

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 1  
MODUL III  
“TIPE DATA & VARIABEL”**



**Disusun oleh :**

**NAMA : Felix Pedrosa Valentino**

**NIM : 103112400056**

**S1 IF-12-01**

**Dosen Pengampu :**

**Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

**2024/2025**

## **DASAR TEORI**

### **1.) Integer Division dan Modulo**

Pembagian pada tipe data integer (integer division atau div) sedikit berbeda dengan pembagian yang sudah sering kita pelajari dari sekolah dasar. Hasil pembagian akan bertipe data integer, artinya kita akan mengabaikan bilangan yang muncul setelah tanda koma atau floating point. Hasil dari operasi div ini dikenal dengan istilah quotient.

Sebagai contoh hasil dari 10 (dividend) dibagi dengan 3 (divisor) adalah 3.333, sedangkan hasil div adalah 3. Artinya kita mengabaikan floating point dari hasil pembagian tersebut. Operator "/" di dalam bahasa pemrograman Go akan melakukan operasi integer division apabila tipe data dari kedua operan adalah integer dan pembagian biasa apabila operan bertipe float.

Modulo atau modulus "mod" adalah operasi untuk mencari nilai integer dari sisa pembagian pada integer division, biasanya dikenal juga dengan istilah remainder. Misalnya untuk operasi 10 dibagi dengan 3, maka sisa pembagian adalah 1. Operator modulo menggunakan "%" di dalam bahasa pemrograman Go.

Hubungan antara div dan mod dapat diilustrasikan sebagai berikut :

$$\text{dividend} = \text{quotient} \times \text{divisor} + \text{remainder}$$

Pada persamaan di atas quotient dan remainder adalah hasil operasi div dan mod antara dividend dan divisor.

Contoh penggunaan adalah mencari nilai digit di dalam suatu bilangan.

Semua bilangan apabila di modulus dengan 10 akan didapat satuan atau digit terakhir.

Misalnya:  $1234 \bmod 10$  akan diperoleh nilai 4,  $5677 \bmod 10$  akan diperoleh nilai 7. Apabila  $1234 \div 10$  akan diperoleh 123, sedangkan  $5677 \div 10$  akan didapat nilai 567. Pola ini bisa dimanfaatkan untuk mencari nilai digit-digit yang lain.

### **2. ) Casting atau Konversi Tipe Data**

Pada bahasa pemrograman Go, tipe data bersifat statis, artinya tipe data yang sudah didefinisikan tidak dapat diganti selama program berjalan. Casting

merupakan salah satu teknik konversi antar tipe data di dalam bahasa pemrograman.

Contoh:

```
var pi float64 = 3.14
```

```
var nilai int = int(pi)
```

Pada contoh tersebut, dilakukan casting terhadap nilai pi 3.14 menjadi bertipe integer pada variabel nilai. Cara casting ini dapat dilakukan dengan menuliskan tipe data tujuan sebagai fungsi yang kemudian nilainya di-assign ke dalam variabel dengan tipe data tujuan.

```
var var_name data_type = data_type(value)
```

Contoh yang lain adalah casting dengan menggunakan paket strconv untuk mengubah tipe data string menjadi integer ataupun sebaliknya. Berikut contoh konversi dari string ke integer:

```
var teks string = "2024"
```

```
tahun, err := strconv.Atoi(teks)
```

Berikutnya adalah contoh konversi dari integer menjadi string:

```
var bilangan int = "113071049"
```

```
var teks string = strconv.Itoa(bilangan)
```

