

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 4
“TIPE DATA & VARIABEL”



DISUSUN OLEH:
Muhammad Shabrian Fadly
103112400087
S1 IF-12-01

DOSEN:
Yohani Setiya Rafika Nur, M. Kom.

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

DASAR TEORI

Tipe data adalah suatu memori atau media pada komputer yang digunakan untuk menampung informasi atau data sementara. Tipe data berfungsi untuk mempresentasikan jenis dari sebuah nilai yang terdapat dalam program. Berikut ini merupakan macam-macam tipe data serta penjelasannya yang terdapat pada Golang :

1. Tipe Data Numerik Non-Desimal

Tipe data numerik non-desimal atau **non floating point** di Go ada beberapa jenis. Secara umum ada 2 tipe data kategori ini yang perlu diketahui.

uint, tipe data untuk bilangan cacah (bilangan positif).

int, tipe data untuk bilangan bulat (bilangan negatif dan positif).

Kedua tipe data di atas kemudian dibagi lagi menjadi beberapa jenis, dengan pembagian berdasarkan lebar cakupan nilainya

2. Tipe Data Numerik Desimal

Tipe data numerik desimal yang perlu diketahui ada 2, float32 dan float64. Perbedaan kedua tipe data tersebut berada di lebar cakupan nilai desimal yang bisa ditampung

3. Tipe Data bool (Boolean)

Tipe data bool berisikan hanya 2 variansi nilai, true dan false. Tipe data ini biasa dimanfaatkan dalam seleksi kondisi dan perulangan

4. Tipe Data string

Ciri khas dari tipe data string adalah nilainya di apit oleh tanda *quote* atau petik dua ("). Selain menggunakan tanda quote, deklarasi string juga bisa dengan tanda *grave accent/backticks* (`), tanda ini terletak di sebelah kiri tombol 1. Keistimewaan string yang dideklarasikan menggunakan backticks adalah membuat semua karakter di dalamnya **tidak di escape**, termasuk \n, tanda petik dua dan tanda petik satu, baris baru, dan lainnya. Semua akan terdeteksi sebagai string

5. Nilai nil & Zero Value

nil bukan merupakan tipe data, melainkan sebuah nilai. Variabel yang isi nilainya nil berarti memiliki nilai kosong.

Semua tipe data yang sudah dibahas di atas memiliki zero value (nilai default tipe data). Artinya meskipun variabel dideklarasikan dengan tanpa nilai awal, tetap akan ada nilai default-nya.

CONTOH SOAL

1. Latihan1

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 D:\Coding manja> go run "d:\Coding manja\Modul4\conso\tempCodeRunnerFile.go"
3661
1 jam 1 menit 1 detik
PS D:\Coding manja> go run "d:\Coding manja\Modul4\conso\tempCodeRunnerFile.go"
7322
2 jam 2 menit 2 detik
PS D:\Coding manja> go run "d:\Coding manja\Modul4\conso\tempCodeRunnerFile.go"
3600
1 jam 0 menit 0 detik
```

Deskripsi Program:

1. **Import Package:** Program mengimpor package fmt, yang digunakan untuk format input dan output.
2. **Deklarasi Variabel:** Tiga variabel detik, jam, dan menit dideklarasikan sebagai tipe integer.
3. **Input dari Pengguna:** Program meminta pengguna untuk memasukkan jumlah detik melalui fmt.Scan(&detik).
4. **Konversi Detik ke Jam:**
 - $\text{jam} = \text{detik} / 3600$: Menghitung jumlah jam dengan membagi total detik dengan 3600 (jumlah detik dalam satu jam).
5. **Konversi Sisa Detik ke Menit:**
 - $\text{menit} = (\text{detik} \% 3600) / 60$: Menghitung jumlah menit dari sisa detik setelah menghitung jam. $\text{detik} \% 3600$ mendapatkan sisa detik setelah jam dihitung, kemudian dibagi 60 untuk mendapatkan menit.

6. **Sisa Detik:**

- $\text{detik} = \text{detik} \% 60$: Menghitung sisa detik setelah menghitung jam dan menit.

7. **Output:** Program mencetak hasil dalam format "X jam Y menit Z detik".

2. Latihan 2

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 D:\Coding manja> go run "d:\Coding manja\Modul4\conso2\tempCodeRunnerFile.go"
362
false
PS D:\Coding manja> go run "d:\Coding manja\Modul4\conso2\tempCodeRunnerFile.go"
256
true
PS D:\Coding manja> go run "d:\Coding manja\Modul4\conso2\tempCodeRunnerFile.go"
189
true
```

