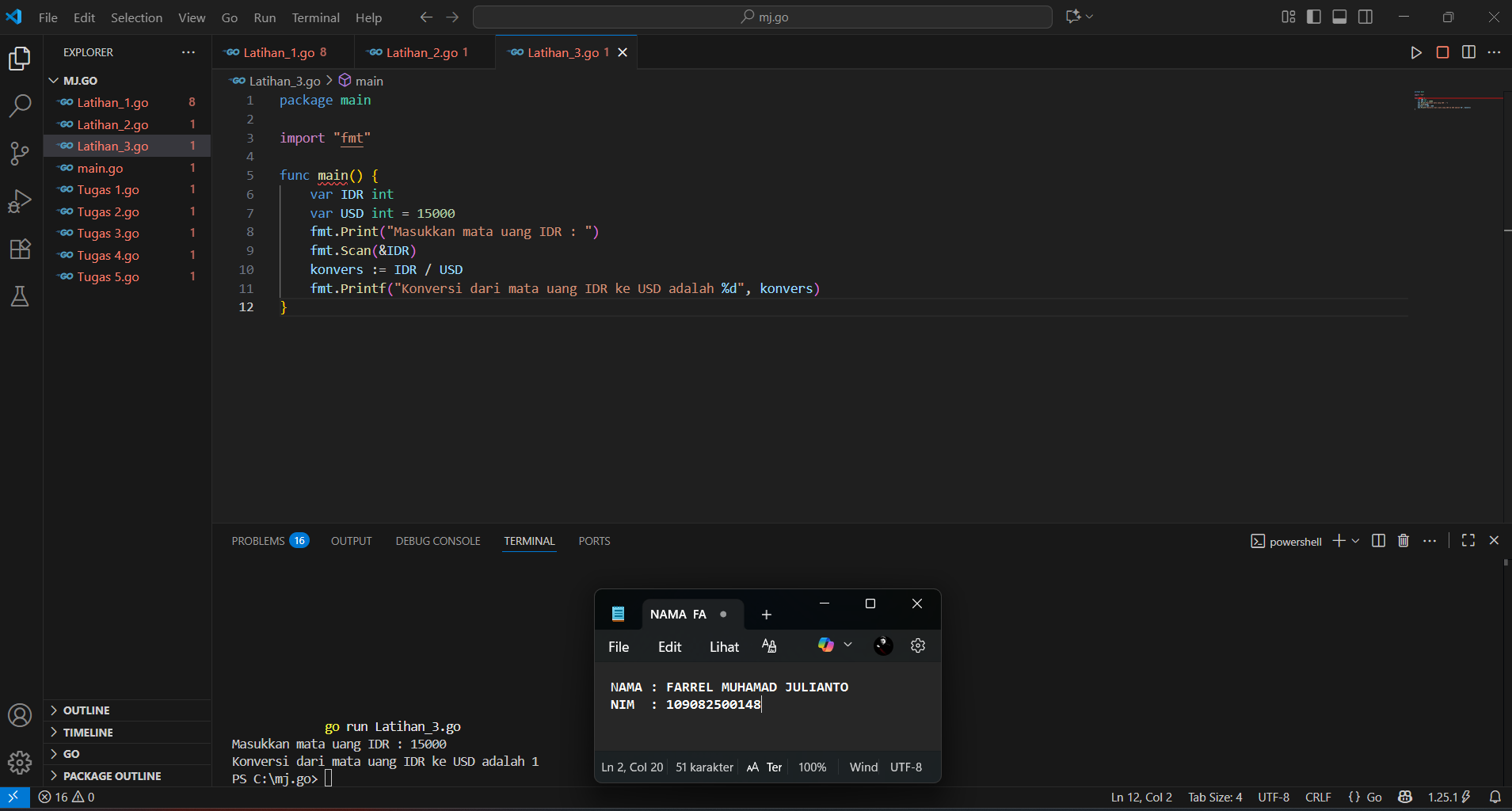
* + **fmt.Println(luas): Program mencetak nilai akhir dari luas segitiga ke konsol, diikuti dengan baris baru.**

1. **Guided 3**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {  var IDR int  var USD int = 15000  fmt.Print("Masukkan mata uang IDR : ")  fmt.Scan(&IDR)  konvers := IDR / USD  fmt.Printf("Konversi dari mata uang IDR ke USD adalah %d", konvers)  } |