## CONTOH SOAL

### 1.) Contoh Soal 1

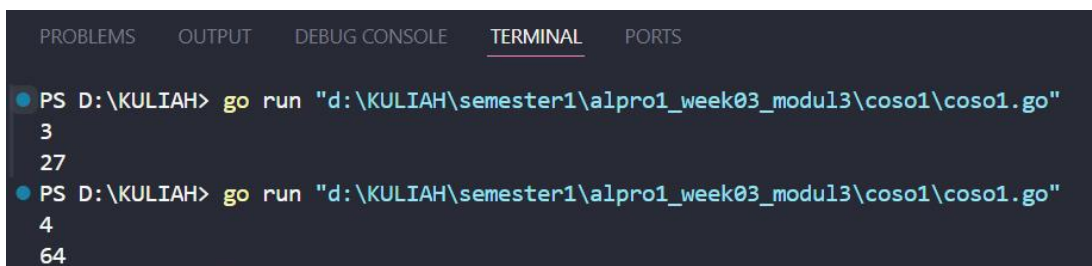
Source Code :

```
package main

import "fmt"

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

Output :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\coso1\coso1.go"
3
27
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\coso1\coso1.go"
4
64
```

Deskripsi Program :

Program tersebut adalah program yang ditulis dalam bahasa Go untuk menghitung volume kubus.

Penjelasan :

1. package main untuk Menandakan bahwa ini adalah program utama yang dapat dijalankan
2. import "fmt" untuk Mengimpor package fmt yang berisi fungsi-fungsi untuk input/output
3. Di dalam fungsi main() :

- Dideklarasikan 2 variabel bertipe float64: `sisi` (untuk panjang sisi kubus) dan `volume` (untuk hasil perhitungan)
- `fmt.Scan(&sisi)` untuk Menerima input dari pengguna untuk nilai sisi kubus
- `volume = (sisi * sisi * sisi)` untuk Menghitung volume kubus dengan rumus  $sisi^3$
- `fmt.Print(volume)` untuk Menampilkan hasil perhitungan volume

Cara kerja program:

1. Program akan menunggu user memasukkan sebuah angka (panjang sisi kubus)
2. Setelah angka dimasukkan, program akan menghitung volume kubus
3. Hasil perhitungan volume akan ditampilkan

Contoh:

- Jika user memasukkan sisi = 5
- Maka  $volume = 5 \times 5 \times 5 = 125$
- Program akan menampilkan output: 125

Program ini merupakan implementasi sederhana dari rumus volume kubus di mana  $volume = sisi \times sisi \times sisi$  atau  $sisi^3$ .

## 2.) Contoh Soal 2

Source Code :

```
package main

import "fmt"

func main() {
    var alas, tinggi, luas float64
    fmt.Print("Masukkan Alas: ")
    fmt.Scan(&alas)
    fmt.Print("tinggi: ")
```

```
fmt.Scan(&tinggi)
luas = (alas * tinggi / 2)
fmt.Print(luas)
}
```

Output :

```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\coso2\coso2.go"
Masukkan Alas: 8
tinggi: 5
20
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\coso2\coso2.go"
Masukkan Alas: 6
tinggi: 7
21
```

Deskripsi Program :

Program tersebut adalah program dalam bahasa Go untuk menghitung luas segitiga.

Penjelasan :

1. package main untuk Menandakan ini adalah program utama yang dapat dijalankan
2. import "fmt" untuk Mengimpor package fmt untuk fungsi input/output
3. Di dalam fungsi main() :
  - Dideklarasikan 3 variabel bertipe float64: `alas`, `tinggi`, dan `luas`
  - Program meminta input dengan cara :
    - Menampilkan "Masukkan Alas: " dan menerima input untuk nilai alas
    - Menampilkan "tinggi: " dan menerima input untuk nilai tinggi
  - Menghitung luas segitiga dengan rumus:  $luas = (alas * tinggi / 2)$
  - Menampilkan hasil perhitungan luas

Cara kerja program:

1. Program akan meminta user memasukkan nilai alas segitiga
2. Kemudian meminta nilai tinggi segitiga

3. Program menghitung luas menggunakan rumus:  $\text{Luas} = \frac{1}{2} \times \text{alas} \times \text{tinggi}$
4. Hasil perhitungan luas ditampilkan

Contoh penggunaan :

Masukkan Alas: 6

tinggi: 4

12

Program ini merupakan implementasi sederhana dari rumus luas segitiga, di mana  $\text{luas} = \frac{1}{2} \times \text{alas} \times \text{tinggi}$ . Program menggunakan tipe data `float64` sehingga dapat menangani nilai desimal dalam perhitungannya.

