

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



**DISUSUN OLEH:  
FEROS PEDROSA VALENTINO  
103112400055  
S1 IF-12-01**

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
PURWOKERTO  
2024/2025**

## DASAR TEORI

Tipe Bentuk merupakan salah jenis tipe data yang didefinisikan oleh pemrograman(*user-defined data type*). Jenis tipe bentuk ada 2 yaitu

1. Tipe Bentuk: Alias
  - a. Deklarasi tipe bentuk sebagai alias.
  - b. Salah satu cara pemberian nama tipe data dengan nama baru sehingga lebih mudah untuk dikenali atau dipahami oleh pemrogram.
2. Tipe Bentuk: Structure/Record:
  - a. Memungkinkan kita untuk mengelompokkan beberapa variabel yang mungkin berelasi menjadi suatu kesatuan.
  - b. Sebagai contoh, data buku biasanya terdiri dari judul buku, nama penulis, tahun terbit, nama penerbit, edisi, dan data-data lainnya.
  - c. Tipe data dari field bisa berupa tipe data dasar ataupun tipe bentuk lainnya.  
Contoh:
    - Tipe Waktu (jam, menit, dan detik)
    - Tipe Orang (nama, gender, alamat, agama dan biodata lainnya)
    - Tipe Buku (judul, penulis, tahun terbit, penerbit, dan data lainnya)

### Fungsi Multiple Return

Di GoLang suatu fungsi dapat mengembalikan nilai lebih dari satu. Teknik ini bisa menjadi alternatif selain menggunakan tipe data kolektif seperti map, slice, atau struct sebagai nilai balik.

Di Go (Golang), fungsi `math.Pow` dan `math.Sqrt` adalah dua fungsi bawaan yang disediakan dalam paket `math` untuk melakukan perhitungan perpangkatan dan akar kuadrat.

1. `math.Pow`
  - Fungsi ini digunakan untuk menghitung hasil perpangkatan dari dua angka.
  - Sintaks: `math.Pow(x, y)`
  - Parameter:
    - `x` adalah basis (angka yang akan dipangkatkan).
    - `y` adalah eksponen (pangkat dari basis).
  - Hasil: Mengembalikan nilai  $x^y$  (`x` pangkat `y`) dalam bentuk `float64`.
2. `math.Sqrt`
  - Fungsi ini digunakan untuk menghitung akar kuadrat dari suatu angka.
  - Sintaks: `math.Sqrt(x)`
  - Parameter: `x` adalah angka yang ingin dihitung akar kuadratnya.
  - Hasil: Mengembalikan akar kuadrat dari `x` dalam bentuk `float64`.

## CONTOH SOAL

### 1. Coso1

Source code:

```
package main

import "fmt"

func main() {
    var detik, jam, menit int
    fmt.Scan(&detik)
    jam = detik / 3600
    menit = (detik % 3600) / 60
    detik = detik % 60
    fmt.Println(jam, "jam", menit, "menit", detik, "detik")
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\coso1\coso1.go"
3661
1 jam 1 menit 1 detik
PS C:\semester1_alpro1\alpro1_week4>
```

Deskripsi program:

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk mengonversi sejumlah total detik yang dimasukkan oleh pengguna menjadi format jam, menit, dan detik. Berikut adalah penjelasan dari program tersebut:

1. Program mendeklarasikan tiga variabel: detik, jam, dan menit, semuanya bertipe integer.
2. Program meminta input dari pengguna dalam bentuk total detik dan menyimpannya ke variabel detik.
3. Program menghitung jumlah jam dengan membagi total detik dengan 3600 (jumlah detik dalam satu jam).
4. Program menghitung jumlah menit dengan membagi sisa detik setelah dihitung jumlah jamnya, kemudian membagi sisa tersebut dengan 60 (jumlah detik dalam satu menit).
5. Sisa detik yang tidak dapat dikonversi menjadi jam atau menit dihitung dengan operasi modulo.
6. Program mencetak hasil konversi dalam format "X jam Y menit Z detik".