**Screenshoot program**

****

**Deskripsi program**

**Program ini fokus pada input dari pengguna dan melakukan pembagian bilangan bulat untuk konversi.**

**1. Deklarasi Variabel**

* **var IDR int: Mendeklarasikan variabel IDR (Rupiah) sebagai bilangan bulat (int). Variabel ini akan menyimpan jumlah uang yang dimasukkan pengguna.**
* **var USD int = 15000: Mendeklarasikan variabel USD (nilai tukar) sebagai bilangan bulat (int) dan langsung memberinya nilai 15000. Ini berarti nilai tukar yang digunakan program adalah Rp 15.000,00 untuk setiap $1 USD.**

**2. Input Pengguna**

* **fmt.Print("Masukkan\nmata uang IDR : "): Program meminta pengguna memasukkan jumlah Rupiah.**
* **fmt.Scan(&IDR): Nilai yang dimasukkan pengguna disimpan ke dalam variabel IDR.**

**3. Logika Konversi**

* **konvers := IDR / USD: Program melakukan konversi dengan membagi jumlah Rupiah yang dimasukkan (IDR) dengan nilai tukar tetap (USD).**

**Hasil USD=Nilai Tukar (15000)Jumlah IDR​**

**Penting: Karena kedua variabel (IDR dan USD) dideklarasikan sebagai int (bilangan bulat), operasi pembagian ini akan menghasilkan pembagian bilangan bulat (integer division). Ini berarti sisa desimal akan diabaikan.**

* + **Contoh: Jika Anda memasukkan IDR = 35000, maka 2, bukan 2.333...**

**4. Output Hasil**

* **fmt.Printf("Konversi\ndari mata uang IDR ke USD adalah %d", konvers): Program mencetak hasil konversi bilangan bulat (konvers) ke layar.**