### 3.) Contoh Soal 3

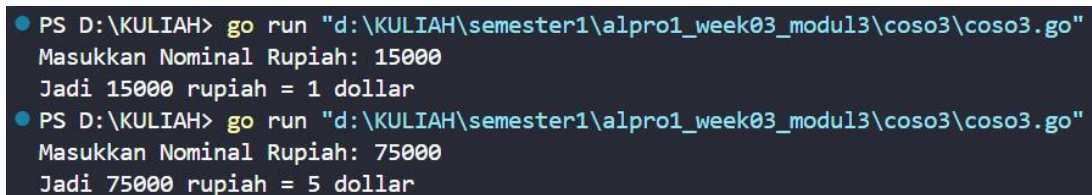
Source Code :

```
package main

import "fmt"

func main() {
    var rupiah, dollar float64
    fmt.Print("Masukkan Nominal Rupiah: ")
    fmt.Scan(&rupiah)
    dollar = (rupiah / 15000)
    fmt.Print("Jadi ", rupiah, " rupiah = ", dollar, " dollar")
}
```

Output :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\coso3\coso3.go"
Masukkan Nominal Rupiah: 15000
Jadi 15000 rupiah = 1 dollar
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\coso3\coso3.go"
Masukkan Nominal Rupiah: 75000
Jadi 75000 rupiah = 5 dollar
```

Deskripsi Program :

Program tersebut adalah program dalam bahasa Go untuk mengkonversi mata uang Rupiah ke Dollar.

Penjelasan :

1. package main untuk Menandakan ini adalah program utama yang dapat dijalankan
2. import "fmt" untuk Mengimpor package fmt untuk fungsi input/output
3. Di dalam fungsi main() :
  - Dideklarasikan 2 variabel bertipe float64: `rupiah` (untuk nominal dalam Rupiah) dan `dollar` (untuk hasil konversi)
  - Program meminta input dengan menampilkan "Masukkan Nominal Rupiah: "
  - Melakukan konversi dengan rumus:  $\text{`dollar`} = (\text{`rupiah`} / 15000)$
  - Menampilkan hasil konversi dalam format: "Jadi [nominal rupiah] rupiah = [hasil] dollar"

Cara kerja program :

1. Program meminta user memasukkan jumlah uang dalam Rupiah
2. Program melakukan konversi dengan kurs 1 Dollar = Rp 15.000
3. Menampilkan hasil konversi dalam format yang mudah dibaca

Contoh penggunaan :

Masukkan Nominal Rupiah: 75000

Jadi 75000 rupiah = 5 dollar

Catatan :

- Program menggunakan kurs tetap 1 Dollar = Rp 15.000
- Menggunakan tipe data float64 untuk menangani hasil pembagian desimal
- Output menampilkan hasil konversi dalam format yang informatif



- Program ini hanya melakukan konversi satu arah (Rupiah ke Dollar)

Program ini merupakan implementasi sederhana dari konverter mata uang, yang bisa berguna untuk memperkirakan nilai tukar Rupiah ke Dollar secara cepat, meskipun menggunakan nilai kurs yang tetap.

## LATIHAN SOAL

### 1.) Latihan Soal 1

Source code :

```
package main

import (
    "fmt"
    "log"
)

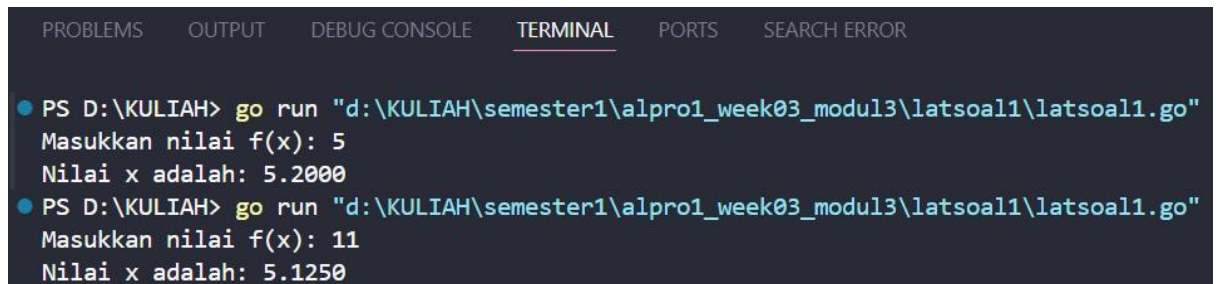
func calculateX(fx float64) (float64, error) {
    // Cek apakah  $fx + 5 = 0$ , yang akan membuat pembagian menjadi
    // tidak valid
    if fx == -5 {
        return 0, fmt.Errorf("nilai f(x) tidak boleh sama dengan -5 karena
menghasilkan pembagian dengan nol")
    }
    // Hitung nilai x dari persamaan  $f(x)$ 
    x := (2 / (fx + 5)) + 5
    return x, nil
}

