

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 1  
MODUL 5 & 6  
“TIPE DATA & VARIABEL”**



**DISUSUN OLEH:  
SAVILA NUR FADILLA  
103112400031  
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

### 1. Paradigma perulangan

Ada 2 jenis algoritma pengulangan yang biasa kita temui dalam pemrograman, yaitu algoritma pengulangan *counted loop* dan *uncounted loop*. Algoritma pengulangan *counted loop* merupakan algoritma pengulangan yang bisa kita gunakan apabila kita sudah mengetahui berapa kali pengulangan yang dibutuhkan. Sementara itu, algoritma pengulangan *uncounted loop* merupakan algoritma pengulangan yang bisa kita gunakan ketika kita belum mengetahui berapa jumlah pengulangan yang dibutuhkan. *Uncounted loop* ini mungkin akan kita butuhkan pada kondisi-kondisi tertentu, seperti ketika pengulangan yang dilakukan terus bergantung pada input yang kita masukkan.

### 2. For Loop (Perulangan berdasarkan iterasi)

Struktur perulangan *for* adalah jenis looping yang paling umum, dipakai untuk melakukan *counted loop*. *For loop* dipakai untuk kasus di mana jumlah iterasi diketahui. Pada kode *for* ini terdapat beberapa komponen yang dicantumkan, antara lain: (1) inisialisasi, (2) kondisi, (3) perubahan nilai, (4) statement yang diulang.

Berikut ini format sintaks untuk kode *for* :

```
for (inisialisasi; kondisi; perubahan_nilai){
    statement;
    ...
}
```

Struktur umum pengulangan *for* :

```
for (int i = 0; i < 5; i++){
    perintah
}
```

Berikut adalah aspek-aspek utama dari logika looping:

- Inisialisasi:** ini adalah langkah awal dalam looping, di mana variabel pengontrol loop diatur.
- Kondisi:** kondisi adalah pernyataan yang dievaluasi sebelum setiap iterasi loop. Jika kondisi bernilai benar (true), loop terus berjalan. Jika kondisi bernilai salah (false), loop akan berhenti. Kondisi memastikan loop tidak berjalan tanpa batas, kecuali dalam kasus tertentu, seperti sengaja dirancang untuk berjalan terus menerus.
- Loop body:** ini adalah blok kode yang diulang dalam setiap iterasi loop. Loop body bisa berisi berbagai instruksi, termasuk pemanggilan fungsi, operasi matematika, atau bahkan loop lain (nested loop).
- Update:** setelah setiap iterasi, ada langkah pembaruan di mana variabel pengontrol loop diubah. Langkah ini penting untuk memastikan kondisi exit loop pada akhirnya akan terpenuhi.

## 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  $b$ .

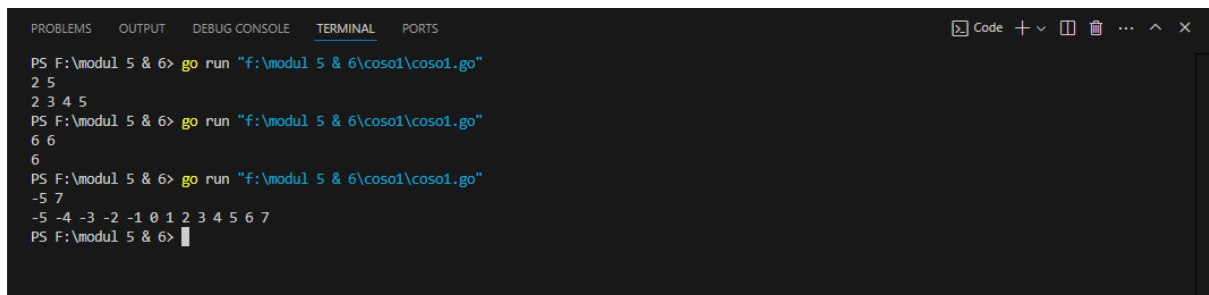
Source Code:

```
package main

import "fmt"

func main() {
    var a, b int
    var j int
    fmt.Scan(&a, &b)
    for j = a; j <= b; j+=1 {
        fmt.Print(j, " ")
    }
}
```

