

**LAPORAN PRAKTIKUM ALGORITMA  
DAN PEMROGRAMAN 1**

**MODUL 3  
TIPE DATA DAN VARIABEL**



**Disusun oleh:**

**AHMAD MALIK ARRAYYAN**

**109082500116**

**S1IF-13-02**

**Asisten Praktikum**

Adithana dharma putra

Alfin Ilham Berlianto

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## LATIHAN KELAS – GUIDED

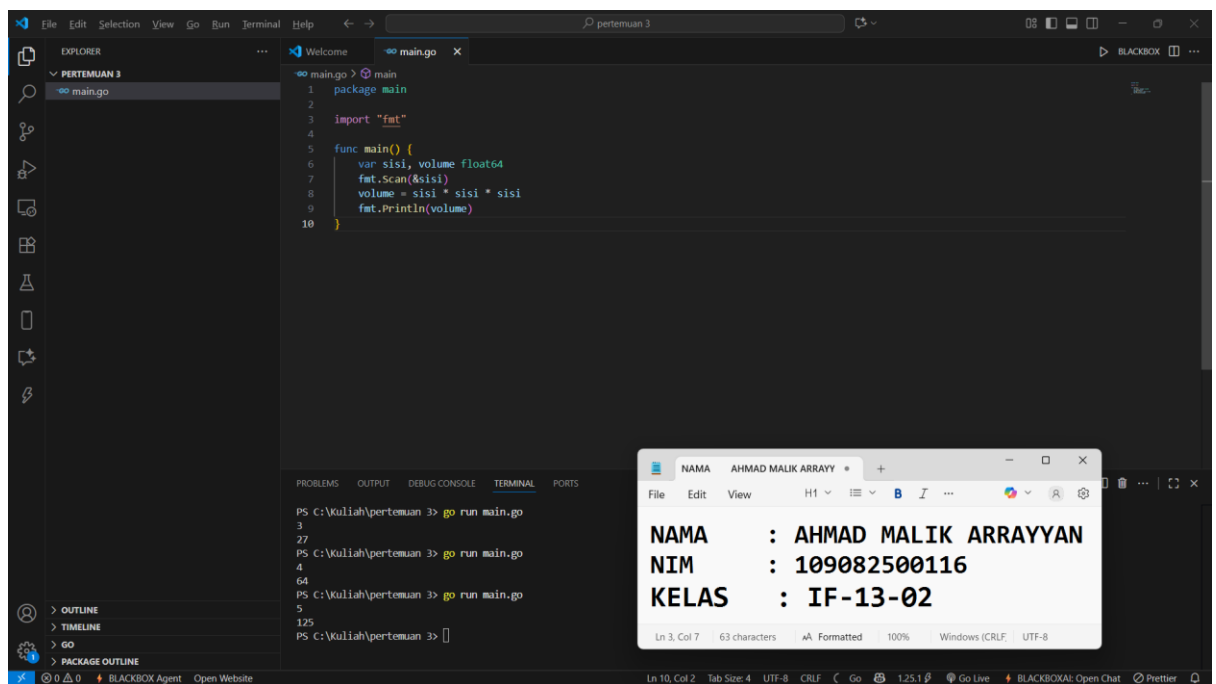
### 1. Guided 1 Source Code

```
package main

import "fmt"

func main() {
    var sisi, volume float64
    fmt.Scan(&sisi)
    volume = sisi * sisi * sisi
    fmt.Println(volume)
}
```