### 2. Coso2

Source code:

```
package main

import "fmt"
```

```
func main() {
    var bilangan, d1, d2, d3 int
    fmt.Scan(&bilangan)
    d1 = bilangan / 100
    d2 = bilangan % 100 / 10
    d3 = bilangan % 10
    fmt.Println(d1 <= d2 && d2 <= d3)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\coso2\coso2.go"
256
true
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\coso2\coso2.go"
361
false
```

Deskripsi Program:

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk memeriksa apakah digit-digit dari suatu bilangan tiga digit yang dimasukkan oleh pengguna tersusun secara tidak menurun. Berikut adalah penjelasan dari program tersebut:

1. Program mendeklarasikan empat variabel: *bilangan*, *d1*, *d2*, dan *d3*, semuanya bertipe integer.
2. Program meminta input dari pengguna, yang diharapkan berupa bilangan tiga digit, dan menyimpannya ke variabel *bilangan*.
3. Program memisahkan bilangan tersebut menjadi tiga digit:
  - *d1*: digit ratusan, diperoleh dengan membagi bilangan dengan 100.
  - *d2*: digit puluhan, diperoleh dengan mengambil sisa pembagian bilangan dengan 100, kemudian membaginya dengan 10.
  - *d3*: digit satuan, diperoleh dengan mengambil sisa pembagian bilangan dengan 10.
4. Program kemudian memeriksa apakah digit ratusan (*d1*) lebih kecil atau sama dengan digit puluhan (*d2*), dan apakah digit puluhan lebih kecil atau sama dengan digit satuan (*d3*).
5. Hasil dari pemeriksaan ini adalah nilai boolean (*true* atau *false*), yang menunjukkan apakah digit-digit bilangan tersebut tersusun dalam urutan yang tidak menurun. Hasilnya kemudian dicetak.

### 3. Coso3

Source code:

```
package main

import "fmt"

func main() {
    var beratBadan, tinggiBadan, bmi float64
    fmt.Print("Masukkan Berat Badan (kg): ")
    fmt.Scan(&beratBadan)
    fmt.Print("Masukkan Tinggi Badan (m): ")
```

```
fmt.Scan(&tinggiBadan)
bmi = beratBadan / (tinggiBadan * tinggiBadan)
fmt.Printf("BMI anda: %.2f", bmi)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\coso3\coso3.go"
Masukkan Berat Badan (kg): 70
Masukkan Tinggi Badan (m): 1.75
BMI anda: 22.86
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\coso3\coso3.go"
Masukkan Berat Badan (kg): 60
Masukkan Tinggi Badan (m): 1.6
BMI anda: 23.44
```

Deskripsi program:

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menghitung Body Mass Index (BMI) berdasarkan berat badan dan tinggi badan yang dimasukkan oleh pengguna. Berikut adalah penjelasan dari program tersebut:

1. Program mendeklarasikan tiga variabel bertipe float64: beratBadan, tinggiBadan, dan bmi untuk menyimpan berat badan, tinggi badan, dan hasil BMI.
2. Program meminta input berat badan dalam kilogram dari pengguna, kemudian menyimpannya ke variabel beratBadan.
3. Program meminta input tinggi badan dalam meter dari pengguna, kemudian menyimpannya ke variabel tinggiBadan.
4. Program menghitung BMI dengan menggunakan rumus:  $BMI = \text{beratBadan} / (\text{tinggiBadan} * \text{tinggiBadan})$ . Hasil BMI disimpan dalam variabel bmi.
5. Program mencetak hasil BMI dengan format dua angka di belakang koma menggunakan fungsi fmt.Printf().

## LATIHAN SOAL MANDIRI

1. Sebuah program digunakan untuk menghitung total harga setelah memperoleh diskon dengan besaran tertentu.

**Masukan** terdiri dari dua baris. Baris pertama adalah bilangan bulat yang menyatakan total belanja awal, sedangkan baris kedua merupakan bilangan bulat yang menyatakan besarnya diskon dalam satuan persen.