Deskripsi Program:

1. **Import Package:** Program mengimpor package `fmt`, yang digunakan untuk input dan output.
2. **Deklarasi Variabel:** Tiga variabel `d1`, `d2`, dan `d3` dideklarasikan sebagai tipe integer, bersama dengan variabel `bilangan` untuk menyimpan input dari pengguna.
3. **Input dari Pengguna:** Program meminta pengguna untuk memasukkan sebuah bilangan melalui `fmt.Scan(&bilangan)`.
4. **Ekstraksi Digit:**
 - `d1 = bilangan / 100`: Mengambil digit pertama (ratusan) dengan membagi bilangan dengan 100.
 - `d2 = bilangan % 100 / 10`: Mengambil digit kedua (puluhan) dengan mendapatkan sisa dari bilangan dibagi 100 dan kemudian membagi hasilnya dengan 10.
 - `d3 = bilangan % 10`: Mengambil digit ketiga (satuan) dengan mendapatkan sisa dari bilangan dibagi 10.
5. **Pemeriksaan Urutan:**
 - `fmt.Println(d1 <= d2 && d2 <= d3)`: Program memeriksa apakah digit pertama kurang dari atau sama dengan digit kedua, dan digit kedua kurang dari atau

sama dengan digit ketiga. Jika kondisi ini benar, hasilnya adalah true; jika tidak, hasilnya adalah false.

3. Latihan 3

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 D:\Coding manja> go run "d:\Coding manja\Modul4\conso3\tempCodeRunnerFile.go"
Masukkan Berat Badan (kg): 70
Masukkan Tinggi Badan (m): 1.75
BMI anda: 22.86
PS D:\Coding manja> go run "d:\Coding manja\Modul4\conso3\tempCodeRunnerFile.go"
Masukkan Berat Badan (kg): 60
Masukkan Tinggi Badan (m): 1.6
BMI anda: 23.44
PS D:\Coding manja> go run "d:\Coding manja\Modul4\conso3\tempCodeRunnerFile.go"
Masukkan Berat Badan (kg): 80
Masukkan Tinggi Badan (m): 1.8
BMI anda: 24.69
```

Deskripsi Program:

1. **Deklarasi Variabel:** Tiga variabel bertipe float64 dideklarasikan:
 - o beratbadan: untuk menyimpan berat badan dalam kilogram.
 - o tinggibadan: untuk menyimpan tinggi badan dalam meter.
 - o bmi: untuk menyimpan hasil perhitungan BMI.
2. **Input Berat Badan:**
 - o Program mencetak prompt "Masukkan Berat Badan (kg): " dan kemudian menerima input dari pengguna melalui `fmt.Scan(&beratbadan)`.
3. **Input Tinggi Badan:**
 - o Program mencetak prompt "Masukkan Tinggi Badan (m): " dan menerima input tinggi badan dari pengguna melalui `fmt.Scan(&tinggibadan)`.

4. Perhitungan BMI:

- BMI dihitung dengan rumus: $bmi = \text{beratbadan} / (\text{tinggibadan} * \text{tinggibadan})$, di mana tinggi badan dikuadratkan.

5. Output BMI:

- Program mencetak hasil BMI dengan format dua angka desimal menggunakan `fmt.Printf("BMI anda: %.2f", bmi)`.

Soal Latihan

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 int

    fmt.Print("Masukkan total belanja awal: ")
    fmt.Scan(&totalBelanja)

    fmt.Print("Masukkan besaran diskon (dalam persen): ")
    fmt.Scan(&diskon)

    totalSetelahDiskon := totalBelanja - (totalBelanja * diskon / 100)

    fmt.Printf("Total belanja setelah diskon: %d\n", totalSetelahDiskon)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\Modul4\soal1\soal1.go"
Masukkan total belanja awal: 100000
Masukkan besaran diskon (dalam persen): 10
Total belanja setelah diskon: 90000
PS D:\Coding manja> go run "d:\Coding manja\Modul4\soal1\soal1.go"
Masukkan total belanja awal: 200000
Masukkan besaran diskon (dalam persen): 20
Total belanja setelah diskon: 160000
PS D:\Coding manja> go run "d:\Coding manja\Modul4\soal1\soal1.go"
Masukkan total belanja awal: 150000
Masukkan besaran diskon (dalam persen): 15
Total belanja setelah diskon: 127500
```

Deskripsi Program:

1. **Import Package:** Program mengimpor package fmt untuk input dan output.
2. **Deklarasi Variabel:** Dua variabel totalBelanja dan diskon dideklarasikan sebagai tipe integer.
3. **Input Total Belanja:** Program meminta pengguna untuk memasukkan total belanja awal dengan fmt.Scan(&totalBelanja).