Output:



```
PS F:\modul 5 & 6> go run "f:\coso1\coso1.go"
2 5
2 3 4 5
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\coso1\coso1.go"
6 6
6
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\coso1\coso1.go"
-5 7
-5 -4 -3 -2 -1 0 1 2 3 4 5 6 7
PS F:\modul 5 & 6> 
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menampilkan baris bilangan dari  $a$  sampai  $b$ . Program ini meminta kita untuk memasukkan dua bilangan bulat  $a$  dan  $b$ , dimana nilai  $a \leq b$ , kemudian program akan menampilkan hasil baris bilangan dari  $a$  sampai  $b$  berdasarkan bilangan yang diinput. Keluaran berupa baris bilangan dari  $a$  sampai dengan  $b$ .

- 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.

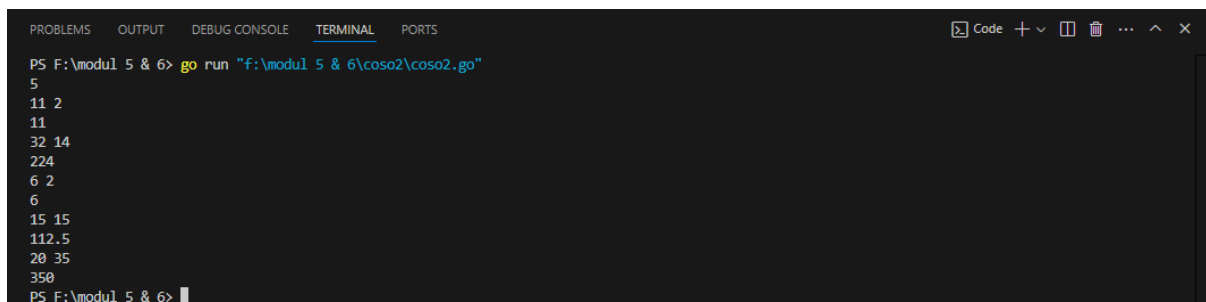
Source Code:

```
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:



```
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\coso2\coso2.go"
5
11 2
11
32 14
224
6 2
6
15 15
112.5
20 35
350
PS F:\modul 5 & 6>
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menampilkan sejumlah n luas segitiga. Input dari program ini terdiri dari  $n+1$  baris. Program ini meminta kita untuk memasukkan suatu bilangan bulat n pada baris pertama, selanjutnya n baris berikutnya masing-masing merupakan alas dan tinggi dari segitiga, kemudian program akan menghitung dan menampilkan hasilnya. Keluaran terdiri dari n baris, yang masing-masing menyatakan luas dari segitiga.

- 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.

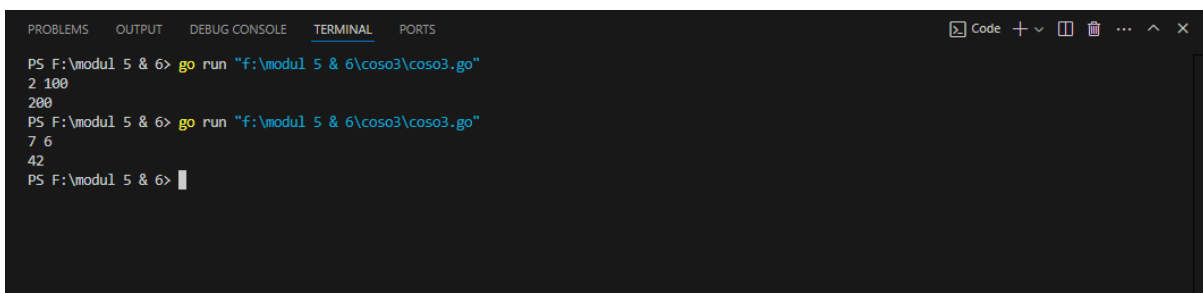
Source Code:

```
package main

import "fmt"

func main(){
    var j, hasil, v1, v2 int
    fmt.Scan(&v1, &v2)
    for j = 1; j <= v2; j++ {
        hasil = hasil + v1
    }
    fmt.Print(hasil)
}
```

Output:



```
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\coso3\coso3.go"
2 100
200
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\coso3\coso3.go"
7 6
42
PS F:\modul 5 & 6> 
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menghitung hasil perkalian dua buah bilangan tanpa memakai operator kali. Program ini meminta kita untuk memasukkan dua bilangan bulat positif, kemudian program akan mengalikan kedua bilangan tersebut dan menampilkan hasilnya. Keluaran berupa hasil perkalian dari dua bilangan berdasarkan input.

## SOAL LATIHAN

- 1.) Buatlah program untuk menjumlahkan sekumpulan bilangan. Masukan terdiri dari suatu bilangan bulat positif n. Keluaran berupa bilangan hasil penjumlahan dari 1 sampai dengan n.

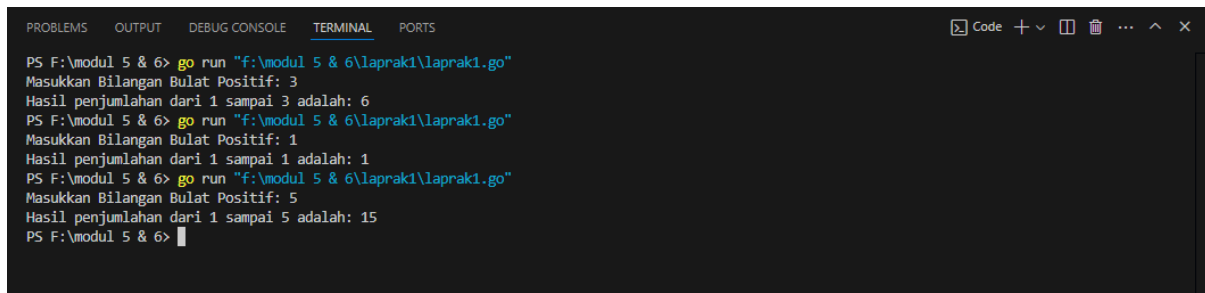
Source Code:

```
package main

import "fmt"

func main() {
    var n int
    fmt.Print("Masukkan Bilangan Bulat Positif: ")
    fmt.Scan(&n)
    hasil := 0
    for i := 1; i <= n; i++ {
        hasil += i
    }
    fmt.Printf("Hasil penjumlahan dari 1 sampai %d adalah: %d\n", n, hasil)
}
```

Output:



```
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\laprak1\laprak1.go"
Masukkan Bilangan Bulat Positif: 3
Hasil penjumlahan dari 1 sampai 3 adalah: 6
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\laprak1\laprak1.go"
Masukkan Bilangan Bulat Positif: 1
Hasil penjumlahan dari 1 sampai 1 adalah: 1
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\laprak1\laprak1.go"
Masukkan Bilangan Bulat Positif: 5
Hasil penjumlahan dari 1 sampai 5 adalah: 15
PS F:\modul 5 & 6> 
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menjumlahkan sekumpulan bilangan. Program ini meminta kita untuk memasukkan suatu bilangan bulat positif n, kemudian program akan menghitung dan menampilkan hasilnya. Keluaran berupa bilangan hasil penjumlahan dari 1 sampai dengan n.

- 2.) Buatlah program yang digunakan untuk menghitung volume sejumlah  $n$  kerucut, apabila diketahui panjang jari-jari alas kerucut dan tinggi dari kerucut. Masukan terdiri dari beberapa baris. Baris pertama adalah bilangan bulat  $n$ , selanjutnya  $n$  baris berikutnya masing-masing merupakan panjang jari-jari alas kerucut dan tinggi dari kerucut. Keluaran terdiri dari beberapa baris, yang masing-masingnya menyatakan volume dari  $n$  kerucut.

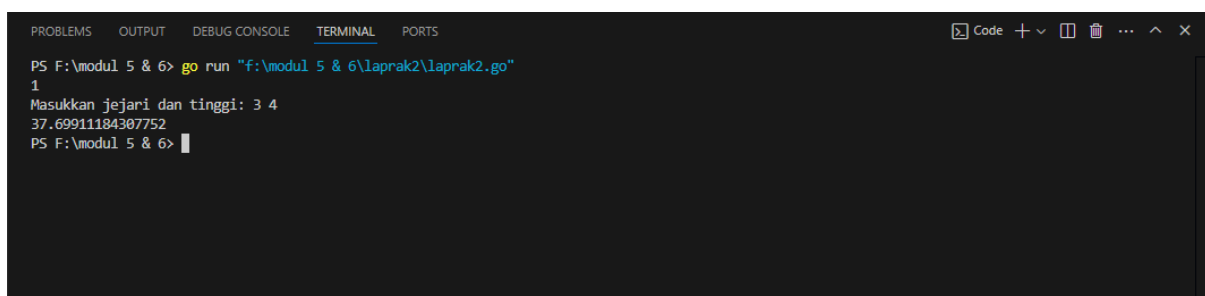
Source Code:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var n int
    var r, tinggi float64
    fmt.Scan(&n)
    for i := 0; i < n; i++ {
        fmt.Print("Masukkan jejari dan tinggi: ")
        fmt.Scan(&r, &tinggi)
        volume := (1.0 / 3.0) * math.Pi * math.Pow(r,2) * tinggi
        fmt.Printf("%.14f\n", volume)
    }
}
```