### Screenshoot program



### Deskripsi program

Jelaskan kode yang ada di source code, semakin detal semakin baik nilainya

## 2. Guided 2

### Source Code

```
package main

import "fmt"

func main() {

    var alas, tinggi, luas float64

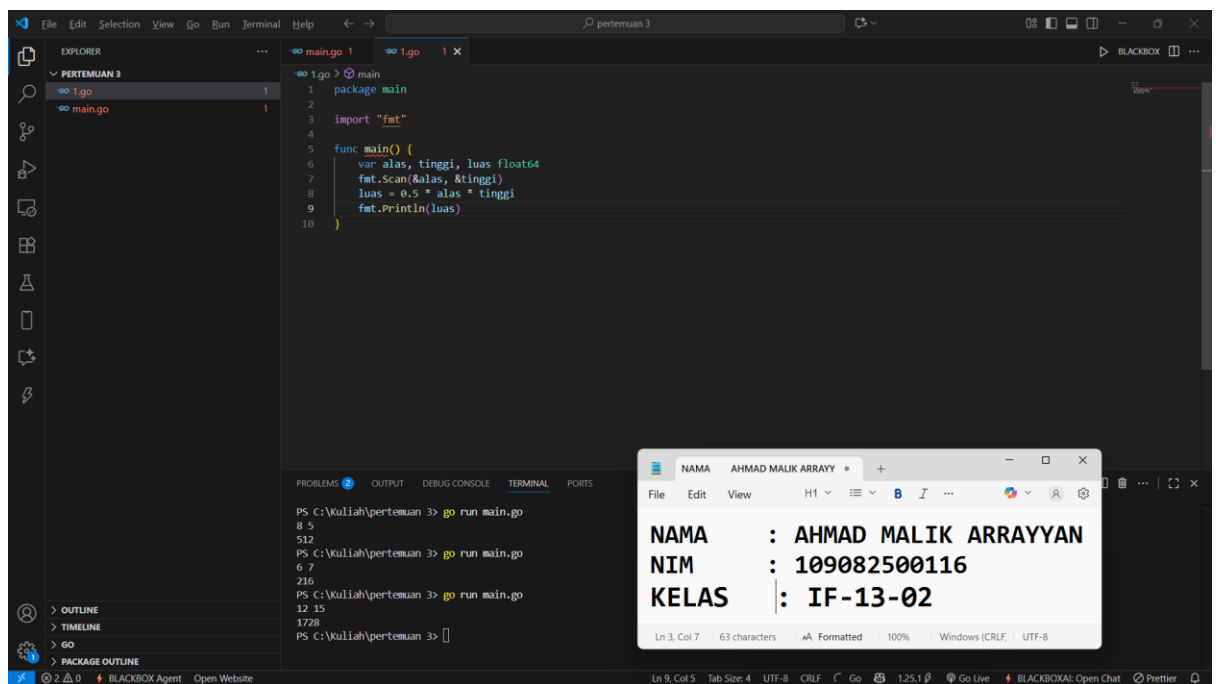
    fmt.Scan(&alas, &tinggi)

    luas = 0.5 * alas * tinggi

    fmt.Println(luas)

}
```