**Keluaran** berupa total belanja akhir setelah dipotong oleh diskon.

Source code:

```
package main

import "fmt"

func main() {
    var totalBelanja, diskon, totalAkhir int
    fmt.Print("Masukkan Total Belanja: ")
    fmt.Scan(&totalBelanja)
    fmt.Print("Masukkan Diskon (%): ")
    fmt.Scan(&diskon)

    totalAkhir = totalBelanja - (totalBelanja * diskon / 100)

    fmt.Println("Total belanja setelah diskon:", totalAkhir)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\diskon\diskon.go"
Masukkan Total Belanja: 100000
Masukkan Diskon (%): 10
Total belanja setelah diskon: 90000
```

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\diskon\diskon.go"
Masukkan Total Belanja: 200000
Masukkan Diskon (%): 20
Total belanja setelah diskon: 160000
```

Deskripsi program:

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menghitung total belanja akhir setelah menerapkan diskon yang diberikan oleh pengguna. Berikut adalah penjelasan dari program tersebut:

1. Program mendeklarasikan tiga variabel bertipe integer: totalBelanja, diskon, dan totalAkhir.
  - totalBelanja menyimpan jumlah total belanja yang dimasukkan oleh pengguna.
  - diskon menyimpan persentase diskon yang diberikan.
  - totalAkhir menyimpan hasil akhir setelah diskon diterapkan.
2. Program meminta input dari pengguna berupa total belanja dalam bentuk angka, yang kemudian disimpan ke dalam variabel totalBelanja.
3. Program juga meminta input berupa persentase diskon, yang disimpan dalam variabel diskon.

4. Program menghitung total belanja setelah diskon dengan rumus:

$$\text{totalAkhir} = \text{totalBelanja} - \left( \text{totalBelanja} * \frac{\text{diskon}}{100} \right)$$

Nilai total setelah diskon disimpan ke variabel totalAkhir.

5. Program mencetak total belanja setelah diskon ke layar.

2. Buatlah program yang digunakan untuk menentukan berat badan seseorang apabila diketahui nilai BMI dan tinggi badannya.

**Masukan** terdiri dari dua buah bilangan riil yang menyatakan nilai BMI dan tinggi badan dalam satuan meter.

**Keluaran** berupa berat badan seseorang dalam satuan kilogram.

Source code:

```
package main

import "fmt"

func main() {
    var bmi, tinggiBadan, beratBadan float64

    fmt.Print("Masukkan nilai BMI: ")
    fmt.Scan(&bmi)
    fmt.Print("Masukkan tinggi badan (m): ")
    fmt.Scan(&tinggiBadan)

    beratBadan = bmi * (tinggiBadan * tinggiBadan)

    fmt.Printf("Berat badan anda adalah: %.f kg\n", beratBadan)
}
```

Output:

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\bmi\bmi.go"
Masukkan nilai BMI: 22.85
Masukkan tinggi badan (m): 1.75
Berat badan anda adalah: 70 kg
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\bmi\bmi.go"
Masukkan nilai BMI: 23.43
Masukkan tinggi badan (m): 1.6
Berat badan anda adalah: 60 kg
```

Deskripsi program:

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menghitung berat badan seseorang berdasarkan nilai BMI (Body Mass Index) dan tinggi badan yang diberikan oleh pengguna. Berikut adalah penjelasan dari program tersebut:

1. Program mendeklarasikan tiga variabel bertipe float64, yaitu:
  - bmi: menyimpan nilai Body Mass Index.
  - tinggiBadan: menyimpan tinggi badan dalam satuan meter.
  - beratBadan: menyimpan hasil perhitungan berat badan.
2. Input dari pengguna:
  - Program meminta pengguna untuk memasukkan nilai BMI, yang kemudian disimpan dalam variabel bmi.

- Program juga meminta pengguna untuk memasukkan tinggi badan dalam meter, yang disimpan dalam variabel tinggiBadan.
3. Perhitungan berat badan:
    - Berat badan dihitung berdasarkan rumus BMI: 
$$\text{BMI} = \frac{\text{beratBadan}}{\text{tinggiBadan} * \text{tinggiBadan}}$$
    - Sehingga untuk menghitung berat badan digunakan rumus: 
$$\text{beratBadan} = \text{BMI} * (\text{tinggiBadan} * \text{tinggiBadan})$$
    - Hasil perhitungan ini disimpan dalam variabel beratBadan.
  4. Program mencetak hasil berat badan ke layar dengan format angka bulat (tanpa angka desimal) menggunakan fungsi fmt.Printf().
3. Dalam sebuah peta, terdapat tiga titik yang membentuk segitiga. Titik A, B, dan C masing-masing memiliki koordinat dalam sistem kartesius 2 dimensi. Tugas Anda adalah menghitung panjang sisi-sisi segitiga yang dibentuk oleh titik-titik tersebut dan menentukan sisi terpanjang dari segitiga tersebut menggunakan teorema Pythagoras.  
**Masukan** terdiri dari tiga baris, yang mana masing-masing berisi dua bilangan riil yang menyatakan koordinat titik A, B, dan C dalam format x y.  
**Keluaran** Sebuah bilangan riil yang menyatakan panjang sisi terpanjang dari segitiga yang dibentuk oleh titik-titik tersebut. Hasil harus ditampilkan dengan dua angka di belakang koma.