func main() {
    var fx float64
    fmt.Print("Masukkan nilai f(x): ")
    _, err := fmt.Scan(&fx)
    if err != nil {
        log.Fatal("Input tidak valid")
    }

    // Hitung nilai x berdasarkan  $f(x)$ 
    x, err := calculateX(fx)
    if err != nil {
        log.Fatal(err)
    }

    fmt.Printf("Nilai x adalah: %.4f\n", x)
}
```

Output :



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

● PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal1\latsoal1.go"
Masukkan nilai f(x): 5
Nilai x adalah: 5.2000
● PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal1\latsoal1.go"
Masukkan nilai f(x): 11
Nilai x adalah: 5.1250
```

Deskripsi Program :

Program ini adalah program yang ditulis dalam bahasa Go untuk menghitung nilai x berdasarkan sebuah fungsi matematika. Program terdiri dari dua fungsi utama dan menggunakan package `fmt` untuk input/output serta package `log` untuk penanganan error.

Penjelasan Program :

1.) Function calculateX

func calculateX(fx float64) (float64, error)

Function ini menerima parameter:

fx (float64): Nilai f(x) yang diinputkan user

Function mengembalikan :

- Nilai x (float64) hasil perhitungan
- Error jika terjadi kesalahan

Rumus yang digunakan :

$$x = (2 / (f(x) + 5)) + 5$$

Validasi yang dilakukan :

- Mengecek apakah  $fx + 5 = 0$  ( $fx = -5$ ) yang akan menyebabkan pembagian dengan nol

- Jika  $fx = -5$ , function akan mengembalikan error

## 2.) Function main

Function main bertugas untuk :

1. Meminta input nilai  $f(x)$  dari user
2. Melakukan validasi input
3. Memanggil function calculateX untuk menghitung nilai x
4. Menampilkan hasil perhitungan atau pesan error

## Cara Kerja Program

1. Program meminta user memasukkan nilai  $f(x)$
2. Program melakukan validasi input untuk memastikan input berupa angka yang valid
3. Program menghitung nilai x menggunakan rumus yang telah ditentukan
4. Jika perhitungan berhasil, program menampilkan nilai x dengan 4 angka desimal
5. Jika terjadi error (input tidak valid atau  $f(x) = -5$ ), program akan menampilkan pesan error yang sesuai

## Penanganan Error

Program memiliki 2 jenis penanganan error:

1. Error input tidak valid (ketika input bukan angka)
2. Error matematis (ketika  $f(x) = -5$  yang menyebabkan pembagian dengan nol)

## Output Program

Program akan menampilkan output dalam format:

- Jika berhasil: "Nilai x adalah: [hasil]" (dengan 4 angka desimal)

- Jika gagal: Pesan error yang sesuai dengan jenis kesalahan

## 2.) Latihan Soal 2

Source Code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var radius int
    fmt.Print("Masukkan jari-jari bola: ")
    fmt.Scan(&radius)