### Screenshoot program



### Deskripsi program

### 3. Guided 3

#### Source Code

```
package main

import "fmt"

func main() {

    var rupiah, dolar int

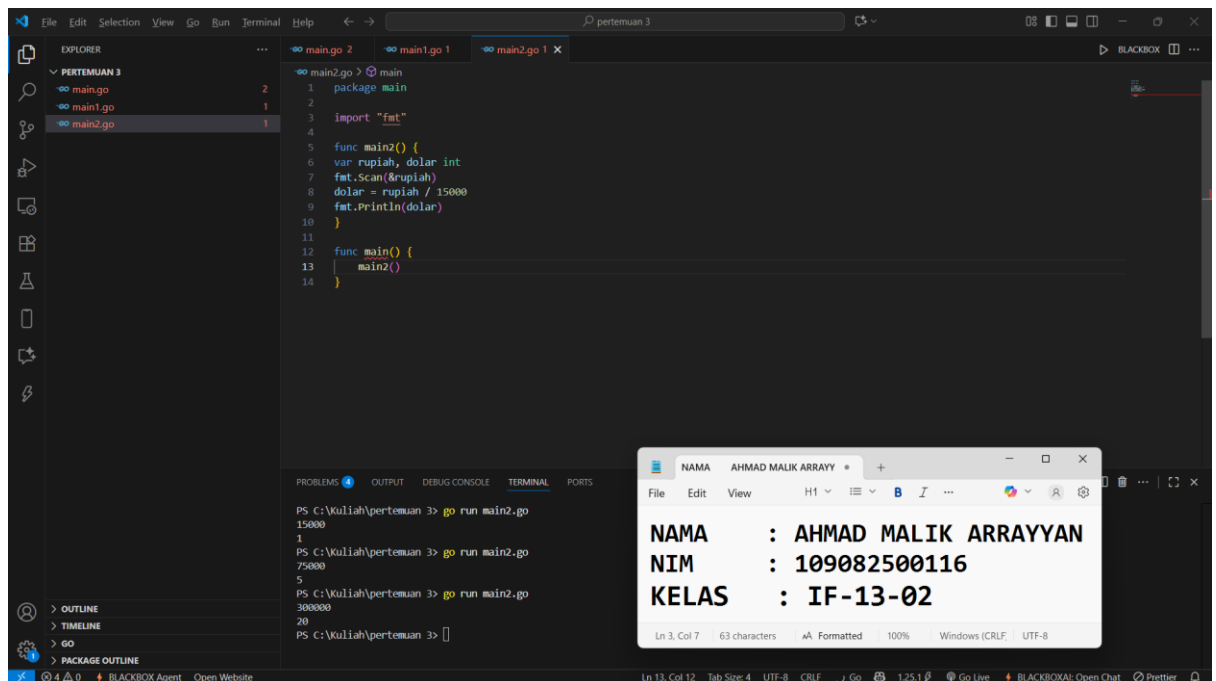
    fmt.Scan(&rupiah)

    dolar = rupiah / 15000

    fmt.Println(dolar)

}
```

#### Screenshoot program



#### Deskripsi program

Program Anda membaca jumlah Rupiah, membaginya dengan 15.000, lalu mencetak hasil sebagai bilangan bulat—artinya potongan (sisa pembagian) diabaikan; jika Anda ingin hasil desimal, ubah tipe ke float64 dan lakukan pembulatan/format yang sesuai.

## TUGAS

### 1. Tugas 1

#### Source code

```
package main

import (
    "fmt"
    "math"
)

// Fungsi untuk menghitung nilai x dari persamaan  $f(x) = 2/(x+5) + 5$ 
// diberikan nilai f(x) (fx)
func calculateX(fx float64) (float64, error) {
    // Persamaan:  $fx = 2/(x+5) + 5$ 
    // Maka:  $fx - 5 = 2/(x+5)$ 
    //  $(x+5) = 2 / (fx-5)$ 
    //  $x = (2 / (fx-5)) - 5$ 
    // Cek pembagi tidak boleh nol
    if fx == 5 {
        return 0, fmt.Errorf("nilai f(x) tidak boleh sama dengan 5 (pembagian oleh nol)")
    }
    x := (2 / (fx - 5)) - 5
    return x, nil
}

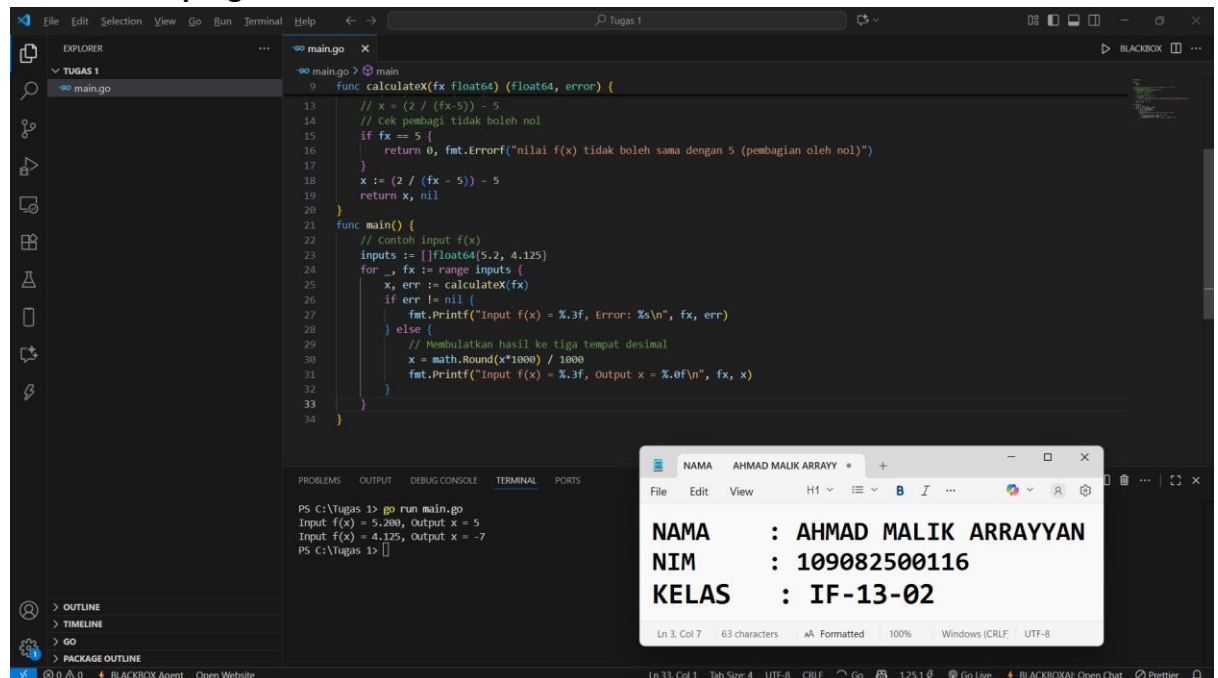
func main() {
    // Contoh input f(x)
    inputs := []float64{5.2, 4.125}
    for _, fx := range inputs {
        x, err := calculateX(fx)
        if err != nil {
```

```

        fmt.Printf("Input f(x) = %.3f, Error: %s\n", fx, err)
    } else {
        // Membulatkan hasil ke tiga tempat desimal
        x = math.Round(x*1000) / 1000
        fmt.Printf("Input f(x) = %.3f, Output x = %.0f\n", fx,
x)
    }
}
}
}

