

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



**DISUSUN OLEH: ANGGUN  
WAHYU WIDIYANA  
103112480280  
S1 IF-12-01  
DOSEN:  
Yohani Setiya Rafika Nur, M. Kom.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024/202**

## **DASAR TEORI**

### **1. Tipe Data**

Tipe data adalah klasifikasi atau kategorisasi dari suatu data yang menentukan jenis nilai yang dapat disimpan dan dimanipulasi oleh program komputer. Tipe data mendefinisikan operasi yang dapat dilakukan pada data, makna dari data, dan cara nilai dari tipe data tersebut dapat disimpan.

Dalam pemrograman Go, tipe data dibagi menjadi dua kategori utama: tipe data primitif dan tipe data komposit.

#### **1. Tipe Data Primitif**

Seperti Tipe data primitif adalah tipe data dasar yang disediakan oleh bahasa pemrograman dan tidak dapat dipecah menjadi tipe data yang lebih sederhana.

Dalam Go, tipe data primitif meliputi:

- a) Numerik: terdapat tiga jenis yaitu Integer, Floating-point, Complex  
Integer: Menyimpan bilangan bulat tanpa bagian desimal.  
int8, int16, int32, int64 : Bilangan bulat bertanda dengan ukuran 8, 16, 32, dan 64 bit. uint8, uint16, uint32, uint64: Bilangan bulat tak bertanda dengan ukuran 8, 16, 32, dan 64 bit.  
Floating-point: Menyimpan bilangan dengan bagian desimal. Float terbagi menjadi dua yaitu float32 : Presisi tunggal (6-9 digit desimal) dan float64 : Presisi ganda (15-17 digit desimal).  
Complex : Menyimpan bilangan kompleks. Ada complex64 : Terdiri dari dua float32 dan complex128: Terdiri dari dua float64.
- b) Boolean: Menyimpan nilai kebenaran, hanya dapat berupa true atau false.
- c) String: Menyimpan urutan karakter, direpresentasikan sebagai slice byte yang tidak dapat diubah.

#### **2. Tipe Data Komposit**

Tipe data komposit adalah tipe data yang terbentuk dari kombinasi tipe data primitif atau tipe data komposit lainnya untuk membentuk struktur data yang lebih kompleks.

Dalam Go, tipe data komposit meliputi:

a. Array

Kumpulan elemen dengan tipe data yang sama dan jumlah yang tetap.

Contoh: ``var numbers [5]int`

b. Slice

Bagian dari array yang ukurannya bisa berubah, lebih fleksibel daripada array.

Contoh: ``numbers := []int{1, 2, 3, 4, 5}`

c. Map

Kumpulan pasangan key-value, di mana setiap key bersifat unik.

Contoh: ``scores := map[string]int{"Alice": 98, "Bob": 87}`

d. Struct

Kumpulan field dengan tipe data yang bisa berbeda-beda, memungkinkan pengelompokan data yang berkaitan.

Contoh:

```
type Person struct {  
    Name string  
    Age  int  
}
```

e. Interface

Kumpulan method signature yang mendefinisikan sebuah set perilaku.

Contoh:

```
type Writer interface {  
    Write([]byte) (int, error)  
}
```

## 2. Variabel

Variabel adalah tempat penyimpanan data dalam memori komputer yang memiliki nama dan tipe data tertentu, di mana nilainya dapat berubah selama eksekusi program.

Deklarasi variabel dalam Go dapat dilakukan dengan beberapa cara

a. Deklarasi eksplisit

```
```go
var nama string
var umur int
```
```

b. Deklarasi dengan inisialisasi

```
```go
var nama string = "Budi"
var umur int = 25
```
```

c. Deklarasi singkat (short declaration)

```
```go
nama := "Budi"
umur := 25
```
```

Penggunaan variabel memungkinkan untuk:

1. Menyimpan dan memanipulasi data selama program berjalan.
2. Membuat kode yang lebih mudah dibaca dengan menggunakan nama yang deskriptif.
3. Menggunakan kembali nilai yang sama di berbagai bagian program.