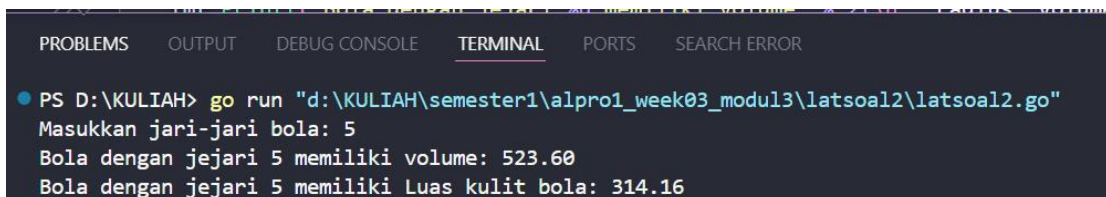
    // Konstanta pi
    const pi float64 = 3.1415926535

    // Hitung volume bola
    volume := (4.0 / 3.0) * pi * math.Pow(float64(radius), 3)

    // Hitung luas kulit bola
    area := 4.0 * pi * math.Pow(float64(radius), 2)

    fmt.Printf("Bola dengan jejari %d memiliki volume: %.2f\n", radius, volume)
    fmt.Printf("Bola dengan jejari %d memiliki Luas kulit bola: %.2f\n", radius, area)
}
```

Output :



```
PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal2\latsoal2.go"
Masukkan jari-jari bola: 5
Bola dengan jejari 5 memiliki volume: 523.60
Bola dengan jejari 5 memiliki Luas kulit bola: 314.16
```

## Deskripsi Program :

Program tersebut adalah program Go yang berfungsi untuk menghitung volume dan luas permukaan (kulit) bola berdasarkan input jari-jari yang dimasukkan pengguna. Berikut deskripsi detail programnya:

### 1. Package dan Import :

- Program menggunakan package `main`
- Mengimpor package `fmt` untuk input/output
- Mengimpor package `math` untuk operasi matematika

### 2. Fungsi Input :

- Program meminta pengguna memasukkan nilai jari-jari bola
- Input disimpan dalam variabel `radius` bertipe integer

### 3. Konstanta :

- Mendefinisikan konstanta `pi` dengan nilai 3.1415926535

### 4. Perhitungan :

- Menghitung volume bola menggunakan rumus:  $V = \frac{4}{3} \times \pi \times r^3$
- Menghitung luas permukaan bola menggunakan rumus:  $A = 4 \times \pi \times r^2$
- Menggunakan `math.Pow()` untuk menghitung perpangkatan
- Melakukan konversi tipe data dari int ke float64 untuk perhitungan yang akurat

### 5. Output :

- Menampilkan hasil perhitungan volume dan luas permukaan bola
- Hasil ditampilkan dengan format 2 angka desimal menggunakan %.2f
- Output mencantumkan jari-jari yang digunakan dalam perhitungan

Contoh penggunaan :

Masukkan jari-jari bola: 5

Bola dengan jejari 5 memiliki volume: 523.60

Bola dengan jejari 5 memiliki Luas kulit bola: 314.16

Program ini berguna untuk membantu menghitung volume dan luas permukaan bola secara cepat dan akurat berdasarkan input jari-jari yang diberikan oleh pengguna.

### 3.) Latihan Soal 3

Source Code :

```
package main

import "fmt"

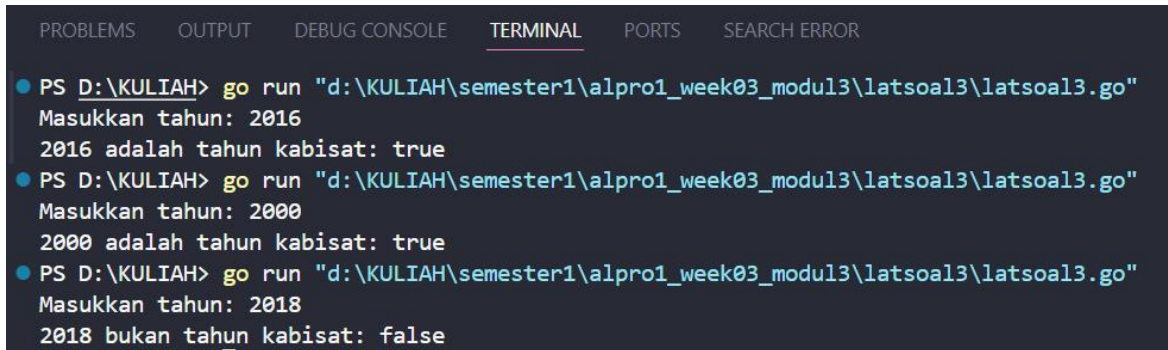
func isKabisat(tahun int) bool {
    if tahun%400 == 0 {
        return true
    } else if tahun%100 == 0 {
        return false
    } else if tahun%4 == 0 {
        return true
    }
    return false
}