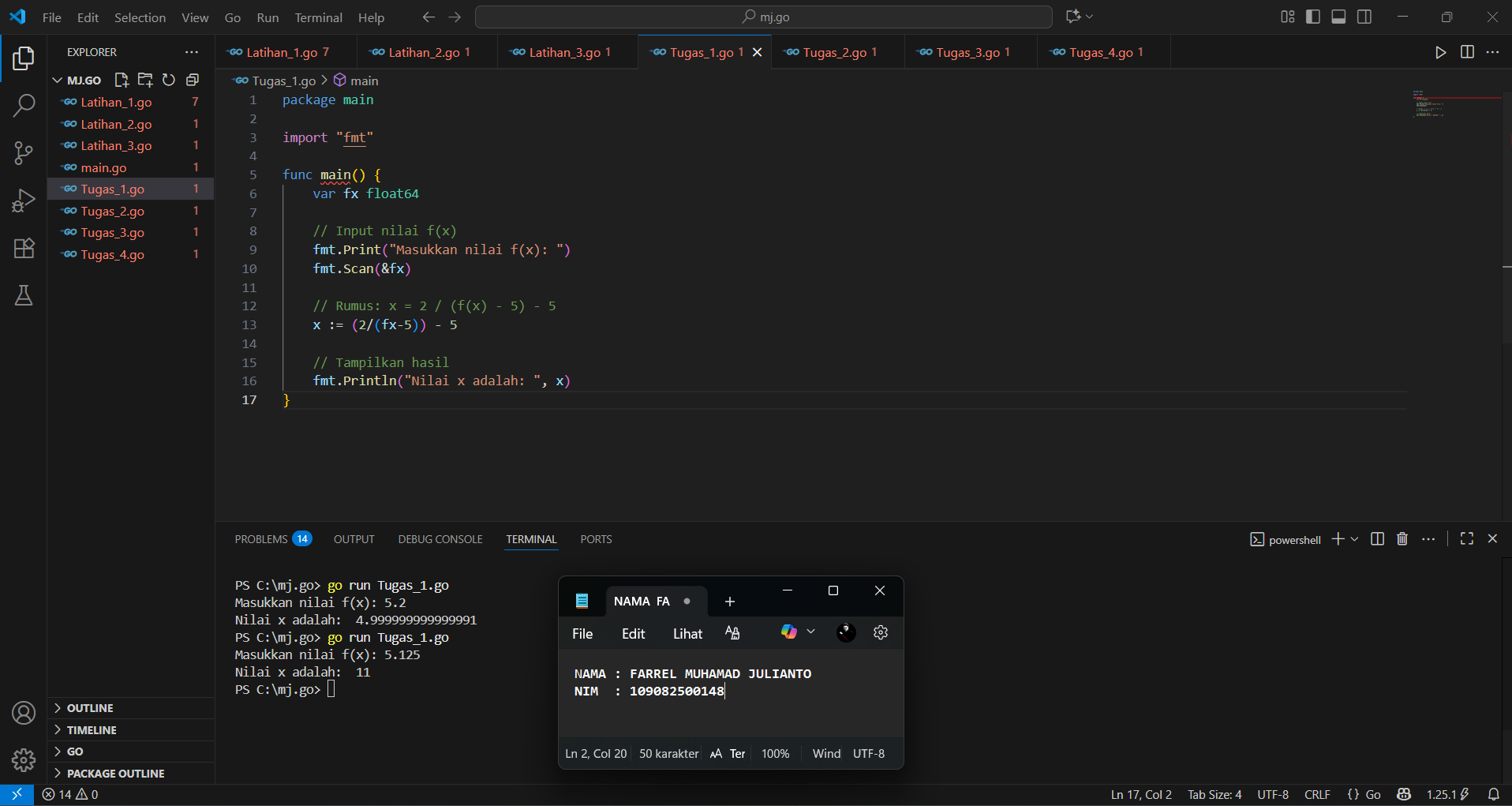
**TUGAS**

1. **Tugas 1**

**Source code**

|  |
| --- |
| package main  import "fmt"  func main() {      var fx float64      // Input nilai f(x)      fmt.Print("Masukkan nilai f(x): ")      fmt.Scan(&fx)      // Rumus: x = 2 / (f(x) - 5) - 5      x := (2/(fx-5)) - 5      // Tampilkan hasil      fmt.Println("Nilai x adalah: ", x)  } |

**Screenshoot program**

****

**Deskripsi program**

**Program ini menjalankan tiga langkah utama: input, perhitungan, dan output.**

**1. Deklarasi dan Input**

* **var fx float64: Mendeklarasikan sebuah variabel bernama fx yang akan menyimpan nilai input. Tipe datanya adalah float64, yang berarti variabel ini dapat menampung angka desimal (bilangan riil) dengan presisi tinggi.**
* **fmt.Print("Masukkan nilai f(x): "): Menampilkan pesan yang meminta pengguna untuk memasukkan angka f(x).**
* **fmt.Scan(&fx): Membaca angka yang diketik pengguna dari *console* dan menyimpannya ke dalam variabel fx.**

**2. Perhitungan Inti (Fungsi Invers)**

* **x := (2/(fx-5)) - 5: Ini adalah bagian inti program. Program menghitung nilai x menggunakan rumus matematika yang diberikan:**

**x=f(x)−52​−5**

**Hasil perhitungan disimpan ke dalam variabel baru bernama x.**

**Secara matematis, rumus ini adalah fungsi invers dari fungsi asalnya f(x)=x+52​+5. Program ini pada dasarnya mencari nilai *input* x yang menghasilkan nilai *output* f(x) tertentu.**

**3. Output**

* **fmt.Println("Nilai x adalah: ", x): Menampilkan hasil akhir perhitungan x kepada pengguna di layar.**

**Batasan (Domain) Matematika**

**Program ini memiliki satu batasan penting yang harus diketahui:**

* **Pembagian dengan Nol: Program akan mengalami masalah jika pengguna memasukkan nilai f(x) yang membuat penyebut menjadi nol, yaitu ketika f(x)−5=0.**