Output:



```
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\laprak2\laprak2.go"
1
Masukkan jejari dan tinggi: 3 4
37.69911184367752
PS F:\modul 5 & 6> 
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menghitung volume sejumlah  $n$  kerucut. Input dari program ini terdiri dari beberapa baris. Program ini meminta kita untuk memasukkan bilangan bulat  $n$  pada baris pertama, selanjutnya  $n$  baris berikutnya masing-masing merupakan jari-jari dari alas kerucut dan tinggi kerucut, kemudian program akan menghitung dan menampilkan hasilnya. Keluaran terdiri dari beberapa baris, yang masing-masing menyatakan volume dari  $n$  kerucut.

- 3.) Buatlah program yang digunakan untuk menghitung hasil pemangkatan dari dua buah bilangan. Program dibuat dengan menggunakan operator perkalian dan struktur kontrol perulangan.

Masukan terdiri dari dua bilangan bulat positif.

Keluaran terdiri dari suatu bilangan yang menyatakan hasil bilangan pertama dipangkatkan dengan bilangan kedua).

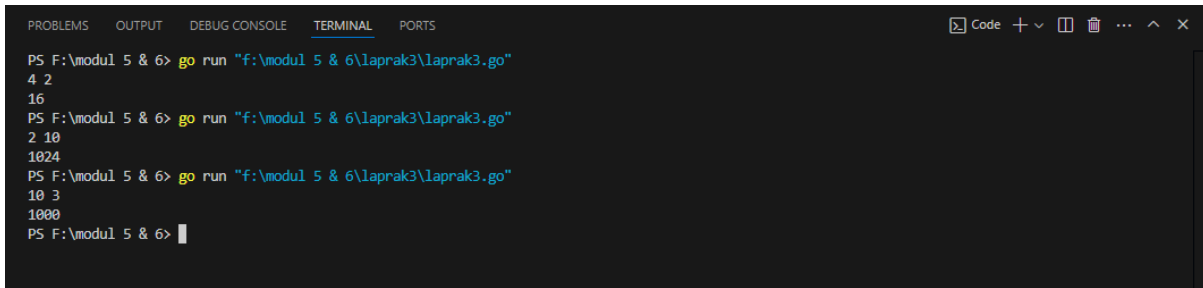
Source Code:

```
package main

import "fmt"

func main() {
    var b1, b2 int
    fmt.Scan(&b1, &b2)
    hasil := 1
    for i := 0; i < b2; i++ {
        hasil *= b1
    }
    fmt.Print(hasil)
}
```

Output:



```
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\laprak3\laprak3.go"
4 2
16
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\laprak3\laprak3.go"
2 10
1024
PS F:\modul 5 & 6> go run "f:\modul 5 & 6\laprak3\laprak3.go"
10 3
1000
PS F:\modul 5 & 6> 
```

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menghitung hasil pemangkatan dari dua buah bilangan. Program dibuat dengan menggunakan operator perkalian dan struktur kontrol perulangan. Program ini meminta kita untuk memasukkan dua bilangan bulat positif, kemudian program akan menghitung dan menampilkan hasilnya. Keluaran terdiri dari suatu bilangan yang menyatakan hasil bilangan pertama dipangkatkan dengan bilangan kedua.



- 4.) Buatlah program yang digunakan untuk menghitung hasil faktorial dari suatu bilangan. Masukan terdiri dari suatu bilangan bulat non negatif. Keluaran terdiri dari hasil faktorial dari bilangan bulat n.

Source Code:

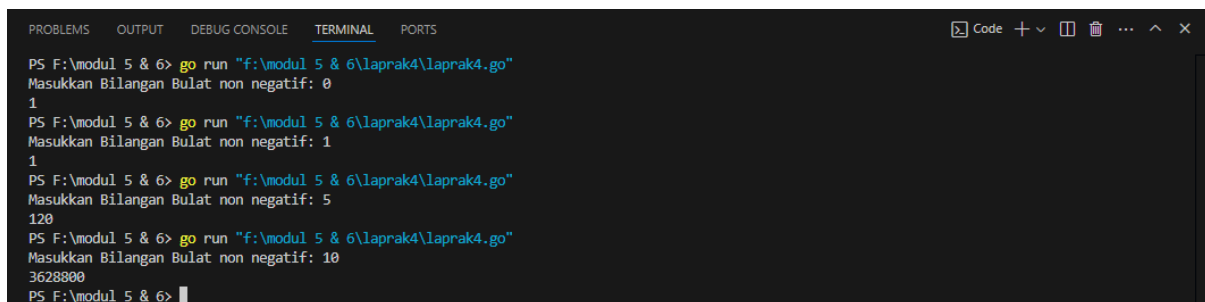
```
package main

import "fmt"

func faktorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    return n * faktorial(n-1)
}

func main() {
    var n int
    fmt.Print("Masukkan Bilangan Bulat non negatif: ")
    fmt.Scan(&n)
    hasil := faktorial(n)
    fmt.Print(hasil)
}
```

Output:

A screenshot of a terminal window with a dark background. The window has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The terminal shows the execution of a Go program. The prompt is 'PS F:\modul 5 & 6>'. The user enters 'go run "f:\modul 5 & 6\laprak4\laprak4.go"'. The program prompts 'Masukkan Bilangan Bulat non negatif: 0' and outputs '1'. Then it prompts 'Masukkan Bilangan Bulat non negatif: 1' and outputs '1'. Then it prompts 'Masukkan Bilangan Bulat non negatif: 5' and outputs '120'. Finally, it prompts 'Masukkan Bilangan Bulat non negatif: 10' and outputs '3628800'. The prompt 'PS F:\modul 5 & 6>' is visible at the bottom.

Deskripsi Program:

Program ini merupakan program sederhana bahasa Go yang bertujuan untuk menghitung hasil faktorial dari suatu bilangan. Program ini meminta kita untuk memasukkan bilangan bulat positif, kemudian program akan menghitung dan menampilkan hasilnya. Keluaran berupa hasil faktorial dari bilangan bulat positif yang diinput.

## DAFTAR PUSTAKA

Coding Studio Team. (2020). *Algoritma Pengulangan dengan For Loop*. Retrieved from Coding Studio : <https://codingstudio.id/blog/algoritma-pengulangan-for-loop/>

Revolusi Citra Edukasi. *Apa Itu Looping? Arti, Fungsi, Contoh*. Retrieved from Revou.co : <https://revou.co/kosakata/looping>

Subhi, Dian Hanifudin dkk. (2019). *Sintaks Perulangan 1*. Retrieved from polinema-programming (github) : <https://polinema-programming.github.io/09/jobsheet.html>