func main() {
    var tahun int
    fmt.Print("Masukkan tahun: ")
    fmt.Scan(&tahun)

    if isKabisat(tahun) {
        fmt.Printf("%d adalah tahun kabisat: true\n", tahun)
    } else {
        fmt.Printf("%d bukan tahun kabisat: false\n", tahun)
    }
}
```

```
}
```

Output :



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
● PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal3\latsoal3.go"
Masukkan tahun: 2016
2016 adalah tahun kabisat: true
● PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal3\latsoal3.go"
Masukkan tahun: 2000
2000 adalah tahun kabisat: true
● PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal3\latsoal3.go"
Masukkan tahun: 2018
2018 bukan tahun kabisat: false
```

Deskripsi Program :

Program tersebut adalah program Go (Golang) yang berfungsi untuk menentukan apakah suatu tahun merupakan tahun kabisat atau bukan. Berikut deskripsi detail programnya :

1. Package dan Import :

- Program menggunakan package `main`
- Mengimpor package `fmt` untuk input/output

2. Fungsi isKabisat :

- Menerima parameter `tahun` bertipe integer
- Mengembalikan nilai boolean (true/false)
- Mengimplementasikan aturan penentuan tahun kabisat :
  - Tahun habis dibagi 400 → tahun kabisat
  - Tahun habis dibagi 100 → bukan tahun kabisat
  - Tahun habis dibagi 4 → tahun kabisat
  - Selain kondisi di atas → bukan tahun kabisat



### 3. Fungsi main :

- Mendeklarasikan variabel `tahun` bertipe integer
- Meminta input tahun dari pengguna
- Memanggil fungsi `isKabisat()` untuk memeriksa tahun
- Menampilkan hasil apakah tahun tersebut kabisat atau bukan

### 4. Logika Penentuan Tahun Kabisat :

- Jika  $\text{tahun} \% 400 = 0 \rightarrow \text{kabisat}$
- Jika  $\text{tahun} \% 100 = 0 \rightarrow \text{bukan kabisat}$
- Jika  $\text{tahun} \% 4 = 0 \rightarrow \text{kabisat}$
- Selainnya  $\rightarrow \text{bukan kabisat}$

Program ini berguna untuk :

- Menentukan tahun kabisat secara cepat dan akurat
- Membantu dalam perhitungan kalender
- Pembelajaran tentang konsep tahun kabisat
- Demonstrasi penggunaan fungsi boolean dan kondisional dalam pemrograman

Catatan: Tahun kabisat adalah tahun yang memiliki 366 hari, dengan tambahan satu hari pada bulan Februari (29 Februari). Sistem ini digunakan untuk menyesuaikan penanggalan dengan revolusi Bumi mengelilingi Matahari.

## 4.) Latihan Soal 4

Source Code :

```
package main

import (
    "fmt"
```

```

    "log"
)

// Function to convert Celsius to other temperature scales
func convertTemperature(celsius float64) (fahrenheit, reamur, kelvin float64) {
    // Convert Celsius to Fahrenheit
    fahrenheit = (celsius * 9 / 5) + 32

    // Convert Celsius to Reamur
    reamur = celsius * 4 / 5

    // Convert Fahrenheit to Kelvin
    kelvin = (fahrenheit + 459.67) * 5 / 9

    return fahrenheit, reamur, kelvin
}

func main() {
    var celsius float64

    // Input temperature in Celsius
    fmt.Println("Masukkan suhu dalam Celsius: ")
    _, err := fmt.Scan(&celsius)
    if err != nil {
        log.Fatal("Input tidak valid. Masukkan angka yang valid.")
    }

    // Convert temperature
    fahrenheit, reamur, kelvin := convertTemperature(celsius)

    // Display results
    fmt.Println("\nHasil Konversi Suhu:")
    fmt.Printf("Derajat reamur dari %.2f°C = %.2f°R\n", celsius, reamur)
    fmt.Printf("Derajat fahrenheit dari %.2f°C = %.2f°F\n", celsius,
fahrenheit)
    fmt.Printf("Derajat kelvin dari %.2f°C = %.2f°K\n", celsius, kelvin)
}