4. **Input Besaran Diskon:** Program meminta pengguna untuk memasukkan besaran diskon dalam persen dengan `fmt.Scan(&diskon)`.
5. **Perhitungan Total Setelah Diskon:**
 - Diskon dihitung dengan rumus: $\text{totalBelanja} * \text{diskon} / 100$.
 - Total belanja setelah diskon adalah $\text{totalBelanja} - (\text{totalBelanja} * \text{diskon} / 100)$.
6. **Output:** Program mencetak total belanja setelah diskon dengan menggunakan `fmt.Printf`.

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 (dalam meter): ")
    fmt.Scan(&tinggiBadan)

    beratBadan = bmi * (tinggiBadan * tinggiBadan)
    fmt.Printf("Berat badan seseorang: %.0f kg\n", beratBadan)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\Modul4\soal2\soal2.go"
Masukkan nilai BMI: 22.85
Masukkan tinggi badan (dalam meter): 1.75
Berat badan seseorang: 70 kg
PS D:\Coding manja> go run "d:\Coding manja\Modul4\soal2\soal2.go"
Masukkan nilai BMI: 23.43
Masukkan tinggi badan (dalam meter): 1.6
Berat badan seseorang: 60 kg
```

Deskripsi Program:

1. **Deklarasi Variabel:** Tiga variabel bertipe float64 dideklarasikan:
 - bmi: untuk menyimpan nilai BMI.
 - tinggiBadan: untuk menyimpan tinggi badan dalam meter.
 - beratBadan: untuk menyimpan hasil perhitungan berat badan.
2. **Input Nilai BMI:** Program meminta pengguna untuk memasukkan nilai BMI.
3. **Input Tinggi Badan:** Program meminta pengguna untuk memasukkan tinggi badan dalam meter.
4. **Perhitungan Berat Badan:**
 - Berat badan dihitung dengan rumus: $\text{beratBadan} = \text{bmi} * (\text{tinggiBadan} * \text{tinggiBadan})$.
5. **Output:** Program mencetak berat badan sebagai bilangan bulat (integer) dengan format %.0f.

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 distance(x1, y1, x2, y2 float64) float64 {
    return math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
}
func main() {
    var ax, ay, bx, by, cx, cy float64
    fmt.Scan(&ax, &ay)
    fmt.Scan(&bx, &by)
    fmt.Scan(&cx, &cy)
    ab := distance(ax, ay, bx, by)
    bc := distance(bx, by, cx, cy)
    ca := distance(cx, cy, ax, ay)
    longest := math.Max(ab, math.Max(bc, ca))
    fmt.Printf("%.2f\n", longest)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\Modul4\soal3\soal3.go"
1.0
1.0
4.0
1.0
1.0
5.0
5.00
```

Deskripsi Program:

- Fungsi distance:**
 - Fungsi ini menerima empat parameter bertipe float64 yang mewakili koordinat dua titik (x1, y1) dan (x2, y2).
 - Menghitung jarak antara dua titik menggunakan rumus jarak Euclidean:
$$\text{distance} = \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$$
 - Mengembalikan nilai jarak sebagai float64.
- Fungsi main:**

- Deklarasi variabel ax, ay, bx, by, cx, cy bertipe float64 untuk menyimpan koordinat titik A, B, dan C.
 - Program menggunakan fmt.Scan untuk membaca input koordinat dari pengguna. Pengguna diminta untuk memasukkan koordinat dalam format x y untuk ketiga titik.
3. **Menghitung Panjang Sisi Segitiga:**
- Program memanggil fungsi distance untuk menghitung panjang sisi segitiga:
 - ab: panjang sisi antara titik A dan B.
 - bc: panjang sisi antara titik B dan C.
 - ca: panjang sisi antara titik C dan A.
4. **Menentukan Sisi Terpanjang:**
- Program menggunakan math.Max untuk membandingkan panjang sisi-sisi yang telah dihitung dan menentukan sisi terpanjang (longest).
5. **Output:**
- Program mencetak panjang sisi terpanjang dengan format dua angka di belakang koma menggunakan fmt.Printf.

DAFTAR PUSTAKA

<https://dasarprogramangolang.novalagung.com/A-tipe-data.html>