**f(x)=5**

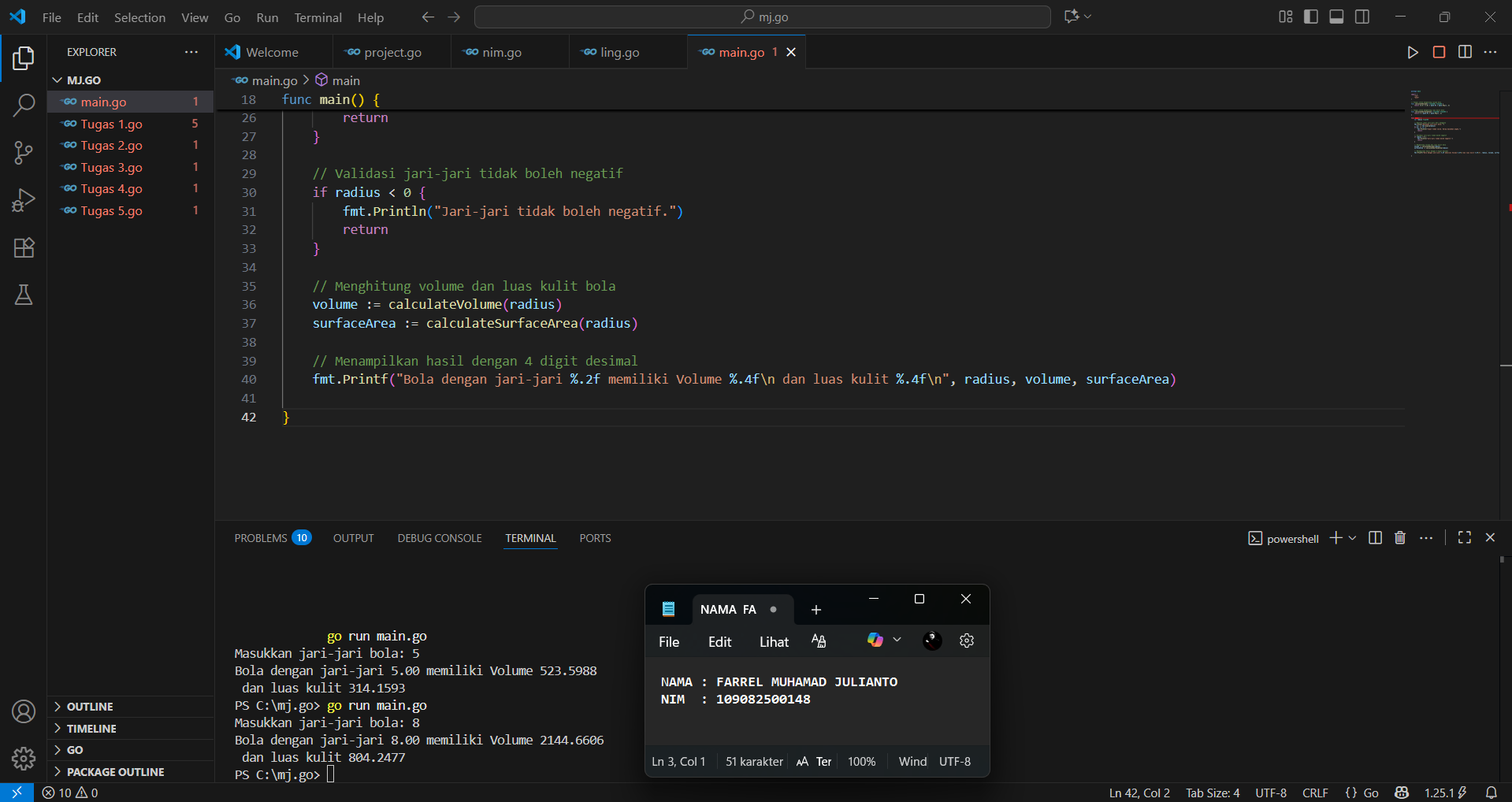
**Jika pengguna memasukkan 5, program akan mencoba membagi 2 dengan 0. Dalam Go untuk tipe float64, ini tidak menyebabkan crash, tetapi akan menghasilkan nilai +Inf (tak terhingga), yang merupakan representasi dari hasil operasi tersebut.**

1. **Tugas 2**

**Source code**

|  |
| --- |
| package main  import (  "fmt"  "math"  )  // Fungsi untuk menghitung volume bola  func calculateVolume(r float64) float64 {  return (4.0 / 3.0) \* math.Pi \* math.Pow(r, 3)  }  // Fungsi untuk menghitung luas kulit bola  func calculateSurfaceArea(r float64) float64 {  return 4 \* math.Pi \* math.Pow(r, 2)  }  func main() {  var radius float64  // Meminta input jari-jari dari pengguna  fmt.Print("Masukkan jari-jari bola: ")  \_, err := fmt.Scan(&radius)  if err != nil {  fmt.Println("Input tidak valid. Harap masukkan angka.")  return  }  // Validasi jari-jari tidak boleh negatif  if radius < 0 {  fmt.Println("Jari-jari tidak boleh negatif.")  return  }  // Menghitung volume dan luas kulit bola  volume := calculateVolume(radius)  surfaceArea := calculateSurfaceArea(radius)  // Menampilkan hasil dengan 4 digit desimal  fmt.Printf("Bola dengan jari-jari %.2f memiliki Volume %.4f\n dan luas kulit %.4f\n", radius, volume, surfaceArea)  } |