```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR

● PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal4\latsoal4.go"
Masukkan suhu dalam Celsius: 50

Hasil Konversi Suhu:
Derajat reamur dari 50.00°C = 40.00°R
Derajat fahrenheit dari 50.00°C = 122.00°F
Derajat kelvin dari 50.00°C = 323.15°K
● PS D:\KULIAH> go run "d:\KULIAH\semester1\alpro1_week03_modul3\latsoal4\latsoal4.go"
Masukkan suhu dalam Celsius: 30

Hasil Konversi Suhu:
Derajat reamur dari 30.00°C = 24.00°R
Derajat fahrenheit dari 30.00°C = 86.00°F
Derajat kelvin dari 30.00°C = 303.15°K
```

Deskripsi Program :

Program ini merupakan sebuah program yang dibuat dalam bahasa Go (Golang) untuk mengonversi suhu dari skala Celsius ke tiga skala suhu lainnya yaitu Reamur, Fahrenheit, dan Kelvin.

Penjelasan :

1. Fungsi convertTemperature :

- Fungsi ini bertugas untuk menghitung dan mengonversi suhu dari Celsius ke tiga skala lainnya :
- Fahrenheit : menggunakan rumus  $\text{fahrenheit} = (\text{celsius} * 9 / 5) + 32$ .
- Reamur : menggunakan rumus  $\text{reamur} = \text{celsius} * 4 / 5$ .
- Kelvin : menggunakan rumus  $\text{kelvin} = (\text{fahrenheit} + 459.67) * 5 / 9$ .
- Fungsi ini mengembalikan tiga hasil perhitungan: suhu dalam Fahrenheit, Reamur, dan Kelvin.

2. Fungsi main:

- Program meminta pengguna untuk memasukkan suhu dalam Celsius menggunakan input dari terminal.
- Setelah suhu dalam Celsius dimasukkan oleh pengguna, program memanggil fungsi `convertTemperature` untuk mengonversi suhu tersebut ke Fahrenheit, Reamur, dan Kelvin.

- Jika terjadi kesalahan input (bukan angka yang valid), program akan menampilkan pesan kesalahan menggunakan `'log.Fatal'` dan menghentikan eksekusi.
- Program menampilkan hasil konversi dalam format yang rapi, menampilkan suhu dalam skala Reamur, Fahrenheit, dan Kelvin, dengan dua angka di belakang koma.

#### Contoh Interaksi:

- Misalkan pengguna memasukkan suhu 30°C, hasil konversi yang akan ditampilkan adalah :
  - 30°C = 24.00°R (Reamur)
  - 30°C = 86.00°F (Fahrenheit)
  - 30°C = 303.15°K (Kelvin)

Perlu diperhatikan bahwa pada hasil akhir, nilai suhu dalam tiga skala ditampilkan dengan format dua angka di belakang koma.

## **DAFTAR PUSTAKA**

Laporan Resmi Praktikum Tipe Data, Konstanta Dan Variabel. (n.d.). Retrieved from <https://id.scribd.com/doc/217152785/Laporan-Resmi-Praktikum-Tipe-Data-Konstanta-dan-Variabel>

(N.d.). Retrieved from [https://jeffmarzal.staff.unja.ac.id/wp-content/uploads/sites/22/2017/08/Temu\\_2\\_tipedata-dan-variabel.pdf](https://jeffmarzal.staff.unja.ac.id/wp-content/uploads/sites/22/2017/08/Temu_2_tipedata-dan-variabel.pdf)