```

## Screenshoot program



## Deskripsi program

- `func calculateX(fx float64) (float64, error)`  
Fungsi menerima satu angka `fx` (jenis `float64`) dan mengembalikan dua nilai: hasil `x` (`float64`) atau sebuah error bila ada kondisi yang tidak boleh (pembagian oleh nol).
- `if fx == 5 { ... }`  
Cek sederhana: kalau `fx` persis sama dengan 5, fungsi mengembalikan error. Kenapa? Karena rumus punya pembagian  $2/(fx-5)$  — kalau `fx == 5`, penyebutnya nol → pembagian oleh nol tidak boleh.
- `x := (2 / (fx - 5)) - 5`  
Hitungan utama: bagi 2 dengan  $(fx-5)$ , lalu kurangi 5 — itu rumus yang ingin dihitung.
- Di main: `inputs := []float64{5.2, 4.125}` adalah dua contoh nilai `f(x)` yang akan diuji.

- Loop for `_`, `fx := range inputs { ... }` memproses setiap input: memanggil `calculateX`, cek apakah ada `err`, kalau ada cetak error; kalau tidak, lakukan pembulatan ke 3 desimal lalu cetak hasil.
- `x = math.Round(x*1000) / 1000`  
Teknik umum untuk membulatkan ke 3 desimal: kalikan 1000, Round ke integer terdekat, lalu bagi 1000 lagi.
- `fmt.Printf(..., Output x = %.0f, fx, x)`  
Perhatikan: format `%.0f` mencetak angka sebagai bulat (nol tempat desimal). Jadi walau sebelumnya Anda sudah membulatkan ke 3 desimal, format ini akan menampilkan angka sebagai integer (dengan pembulatan ke integer terdekat saat formatting).