### 3. Konstanta

Konstanta adalah nilai tetap yang dideklarasikan sekali dan tidak dapat diubah selama eksekusi program. Konstanta digunakan untuk menyimpan nilai yang tidak berubah, seperti nilai matematika atau konfigurasi program.

Dalam Go, konstanta dideklarasikan menggunakan keyword `const`.

Contoh deklarasi konstanta:

```
```go
const PI = 3.14159
const GRAVITY = 9.8
```
```

Konstanta juga dapat dideklarasikan dalam blok:

```
```go
const (
    MONDAY    = "Senin"
    TUESDAY   = "Selasa"
    WEDNESDAY = "Rabu"
)
```
```

Go juga mendukung konstanta yang tidak ditentukan tipe datanya secara eksplisit (untyped constants), yang memungkinkan fleksibilitas dalam penggunaannya. Konstanta tanpa tipe dapat digunakan dalam ekspresi dengan tipe yang berbeda tanpa perlu konversi eksplisit.

Penggunaan konstanta memiliki beberapa keuntungan:

1. Meningkatkan keamanan program dengan mencegah perubahan nilai yang tidak diinginkan.
2. Meningkatkan keterbacaan kode dengan memberikan nama yang bermakna pada nilai-nilai tetap.
3. Memudahkan pemeliharaan kode, karena perubahan nilai konstanta hanya perlu dilakukan di satu tempat

## CONTOH SOAL

### Latihan1

Buatlah program yang digunakan untuk menghitung hasil penjumlahan 5 bilangan bulat.

**Masukan** terdiri dari lima bilangan bulat a, b, c, d, dan e.

**Keluaran** berupa bilangan hasil penjumlahan lima bilangan bulat a, b, c, d, dan

Contoh masukan dan keluaran:

| No | Masukkan       | Keluaran |
|----|----------------|----------|
| 1. | 3 2 7 10 2     | 24       |
| 2. | 11 22 33 44 55 | 165      |

Source Code:

```
package main

import "fmt"

func main() {
    var hasil, a, b, c, d, e int
    fmt.Scan(&a, &b, &c, &d, &e)
    hasil = a + b + c + d + e
    fmt.Println(hasil)
}
```

Output:

```
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Cosomod1\Cosomod2\Coso1.go"
3 2 7 10 2
24
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Cosomod1\Cosomod2\Coso1.go"
11 22 33 44 55
165
PS C:\Users\User\go\pkg> |
```

Deskripsi Program:

Program Go di atas adalah program untuk menghitung jumlah dari lima bilangan bulat dan menampilkan hasilnya.

## Latihan 2

Sebuah program digunakan untuk menghitung persamaan  $f(x) = \frac{2}{x+5} + 5$

**Masukan** terdiri dari sebuah bilangan bulat.

**Keluaran** berupa bilangan yang menyatakan nilai dari  $f(x)$ .

Contoh masukan dan keluaran:

| No | Masukkan | Keluaran            |
|----|----------|---------------------|
| 1. | 5        | 24                  |
| 2. | -23      | 4.88888888888888889 |

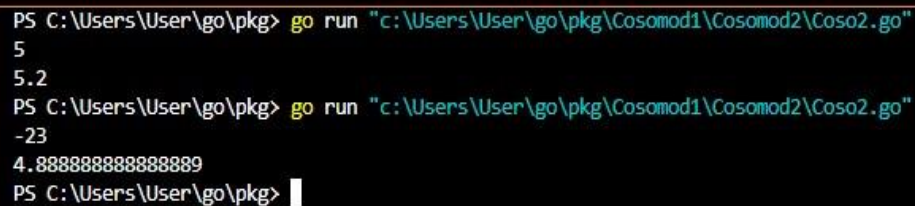
Source Code:

```
package main

import "fmt"

func main() {
    var x, fx float64
    fmt.Scan(&x)
    fx = 2/(x+5) + 5
    fmt.Print(fx)
}
```

Output :



```
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Cosomod1\Cosomod2\Coso2.go"
5
5.2
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Cosomod1\Cosomod2\Coso2.go"
-23
4.88888888888888889
PS C:\Users\User\go\pkg>
```

Deskripsi Program:

Program Go di atas adalah program yang menghitung nilai fungsi matematis berdasarkan input pengguna.

