

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 5**  
**“TIPE DATA & VARIABEL”**



**DISUSUN OLEH:**  
**Keishin Naufa Alfaridzhi**  
**103112400061**  
**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

### A. Bahasa Yang Digunakan

Pada praktikum ini bahasa pemrograman yang digunakan adalah bahasa pemrograman Go, sesuai dengan modul yang menjadi acuan praktikum. Golang (atau Go) adalah bahasa pemrograman baru, yang mulai dilirik oleh para developer karena kelebihan-kelebihan yang dimilikinya. Sudah banyak Perusahaan besar yang menggunakan bahasa ini untuk produk-produk mereka hingga di level production.

### B. Komentar

Komentar biasa dimanfaatkan untuk menyisipkan catatan pada kode program, menulis penjelasan atau deskripsi mengenai suatu blok kode, atau bisa juga digunakan untuk me-remark kode (men-non-aktifkan kode yang tidak digunakan). Komentar akan diabaikan Ketika kompilasi maupun eksekusi program.

Ada 2 jenis komentar di Golang, yaitu inline dan multiline.

#### 1. Komentar Inline

Penulisan komentar jenis ini diawali dengan tanda *double slash* (*//*) lalu diikuti pesan komentarnya. Komentar inline hanya berlaku untuk satu baris pesan saja. Jika pesan komentar lebih dari satu baris, maka tanda *double slash* harus ditulis lagi di baris selanjutnya.

#### 2. Komentar Multiline

Komentar yang cukup panjang akan lebih rapi jika ditulis menggunakan teknik komentar multiline. Ciri dari komentar jenis ini adalah penulisannya diawali dengan tanda *(/\**) dan diakhiri *(\*/)*.

### C. Variabel

Golang mengadopsi 2 jenis penulisan variabel, yang dituliskan tipe data-nya dan yang tidak. Kedua cara tersebut intinya adalah sama, pembedanya hanyalah cara penulisannya saja. Untuk penulisan variabel dengan tipe data, keyword *var* digunakan untuk deklarasi variabel kemudian diakhiri dengan tipe data misalnya *string*. Kemudian untuk penulisan variabel tanpa tipe data, variabel dideklarasikan dengan menggunakan metode type inference. Penandanya tipe data tidak dituliskan pada saat deklarasi. Pada penggunaan metode ini, operand (*=*) harus diganti dengan (*:=*) dan keyword *var* dihilangkan.

Golang memiliki aturan unik yang tidak dimiliki bahasa lain, yaitu tidak boleh ada satupun variabel yang menganggur. Artinya, semua variabel yang dideklarasikan harus digunakan. Jika terdapat variabel yang tidak digunakan tapi dideklarasikan, program akan gagal dikompilasi. Untuk mengatasi itu, golang memiliki variabel yaitu underscore. Underscore ( \_ ) adalah predefined variabel yang bisa dimanfaatkan untuk menampung nilai yang tidak dipakai.

#### **D. Tipe Data**

Golang mengenal beberapa jenis tipe data, diantaranya adalah tipe data numerik (decimal dan non-desimal), string, dan boolean.

1. Tipe Data Numerik Non-Desimal (uint, int)
2. Tipe Data Numerik Desimal (float64, float32)
3. Tipe Data Bool (true, false)
4. Tipe Data String (string, “ “)

#### **E. Operator Aritmatika**

Operator aritmatika merupakan operator yang digunakan untuk operasi yang sifatnya perhitungan. Golang mendukung beberapa operator aritmatika standar, yaitu:

1. Penjumlahan (+)
2. Pengurangan (-)
3. Perkalian (\*)
4. Pembagian (/)
5. Modulus atau sisa hasil pembagian (%)

#### **F. Seleksi Kondisi**

Seleksi kondisi pada program berguna untuk mengontrol sebuah blok kode yang akan dieksekusi. Yang dijadikan acuan oleh seleksi kondisi adalah nilai bertipe bool, bisa berasal dari variabel, ataupun hasil operasi perbandingan. Nilai tersebut menentukan blok kode mana yang akan dieksekusi. Go memiliki 2 macam keyword untuk seleksi kondisi, yaitu if else dan switch.

##### **1. If Expression**

If adalah salah satu kata kunci yang digunakan dalam percabangan. Percabangan artinya kita bisa mengeksekusi kode program tertentu ketika suatu kondisi terpenuhi. Hampir semua bahasa pemrograman mendukung if expression.

##### **2. Else if expression**

Terkadang kita butuh membuat beberapa kondisi. Kasus seperti ini dapat menggunakan else if expression. If mendukung short statement sebelum kondisi.

Hal ini sangat cocok untuk membuat statement yang sederhana sebelum melakukan pengecekan terhadap kondisi.