**Screenshoot program**

****

**Deskripsi program**

**Program ini menggunakan dua fungsi khusus untuk perhitungan dan satu fungsi utama (main) untuk interaksi dengan pengguna.**

**1. Import Packages**

**Program ini mengimpor dua *package* standar Go:**

* **fmt: Digunakan untuk operasi input dan output, seperti meminta input (fmt.Print, fmt.Scan) dan mencetak hasil (fmt.Printf).**
* **math: Menyediakan konstanta matematika seperti (math.Pi) dan fungsi-fungsi matematika seperti perpangkatan (math.Pow).**

**2. Fungsi Perhitungan**

**Program mendefinisikan dua fungsi untuk melakukan perhitungan geometri bola:**

**A. calculateVolume(r float64) float64**

* **Tujuan: Menghitung volume bola.**
* **Argumen: Menerima r (jari-jari) bertipe float64 (bilangan desimal).**
* **Rumus: Mengembalikan hasil dari rumus Volume Bola: .**
  + **Dalam kode: (4.0 / 3.0) \* math.Pi \* math.Pow(r, 3)**

**B. calculateSurfaceArea(r float64) float64**

* **Tujuan: Menghitung luas permukaan (kulit) bola.**
* **Argumen: Menerima r (jari-jari) bertipe float64.**
* **Rumus: Mengembalikan hasil dari rumus Luas Permukaan Bola: .**
  + **Dalam kode: 4 \* math.Pi \* math.Pow(r, 2)**

**3. Fungsi Utama (main)**

**Ini adalah alur utama eksekusi program:**

1. **Deklarasi Variabel: Mendeklarasikan variabel radius bertipe float64 untuk menyimpan input jari-jari.**
2. **Input Pengguna: Program meminta pengguna untuk "Masukkan jari-jari bola". Nilai yang dimasukkan disimpan ke dalam variabel radius menggunakan fmt.Scan.**
3. **Validasi Input:**
   * **Cek Tipe Data: Memeriksa apakah input valid sebagai angka (if err != nil). Jika tidak, program menampilkan pesan error dan berhenti.**
   * **Cek Nilai: Memeriksa apakah jari-jari tidak negatif (if radius < 0). Jika negatif, program menampilkan pesan peringatan dan berhenti.**
4. **Perhitungan:**
   * **Program memanggil calculateVolume(radius) dan menyimpan hasilnya di variabel volume.**
   * **Program memanggil calculateSurfaceArea(radius) dan menyimpan hasilnya di variabel surfaceArea.**
5. **Output Hasil: Program mencetak hasil akhir ke layar.**
   * **%.2f: Digunakan untuk mencetak nilai jari-jari dengan dua angka di belakang koma.**
   * **%.4f: Digunakan untuk mencetak nilai volume dan luas kulit dengan empat angka di belakang koma agar lebih akurat.**