Source code:

```
package main
import (
    "fmt"
    "math"
)

func main() {
    var ax, ay, bx, by, cx, cy float64

    fmt.Print("Masukkan koordinat titik A (x y): ")
    fmt.Scan(&ax, &ay)
    fmt.Print("Masukkan koordinat titik B (x y): ")
    fmt.Scan(&bx, &by)
    fmt.Print("Masukkan koordinat titik C (x y): ")
    fmt.Scan(&cx, &cy)

    AB := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))
    BC := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))
    CA := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))

    sisiTerpanjang := math.Max(AB, math.Max(BC, CA))

    fmt.Printf("Sisi terpanjang dari segitiga tersebut adalah: %.2f\n",
        sisiTerpanjang)
}
```



Output:

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\peta\peta.go"
Masukkan koordinat titik A (x y): 1.0
1.0
Masukkan koordinat titik B (x y): 4.0
1.0
Masukkan koordinat titik C (x y): 1.0
5.0
Sisi terpanjang dari segitiga tersebut adalah: 5.00
```

```
PS C:\semester1_alpro1\alpro1_week4> go run "c:\semester1_alpro1\alpro1_week4\peta\peta.go"
Masukkan koordinat titik A (x y): 0.0
0.0
Masukkan koordinat titik B (x y): 3.0
0.0
Masukkan koordinat titik C (x y): 3.0
4.0
Sisi terpanjang dari segitiga tersebut adalah: 5.00
```

Deskripsi program:

Kode di atas merupakan program Go yang menghitung panjang sisi terpanjang dari sebuah segitiga berdasarkan koordinat tiga titik, yaitu titik A, B, dan C. Berikut penjelasan mengenai program tersebut:

1. Paket dan Impor:
  - Program berada dalam paket main yang diperlukan untuk menjalankan program Go.
  - Paket fmt digunakan untuk menangani input/output seperti membaca data dari pengguna dan mencetak hasil ke layar.
  - Paket math digunakan untuk melakukan perhitungan matematis, seperti menghitung akar kuadrat dan eksponen.
2. Tiga titik A, B, dan C memiliki koordinat dalam bentuk variabel ax, ay, bx, by, cx, cy yang semuanya bertipe float64 untuk mendukung angka desimal.
3. Program meminta pengguna untuk memasukkan koordinat titik A, B, dan C menggunakan perintah fmt.Scan, yang mengisi variabel koordinat tersebut dengan nilai yang diberikan oleh pengguna.
4. Menghitung Panjang Sisi:
  - Panjang sisi-sisi segitiga (AB, BC, dan CA) dihitung menggunakan teorema Pythagoras:
    - AB dihitung sebagai jarak antara titik A dan B.
    - BC dihitung sebagai jarak antara titik B dan C.
    - CA dihitung sebagai jarak antara titik C dan A.
  - Formula jarak antar dua titik dalam koordinat kartesius adalah:
$$\text{jarak} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
Perhitungan dilakukan dengan fungsi math.Sqrt untuk akar kuadrat dan math.Pow untuk perpangkatan.
5. Program menggunakan fungsi math.Max untuk mencari sisi terpanjang dari ketiga sisi (AB, BC, CA).
6. Program menampilkan panjang sisi terpanjang dengan dua angka di belakang koma menggunakan format %.2f.

## DAFTAR PUSTAKA

A.11. Konstanta. (n.d.). Retrieved from

<https://dasarpemrogramangolang.novalagung.com/A-konstanta.html>

A.19. Fungsi Multiple Return. (n.d.). Retrieved from

<https://dasarpemrogramangolang.novalagung.com/A-fungsi-multiple-return.html>