## **G. Perulangan**

Perulangan merupakan proses mengulang dan mengeksekusi blok kode tanpa henti sesuai dengan kondisi yang dijadikan acuan. Biasanya disiapkan variabel untuk iterasi atau penanda kapan perulangan akan dihentikan.

### **a. For Loop**

For loop merupakan statement perulangan dasar dan cukup sering ditemui. Format for loop yaitu sebagai berikut.

- *Init Statement*: bagian ini akan dieksekusi sebelum perulangan dimulai. Biasanya diisi dengan mendeklarasi variabel iterasi.
- *Condition Expression*: bagian ini akan dicek dan dieksekusi setiap perulangan yang dilakukan, jika true maka perulangan akan terus berjalan hingga kondisi bernilai false.
- *Post Statement*: statement ini akan dieksekusi pada akhir iterasi. Jika terdapat range, maka perulangan akan dieksekusi untuk setiap item pada range.

## CONTOH SOAL

1. Buatlah program untuk menampilkan baris bilangan dari  $a$  sampai dengan  $b$ .

**Masukan** terdiri dari dua bilangan bulat  $a$  dan  $b$ , yang mana  $a \leq b$ .

**Keluaran** berupa baris bilangan dari  $a$  sampai dengan  $b$ .

### Source Code:

```
package main
import "fmt"
func main() {
    var a, b int
    var j int
    fmt.Scan(&a, &b)
    for j = a; j <= b; j++ {
        fmt.Print(j, " ")
    }
}
```

### Output:

```
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\modul5\coso1\tempCodeRu
nnerFile.go"
2 5
2 3 4 5
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\modul5\coso1\tempCodeRu
nnerFile.go"
6 6
6
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\modul5\coso1\tempCodeRu
nnerFile.go"
-5 7
-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
```

2. Buatlah program yang digunakan untuk menampilkan sejumlah  $n$  luas segitiga, apabila diketahui sisi alas dan tinggi dari masing-masing segitiga.

**Masukan** terdiri dari  $n + 1$  baris. Baris pertama merupakan suatu bilangan bulat  $n$ , selanjutnya  $n$  baris berikutnya masing-masing merupakan panjang sisi alas dan tinggi dari segitiga

**Keluaran** terdiri dari  $n$  baris, yang masing-masing menyatakan luas dari segitiga.

### Source Code:

```
package main
import "fmt"
func main() {
    var j, alas, tinggi, n int
    var luas float64
    fmt.Scan(&n)
    for j = 1; j <= n; j += 1 {
        fmt.Scan(&alas, &tinggi)
        luas = 0.5 * float64(alas*tinggi)
        fmt.Println(luas)
    }
}
```

```
}  
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\Labrak4>go run "d:\Tugas\Alpro\Praktikum\Labrak4\modul5\coso2\coso2.go"  
5  
11 2  
11  
32 14  
224  
6 2  
6  
15 15  
112.5  
20 35  
350
```

3. Buatlah program yang digunakan untuk menghitung hasil perkalian dua buah bilangan tanpa menggunakan operator kali "\*".

**Masukan** terdiri dari dua bilangan bulat positif.

**Keluaran** terdiri dari sebuah bilangan yang menyatakan hasil perkalian dari dua bilangan pada masukan.

**Source Code:**

```
package main  
import "fmt"  
func main() {  
    var j, hasil, v1, v2 int  
    fmt.Println("Enter two numbers:")  
    fmt.Scan(&v1, &v2)  
    for j = 1; j <= v2; j++ {  
        hasil = hasil + v1  
    }  
    fmt.Print(hasil)  
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\Labrak4>go run "d:\Tugas\Alpro\Praktikum\Labrak4\modul5\coso3\coso3.go"  
Enter two numbers:  
2 100  
200  
D:\Tugas\Alpro\Praktikum\Labrak4>go run "d:\Tugas\Alpro\Praktikum\Labrak4\modul5\coso3\coso3.go"  
Enter two numbers:  
7 6  
42
```

## LATIHAN SOAL

### 1. Latihan1

Source Code:

```
package main
import "fmt"
func main() {
    var (
        hasil, j int
    )
    fmt.Scan(&j)
    for i := 1; i <= j; i++ {
        hasil = hasil + i
    }
    fmt.Print(hasil)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso1\latso1.go"
3
6
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso1\latso1.go"
1
1
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso1\latso1.go"
5
15
D:\Tugas\Alpro\Praktikum\Laprak4>
```

Deskripsi Program:

Program pertambahan dengan statement pengulangan atau loop. Pada program ini kita menggunakan variabel  $j$  sebagai kondisi iterasi. Setelah melakukan input untuk variabel  $j$ , maka perulangan akan dieksekusi. Kemudian didalamnya terdapat kode menggunakan variabel hasil dengan rumus  $\text{hasil} = \text{hasil} + i$ , artinya kita akan menambahkan value dari variabel hasil dengan variabel  $i$  yaitu variabel iterasi hingga kondisi bernilai false. Contoh input adalah 3, maka pada perulangan pertama hasil akan ditambah dengan nilai 1, perulangan kedua akan ditambah dengan nilai 2, dan perulangan ketiga akan ditambah dengan nilai 3.