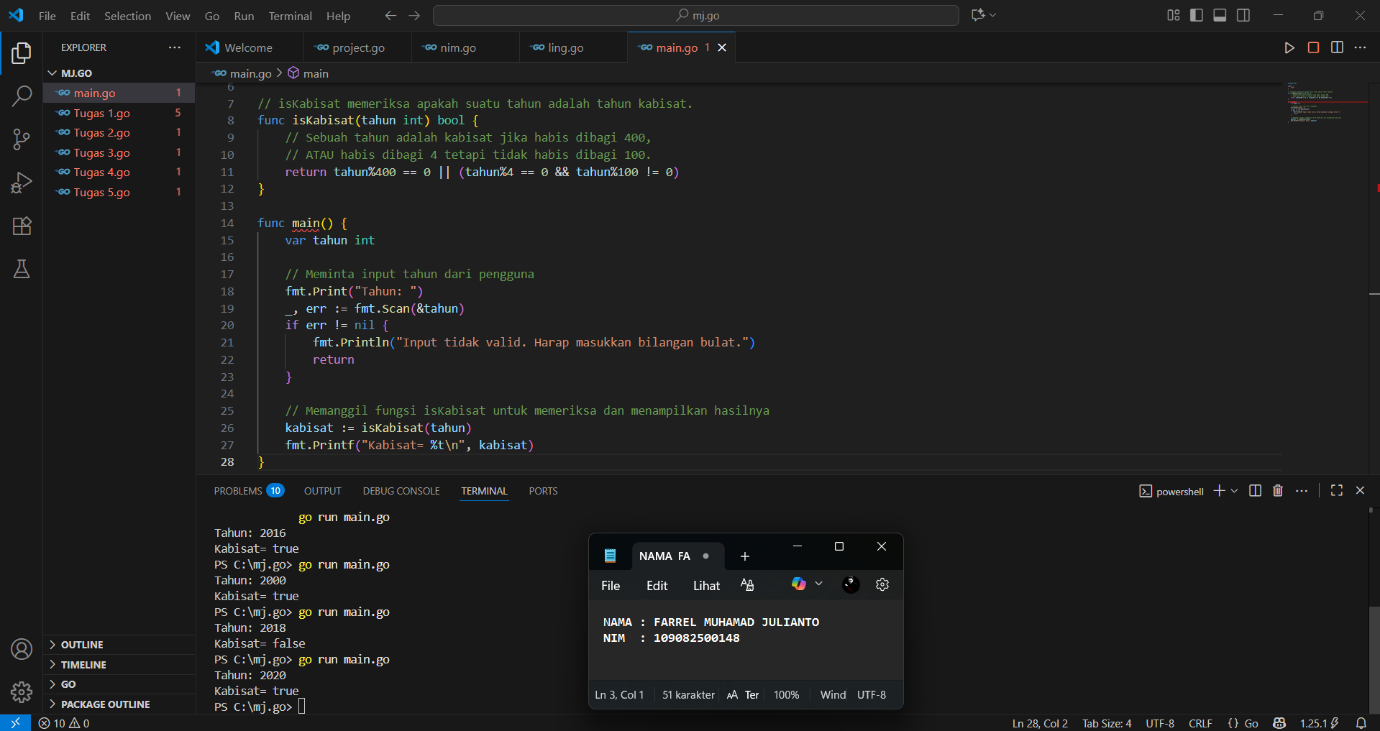
**Secara keseluruhan, program ini memastikan input yang valid dan kemudian menerapkan rumus matematika yang benar untuk memberikan hasil perhitungan volume dan luas permukaan bola.**

1. **Tugas 3**

**Source code**

|  |
| --- |
| package main  import (  "fmt"  )  // isKabisat memeriksa apakah suatu tahun adalah tahun kabisat.  func isKabisat(tahun int) bool {  // Sebuah tahun adalah kabisat jika habis dibagi 400,  // ATAU habis dibagi 4 tetapi tidak habis dibagi 100.  return tahun%400 == 0 || (tahun%4 == 0 && tahun%100 != 0)  }  func main() {  var tahun int  // Meminta input tahun dari pengguna  fmt.Print("Tahun: ")  \_, err := fmt.Scan(&tahun)  if err != nil {  fmt.Println("Input tidak valid. Harap masukkan bilangan bulat.")  return  }  // Memanggil fungsi isKabisat untuk memeriksa dan menampilkan hasilnya  kabisat := isKabisat(tahun)  fmt.Printf("Kabisat= %t\n", kabisat)  } |

**Screenshoot program**

****

**Deskripsi program**

Program ini sangat lugas, berfokus pada implementasi logika standar untuk pengecekan tahun kabisat dalam sebuah fungsi khusus.

**1. Fungsi isKabisat(tahun int) bool**

Ini adalah inti dari program, yang menerapkan aturan Tahun Kabisat:

* **Tujuan:** Menerima sebuah bilangan bulat (int) sebagai tahun dan mengembalikan nilai boolean (bool), yaitu true jika itu tahun kabisat, dan false jika bukan.
* **Logika Aturan Kabisat:** Tahun kabisat memiliki dua kondisi utama, yang digabungkan dengan operator **OR (||)**:
  1. Tahun **habis dibagi 400** (tahun%400 == 0). *Contoh: Tahun 2000.* **ATAU**
  2. Tahun **habis dibagi 4** (tahun%4 == 0) **DAN** **tidak habis dibagi 100** (tahun%100 != 0). *Contoh: Tahun 2024, 1996.*
* **Pengecualian 100:** Pengecekan tahun%100 != 0 penting karena tahun-tahun yang habis dibagi 100 (seperti 1900, 2100) *bukan* tahun kabisat, kecuali jika mereka juga habis dibagi 400.