## 2. Tugas 2

### Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    const pi = 3.1415926535

    var jari int
    fmt.Print("jari = ")
    fmt.Scan(&jari)

    volume := (4.0 / 3.0) * pi * math.Pow(float64(jari), 3)
    luas := 4 * pi * math.Pow(float64(jari), 2)

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

```
}
```

## Screenshoot program

```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     const pi = 3.1415926535
10
11     var jari int
12     fmt.Print("jari = ")
13     fmt.Scan(&jari)
14
15     volume := (4.0 / 3.0) * pi * math.Pow(float64(jari), 3)
16     luas := 4 * pi * math.Pow(float64(jari), 2)
17     var volume float64
18     fmt.Printf("Bola dengan jari %d memiliki volume %.4f dan luas kulit %.4f\n", jari, volume, luas)
19 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Tugas 2> go run main.go
Jari = 2
Bola dengan jari 2 memiliki volume 33.5103 dan luas kulit 50.2655
PS C:\Tugas 2> go run main.go
jari = 67
Bola dengan jari 67 memiliki volume 1259833.1883 dan luas kulit 56410.4377
PS C:\Tugas 2>
```

## Deskripsi program

- `const pi = 3.1415926535`  
Menyimpan nilai  $\pi$  sebagai konstanta. (Bisa juga pakai `math.Pi` yang sudah tersedia di paket `math`.)
- `var jari int`  
Menyiapkan variabel jari bertipe integer untuk menampung input radius dari user.
- `fmt.Print("jari = ")` lalu `fmt.Scan(&jari)`  
Menampilkan prompt sederhana lalu membaca input dari keyboard. `&jari` berarti kita memberi alamat variabel agar `Scan` bisa mengisinya. (Catatan: kode ini tidak memeriksa error dari `fmt.Scan`.)
- `volume := (4.0 / 3.0) * pi * math.Pow(float64(jari), 3)`
  1. `4.0 / 3.0` memastikan pembagian dilakukan dalam angka desimal (float).
  2. `math.Pow(float64(jari), 3)` mengangkat jari ke pangkat 3, tapi karena `math.Pow` menerima `float64`, kita konversi jari (int) ke `float64`.
  3. Hasilnya tipe `float64` disimpan di `volume`.
- `luas := 4 * pi * math.Pow(float64(jari), 2)`  
Sama ide, tetapi pangkat 2 untuk luas permukaan.
- `fmt.Printf(..., jari, volume, luas)`  
Menampilkan teks lengkap; `%d` untuk integer jari, `%.4f` untuk menampilkan volume dan luas dengan 4 angka di belakang koma.