### Latihan 3

Tipe karakter sebenarnya hanya apa yang tampak dalam tampilan. Di dalamnya tersimpan dalam bentuk biner 8 bit (byte) atau 32 bit (rune) saja. Buat program ASCII yang akan membaca 5 buah data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII)

**Masukan** terdiri dari dua baris. Baris pertama berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

**Keluaran** juga terdiri dari dua baris. Baris pertama berisi 5 buah representasi karakter dari data yang diberikan, yang berdampingan satu dengan lain, tanpa dipisahkan spasi. Baris kedua berisi 3 buah karakter (juga tidak dipisahkan oleh spasi).

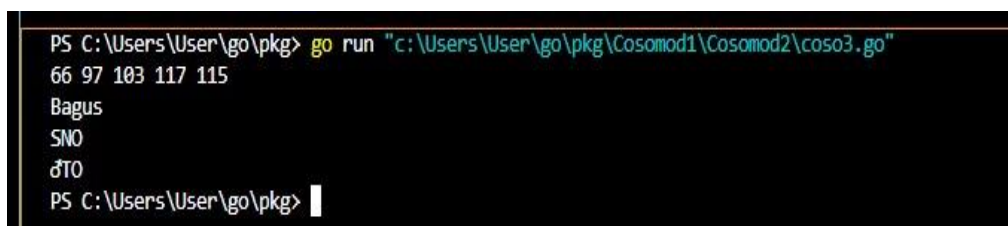
Contoh masukan dan keluaran:

| No | Masukkan          | Keluaran |
|----|-------------------|----------|
| 1. | 66 97 103 117 115 | Bagus    |
| 2. | SNO               | TOP      |

Source Code:

```
Package main
import "fmt"
func main() {
    // 1. input 5 buah byte, output 5 buah karakter
    var c1, c2, c3, c4, c5 byte
    fmt.Scan(&c1, &c2, &c3, &c4, &c5)
    fmt.Printf("%c%c%c%c%c", c1, c2, c3,
    c4, c5) fmt.Println()
    // 2. input 3 buah rune, output karakter
    setelahnya var b1, b2, b3 rune
    fmt.Scanf("%c%c%c", &b1, &b2, &b3)
    fmt.Printf("%c%c%c", b1+1, b2+1, b3+1)
}
```

Output:



```
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Cosomod1\Cosomod2\coso3.go"
66 97 103 117 115
Bagus
SNO
TOP
PS C:\Users\User\go\pkg>
```



Deskripsi Program:

Program Go di atas bertujuan untuk mengambil lima karakter byte dari pengguna dan mencetaknya dan mengambil tiga karakter rune dari pengguna, lalu mencetak karakter dari rune yang diinput.

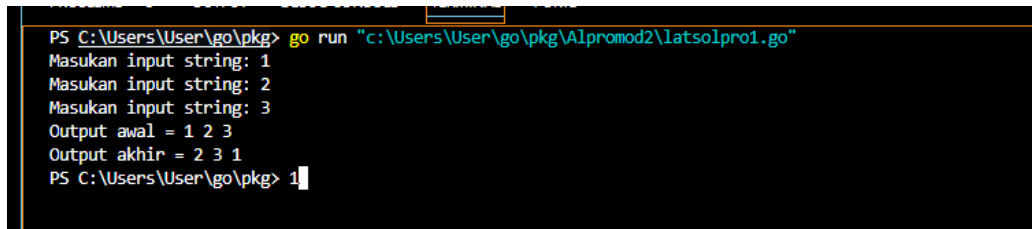
## SOAL LATIHAN (Statement Perulangan)