**2. Fungsi Utama (main)**

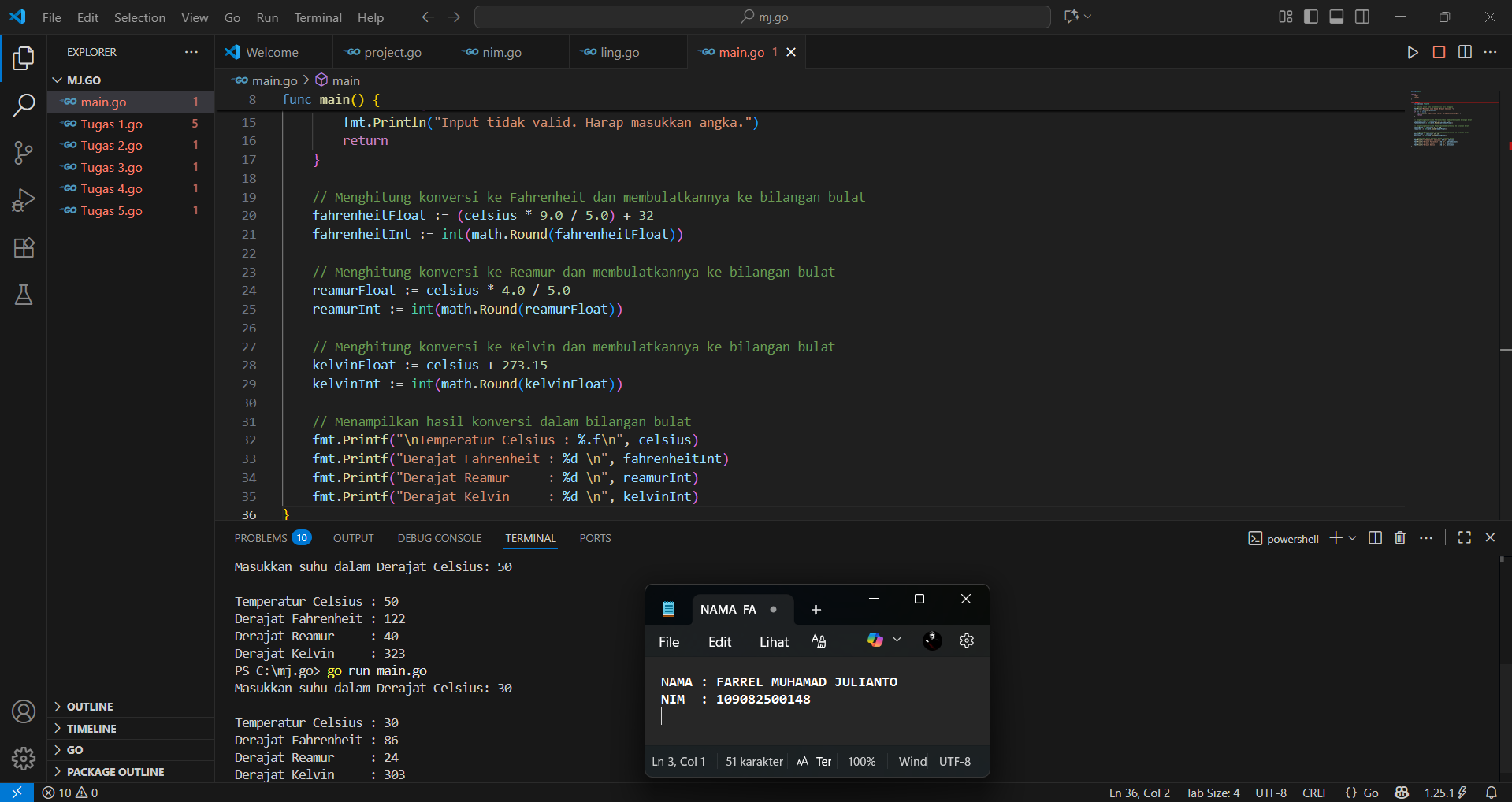
Fungsi ini menangani interaksi pengguna dan menampilkan hasil:

1. **Input Pengguna:** Program meminta pengguna untuk memasukkan sebuah tahun (fmt.Print("Tahun: ")).
2. **Validasi Input:** Program memeriksa apakah input yang diberikan valid sebagai bilangan bulat (int). Jika tidak, ia mencetak pesan kesalahan dan menghentikan eksekusi.
3. **Pemanggilan Fungsi:** Nilai tahun yang valid diteruskan ke fungsi isKabisat(). Hasil boolean (true atau false) disimpan di variabel kabisat.
4. **Output Hasil:** Program mencetak hasilnya dalam format: "Kabisat= [true/false]". (%t digunakan untuk menampilkan nilai boolean).
5. **Tugas 3**

**Source code**

|  |
| --- |
| package main  import (  "fmt"  "math"  )  func main() {  var celsius float64  // Meminta input suhu dalam Celcius dari pengguna  fmt.Print("Masukkan suhu dalam Derajat Celsius: ")  \_, err := fmt.Scan(&celsius)  if err != nil {  fmt.Println("Input tidak valid. Harap masukkan angka.")  return  }  // Menghitung konversi ke Fahrenheit dan membulatkannya ke bilangan bulat  fahrenheitFloat := (celsius \* 9.0 / 5.0) + 32  fahrenheitInt := int(math.Round(fahrenheitFloat))  // Menghitung konversi ke Reamur dan membulatkannya ke bilangan bulat  reamurFloat := celsius \* 4.0 / 5.0  reamurInt := int(math.Round(reamurFloat))  // Menghitung konversi ke Kelvin dan membulatkannya ke bilangan bulat  kelvinFloat := celsius + 273.15  kelvinInt := int(math.Round(kelvinFloat))  // Menampilkan hasil konversi dalam bilangan bulat  fmt.Printf("\nTemperatur Celsius : %.f\n", celsius)  fmt.Printf("Derajat Fahrenheit : %d \n", fahrenheitInt)  fmt.Printf("Derajat Reamur : %d \n", reamurInt)  fmt.Printf("Derajat Kelvin : %d \n", kelvinInt)  } |

**Screenshoot program**

****

**Deskripsi program**

**1. Tujuan Utama**

Tujuan program ini adalah menghitung dan menampilkan hasil konversi dari Celsius ke Fahrenheit, Reamur, dan Kelvin. Semua perhitungan dilakukan dengan bilangan desimal (float64) untuk akurasi, tetapi hasil akhir ditampilkan sebagai bilangan bulat (int).

**2. Alur Program Utama (main)**

1. **Input Pengguna:** Program meminta pengguna untuk **memasukkan suhu dalam Derajat Celsius** dan menyimpannya di variabel celsius bertipe float64.
2. **Validasi Input:** Program memeriksa apakah input yang diberikan valid sebagai angka. Jika input tidak valid (misalnya, teks), ia mencetak pesan kesalahan dan berhenti.

**3. Logika Konversi Suhu**

Setelah mendapatkan input Celsius yang valid, program melakukan tiga perhitungan terpisah menggunakan rumus konversi standar:

| Skala Target | Rumus Konversi (dari ) | Implementasi Kode | Pembulatan |
| --- | --- | --- | --- |
| **Fahrenheit** |  | (celsius \* 9.0 / 5.0) + 32 | Hasilnya dibulatkan ke bilangan bulat terdekat menggunakan math.Round(). |
| **Reamur** |  | celsius \* 4.0 / 5.0 | Hasilnya dibulatkan ke bilangan bulat terdekat menggunakan math.Round(). |
| **Kelvin** |  | celsius + 273.15 | Hasilnya dibulatkan ke bilangan bulat terdekat menggunakan math.Round(). |

**Penting:** Penggunaan math.Round() memastikan hasil konversi seperti akan menjadi, dan akan menjadi . Hasil pembulatan ini kemudian diubah menjadi tipe data int.

**4. Output Hasil**

Program mencetak input Celsius asli (yang mungkin ditampilkan dengan nol desimal karena format %.f) dan kemudian menampilkan hasil konversi **bilangan bulat** untuk Fahrenheit, Reamur, dan Kelvin.