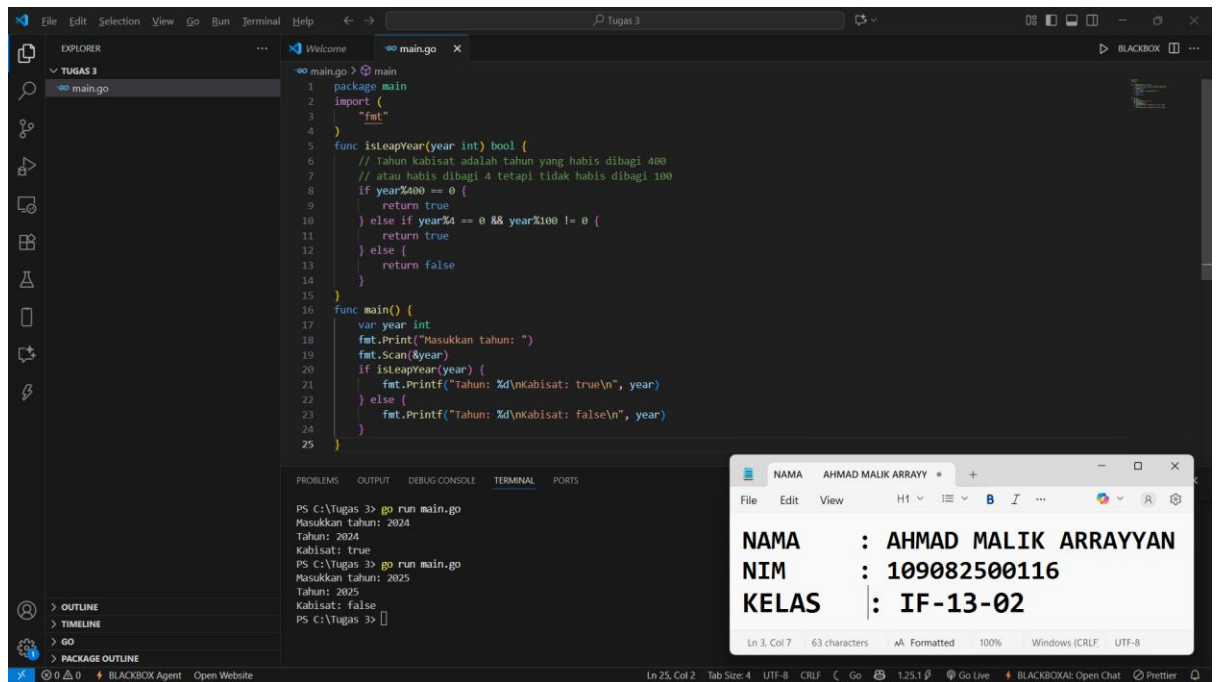


### 3. Tugas 3

#### Source code

```
package main
import (
    "fmt"
)
func isLeapYear(year int) bool {
    // Tahun kabisat adalah tahun yang habis dibagi 400
    // atau habis dibagi 4 tetapi tidak habis dibagi 100
    if year%400 == 0 {
        return true
    } else if year%4 == 0 && year%100 != 0 {
        return true
    } else {
        return false
    }
}
func main() {
    var year int
    fmt.Print("Masukkan tahun: ")
    fmt.Scan(&year)
    if isLeapYear(year) {
        fmt.Printf("Tahun: %d\nKabisat: true\n", year)
    } else {
        fmt.Printf("Tahun: %d\nKabisat: false\n", year)
    }
}
```

#### Screenshoot program



### Deskripsi program

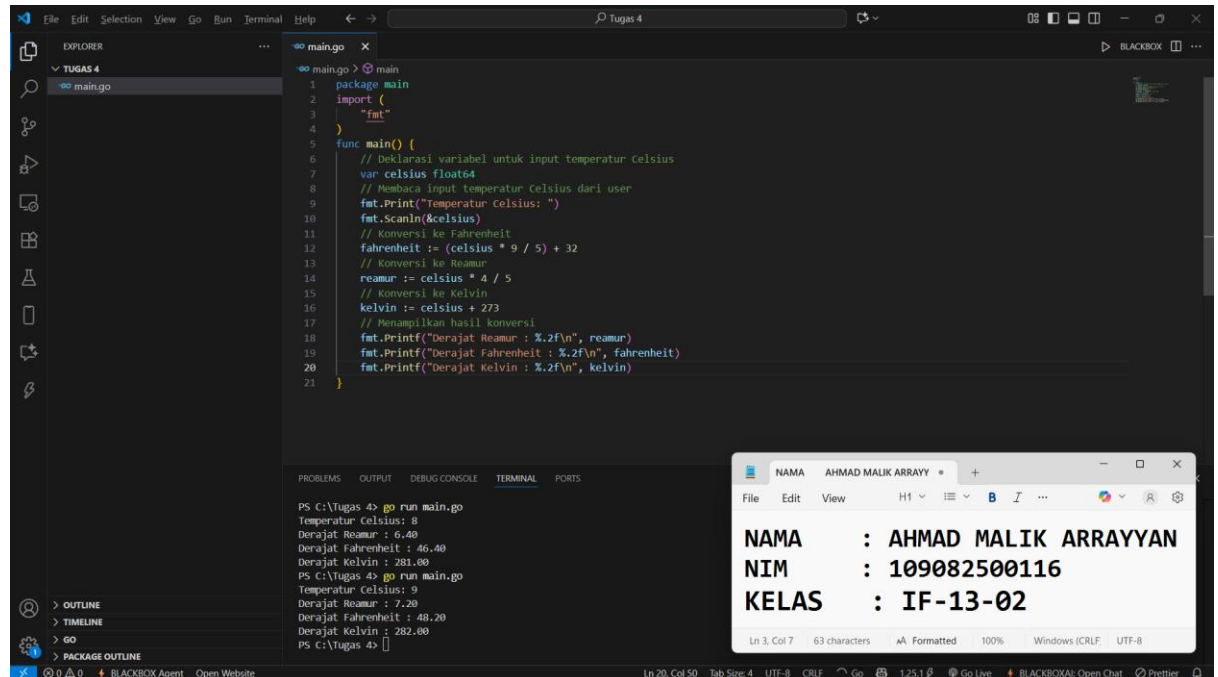
1. `var year int` — membuat tempat untuk menyimpan tahun yang akan dimasukkan (tipe integer).
2. `fmt.Print("Masukkan tahun: ")` — menampilkan pesan agar pengguna tahu harus mengisi apa.
3. `fmt.Scan(&year)` — membaca input dari keyboard; program akan berhenti (menunggu) sampai pengguna mengetik angka dan menekan Enter. `&year` artinya kita memberi alamat variabel supaya `Scan` bisa menaruh nilai ke situ.
4. `isleapYear(year)` — memanggil fungsi yang mengecek aturan kabisat.
  - Di dalam `isleapYear` ada 3 cabang:
  - jika `year % 400 == 0` → langsung `true` (tahun yang habis dibagi 400 adalah kabisat).
  - else if `year % 4 == 0 && year % 100 != 0` → `true` (habis dibagi 4 dan bukan kelipatan 100 → kabisat).
  - selain itu → `false`.
5. Hasil (`true/false`) ditampilkan dengan `fmt.Printf`.