### Latihan 1

Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?:

```
1) package main
2) import "fmt"
3)
4) func main() {
5)     var (
6)         satu, dua, tiga string
7)         temp string
8)     )
9)     fmt.Print("Masukan input string: ")
10)    fmt.Scanln(&satu)
11)    fmt.Print("Masukan input string: ")
12)    fmt.Scanln(&dua)
13)    fmt.Print("Masukan input string: ")
14)    fmt.Scanln(&tiga)
15)    fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
16)    temp = satu
17)    satu = dua
18)    dua = tiga
19)    tiga = temp
20)    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
21) }
```

Output:



```
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Alpromod2\latsolpro1.go"
Masukan input string: 1
Masukan input string: 2
Masukan input string: 3
Output awal = 1 2 3
Output akhir = 2 3 1
PS C:\Users\User\go\pkg> 1
```

Deskripsi Program:

Program Go di atas mengelola tiga input string dan melakukan rotasi nilai antar variable.

## Latihan 2

Buatlah program yang digunakan untuk menampilkan data biodata yang diberikan dari masukan pengguna, kemudian menampilkan data tersebut!

**Masukan** terdiri dari 3 buah teks yang menyatakan nama, nim dan kelas.

**Keluaran** menampilkan resume singkat mahasiswa sesuai masukan yang diberikan.

Catatan: tidak ada batasan terkait resume yang akan ditampilkan. Silahkan berkreasi.

Source Code:

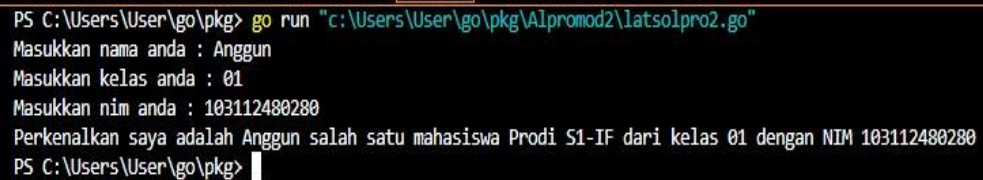
```
package main

import "fmt"

func main() {
    var nama, nim, kelas string
    fmt.Print("Masukkan nama anda : ")
    fmt.Scan(&nama)
    fmt.Print("Masukkan kelas anda : ")
    fmt.Scan(&kelas)
    fmt.Print("Masukkan nim anda : ")
    fmt.Scan(&nim)

    fmt.Println("Perkenalkan saya adalah " + nama + " salah satu mahasiswa Prodi S1-IF dari kelas " + kelas + " dengan NIM " + nim)
}
```

Output :



```
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\A1promod2\latsolpro2.go"
Masukkan nama anda : Anggun
Masukkan kelas anda : 01
Masukkan nim anda : 103112480280
Perkenalkan saya adalah Anggun salah satu mahasiswa Prodi S1-IF dari kelas 01 dengan NIM 103112480280
PS C:\Users\User\go\pkg> |
```

Deskripsi Program:

Program Go di atas bertujuan untuk meminta informasi pengguna dan kemudian mencetak perkenalan berdasarkan informasi tersebut.

### Latihan 3

Sebuah program digunakan untuk menghitung luas lingkaran berdasarkan panjang jari- jari.

**Masukan** terdiri dari sebuah bilangan riil yang menyatakan jari-jari lingkaran.

**Keluaran** berupa hasil perhitungan yang menyatakan luas lingkaran.

**Contoh masukan dan keluaran:**

| No | Masukkan | Keluaran |
|----|----------|----------|
| 1. | 7        | 153.9    |
| 2. | 14       | 615.8    |
| 3. | 20       | 1256.6   |

Source Code:

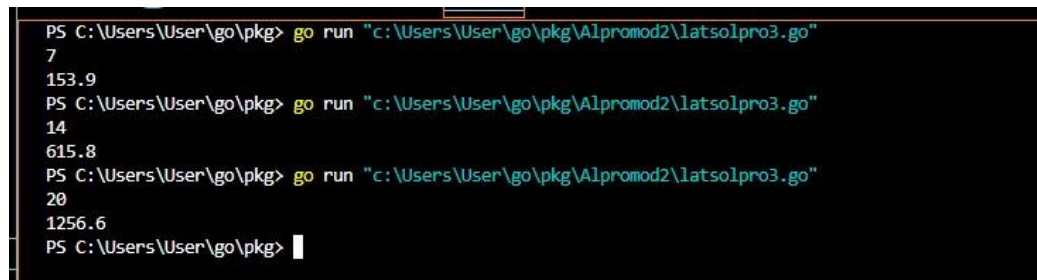
```
package main

import (
    "fmt"
    "math"
)

func main() {
    var r, luas float64
    fmt.Scan(&r)
    luas = math.Pi * math.Pow(r, 2)

    fmt.Printf("%.1f\n", luas)
}
```

Output :



```
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Alpromod2\latsolpro3.go"
7
153.9
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Alpromod2\latsolpro3.go"
14
615.8
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Alpromod2\latsolpro3.go"
20
1256.6
PS C:\Users\User\go\pkg> █
```

Deskripsi Program:

Program Go di atas menghitung luas lingkaran berdasarkan jari-jari yang diinput pengguna.

#### Latihan 4

Sebuah program digunakan untuk melakukan konversi suhu dari Fahrenheit (F) ke Celcius (C) dengan persamaan sebagai berikut:  $F = C \times \frac{9}{5} + 32$

**Masukan** terdiri dari bilangan bulat yang menyatakan suhu dalam satuan Fahrenheit.

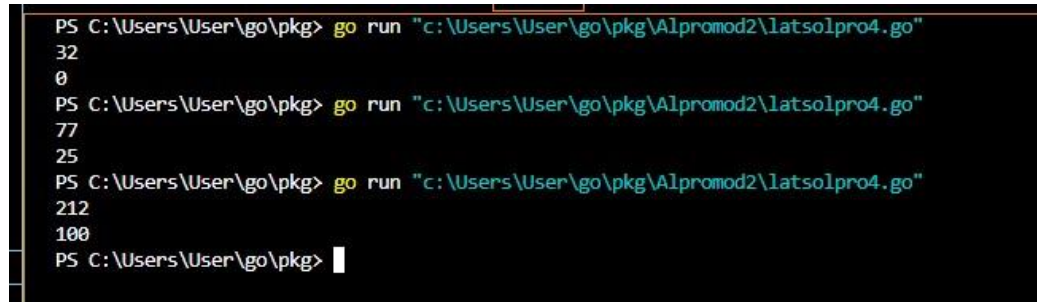
**Keluaran** berupa suhu dalam satuan Celcius.

| No | Masukkan | Keluaran |
|----|----------|----------|
| 1. | 32       | 0        |
| 2. | 77       | 25       |
| 3. | 212      | 100      |

Source Code:

```
package main
import "fmt"
func main() {
    var f, c int
    fmt.Scan(&f)
    c = (f - 32) * 5 / 9
    fmt.Println(c)
}
```

Output :



```
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Alpromod2\latsolpro4.go"
32
0
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Alpromod2\latsolpro4.go"
77
25
PS C:\Users\User\go\pkg> go run "c:\Users\User\go\pkg\Alpromod2\latsolpro4.go"
212
100
PS C:\Users\User\go\pkg> 
```

Deskripsi Program:

Program Go di atas bertujuan untuk mengambil suhu dalam Fahrenheit dari masukan pengguna dan mengubahnya ke Celsius.

## DAFTAR PUSTAKA

- 1 Raharjo, Budi. 2022. *Mudah Belajar Go: Pemrograman Golang untuk Pemula*. Bandung.
  - 2 Heriyanto. 2023. *Pemrograman Go: Konsep & Implementasi*. Jakarta: Elex Media Komputindo.
  - 3 Utomo, Eko Kurniawan. 2021. *Pemrograman Go: Dari Dasar Sampai Mahir*. Yogyakarta: Andi Publisher.
  - 4 Donovan, Alan A. A., & Kernighan, Brian W. 2016. *The Go Programming Language*. Addison-Wesley Professional.
-