### 2. Latihan2

Source Code:

```
package main
import (
    "fmt"
```

```

    "math"
)
func main() {
    var (
        n      int
        r, t, volume float64
    )
    fmt.Scan(&n)
    for i := 0; i < n; i++ {
        fmt.Scan(&r, &t)
        volume = 1.0 / 3 * math.Pi * r * r * t
        fmt.Println("Output:", volume)
    }
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso2\latso2.go"
1
3 4
Output: 37.699111843077524

D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso2\latso2.go"
3
1 1
Output: 1.0471975511965979
2 2
Output: 8.377580409572783
3 3
Output: 28.27433388230814

```

Deskripsi Program:

Program mencari volume kerucut biasa, dengan perulangan maka kita dapat menghitung volume kerucut berkali-kali sesuai yang kita inginkan dengan input pada variabel n.

### 3. Latihan3

Source Code:

```

package main

import "fmt"

func main() {
    var (
        n, j, pow int
    )
    pow = 1
    fmt.Scan(&n, &j)
    for i := 1; i <= j; i++ {
        pow = pow * n
    }
}

```



```
    fmt.Print(pow)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso3\latso3.go"
4 2
16
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso3\latso3.go"
2 10
1024
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso3\latso3.go"
10
3
1000
```

Deskripsi Program:

Program untuk pemangkatan dengan menggunakan perulangan. Cara kerjanya adalah pertama masukkan angka yang akan dipangkatkan, dalam kasus ini adalah variabel *n*, selanjutnya masukkan angka yang menjadi pangkatnya dalam kasus ini adalah variabel *j* yang digunakan sebagai berapa kali perkalian yang akan dilakukan. contoh jika memasukkan angka 4 dan 2, maka akan menjadi  $4^2$  atau  $4 \times 4$ . Rumus untuk perpangkatan ini adalah variabel  $pow = pow * n$ . Karena variabel *pow* belum kita tentukan value nya, maka  $pow = 0$  yang akan membuat program tidak akan berjalan dengan yang diinginkan karena 0 dikali berapapun hasilnya akan 0. Untuk mengatasi hal tersebut maka kita tentukan valuenya sebelum statement *peru*

#### 4. Latihan4

Source Code:

```
package main
import "fmt"
func main() {
    var (
        n, hasil int
    )
    fmt.Scan(&n)
    hasil = 1
    for i := 1; i <= n; i++ {
        hasil *= i
    }
    fmt.Println(hasil)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso4\latso4.go"
0
1

D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso4\latso4.go"
1
1

D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso4\latso4.go"
5
120

D:\Tugas\Alpro\Praktikum\Laprak4>go run "d:\Tugas\Alpro\Praktikum\Laprak4\latso\latso4\latso4.go"
10
3628800
```

Deskripsi Program:

Program faktorial dari suatu bilangan bulat. Cara kerja program ini adalah dengan mengalikan variabel hasil dengan variabel  $i$  berulang kali hingga perulangan selesai. Pertama kita input dahulu variabel  $n$ , yaitu bilangan bulat yang akan dicari faktorialnya sekaligus sebagai kondisi perulangan. Kita juga perlu menentukan value dari variabel hasil menjadi hasil = 1 untuk mencegah perkalian 0 terjadi. Perulangan pertama variabel hasil akan dikalikan dengan variabel  $i = 1$ , perkalian kedua akan dikalikan dengan  $i = 2$ , dan seterusnya hingga kondisi  $n$  tercapai. Contoh jika kita memasukkan nilai 5 untuk  $n$ , maka variabel hasil akan dikalikan dengan variabel  $i$  sebanyak 5 kali dengan value  $i = 1$  pada perulangan pertama hingga  $i = 5$  pada perulangan ke 5.

## DAFTAR PUSTAKA

Noval Agung Prayogo. *Dasar Pemrograman Golang*. Diakses pada 01 Oktober 2024.  
<https://dasarpemrogramangolang.novalagung.com>

Annisa Nur Isnaeni. *Golang — Seleksi Kondisi*. Diakses pada 01 Oktober 2024.  
<https://medium.com/@annisaisna/golang-seleksi-kondisi-f988ead004b4>

Parvez Alam, *Golang for loop example | Golang Loops Tutorial – Phpflow.com*  
<https://medium.com/@parvez1487/golang-for-loop-example-golang-loops-tutorial-phpflow-com-f4b2b0e57944>