#### 4. Tugas 4

##### Source code

```
package main
import (
    "fmt"
)
func main() {
    // Deklarasi variabel untuk input temperatur Celsius
    var celsius float64
    // Membaca input temperatur Celsius dari user
    fmt.Print("Temperatur Celsius: ")
    fmt.Scanln(&celsius)
    // Konversi ke Fahrenheit
    fahrenheit := (celsius * 9 / 5) + 32
    // Konversi ke Reamur
    reamur := celsius * 4 / 5
    // Konversi ke Kelvin
    kelvin := celsius + 273
    // Menampilkan hasil konversi
    fmt.Printf("Derajat Reamur : %.2f\n", reamur)
    fmt.Printf("Derajat Fahrenheit : %.2f\n", fahrenheit)
    fmt.Printf("Derajat Kelvin : %.2f\n", kelvin)
}
```

##### Screenshoot program



### Deskripsi program

- `var celsius float64`  
Menyediakan tempat untuk menyimpan suhu yang dimasukkan. `float64` dipilih supaya bisa menampung angka desimal (mis. 36.6).
- `fmt.Print("Temperatur Celsius: ")`  
Menampilkan prompt supaya pengguna tahu harus mengetik apa.
- `fmt.Scan(&celsius)`  
Membaca apa yang diketik user dari keyboard. Program akan menunggu sampai user mengetik dan tekan Enter. `&celsius` berarti alamat variabel — `Scan` butuh alamat agar bisa menaruh nilai ke variabel tersebut. Catatan: `fmt.Scan` mengembalikan jumlah item yang berhasil dibaca dan error; di kode ini hasil tersebut tidak diperiksa.
- `fahrenheit := (celsius * 9 / 5) + 32`  
Rumus konversi Celsius → Fahrenheit:  $F = C \times 9/5 + 32$   
 $F = C \times 9/5 + 32$ .
- `reamur := celsius * 4 / 5`  
Rumus Celsius → Reamur:  $R = C \times 4/5$   
 $R = C \times 4/5$ .
- `kelvin := celsius + 273`  
Rumus Celsius → Kelvin:  $K = C + 273$   
 $K = C + 273$ . (Di dunia ilmiah sering dipakai 273.15 sebagai offset; kode di screenshot memakai 273 saja.)
- `fmt.Printf(..., "%.2f")`  
Menampilkan angka dengan 2 angka di belakang koma. Contoh: 25 → 25.00.