

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1**

MODUL 5-6

TUGAS PENDAHULUAN MODUL 5-6



Disusun Oleh :

Nafila Setyani/103112430019

12-IF-05

Asisten Praktikum :

Ayu Susilowati

Noviana Rizki Anisa Putri

Dosen Pengampu :

Yudha Islami Sulistya

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

TUGAS PENDAHULUAN

A. PRAKTIKUM

Soal Studi Case 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 .

Sourcecode

```
package main

import (
    "fmt"
)

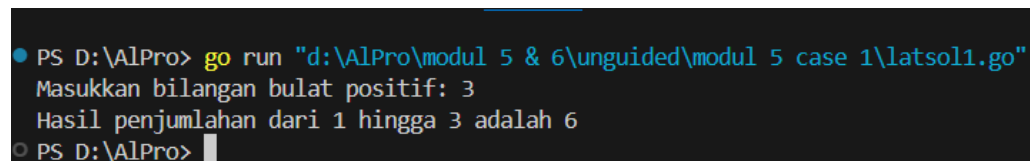
func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif: ")
    fmt.Scan(&n)

    // Inisialisasi variabel untuk menyimpan hasil penjumlahan
    sum := 0

    // Lakukan penjumlahan dari 1 hingga n
    for i := 1; i <= n; i++ {
        sum += i
    }

    // Cetak hasil penjumlahan
    fmt.Printf("Hasil penjumlahan dari 1 hingga %d adalah %d\n", n,
sum)
}
```

Screenshoot Output



```
PS D:\AlPro> go run "d:\AlPro\modul 5 & 6\unguided\modul 5 case 1\latsol1.go"
Masukkan bilangan bulat positif: 3
Hasil penjumlahan dari 1 hingga 3 adalah 6
PS D:\AlPro>
```

Deskripsi Program

Input:

- Program meminta pengguna untuk memasukkan satu bilangan bulat positif `n`.

Proses:

- Program menghitung penjumlahan dari semua bilangan bulat mulai dari 1 hingga `n` menggunakan loop.
- Penjumlahan dilakukan secara bertahap dan hasilnya disimpan dalam variabel `sum`.

Output:

- Program menampilkan hasil penjumlahan dari 1 sampai dengan `n` dalam bentuk pesan yang terformat.

Algoritma dan Cara Kerja:

1. **Inisialisasi Variabel:**
 - Variabel `n` digunakan untuk menyimpan input dari pengguna, dan variabel `sum` diinisialisasi dengan nilai 0 untuk menyimpan hasil penjumlahan.
2. **Input Data:**
 - Program meminta input bilangan bulat positif `n` dari pengguna dengan menggunakan fungsi `fmt.Scan()`.
3. **Proses Penjumlahan:**
 - Loop `for` digunakan untuk melakukan iterasi dari 1 hingga `n`.
 - Pada setiap iterasi, nilai `i` ditambahkan ke dalam variabel `sum`.
4. **Output Hasil:**
 - Setelah loop selesai, program menampilkan hasil penjumlahan dengan format "Hasil penjumlahan dari 1 hingga `n` adalah `sum`", di mana `n` dan `sum` adalah nilai yang diinput dan dihitung sebelumnya.

Contoh Cara Kerja:

Jika pengguna memasukkan angka 5:

1. Program membaca input `n = 5`.
2. Program menjumlahkan $1 + 2 + 3 + 4 + 5$, menghasilkan `sum = 15`.
3. Program mencetak hasil: "Hasil penjumlahan dari 1 hingga 5 adalah 15".

Soal Studi Case 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.

Sourcecode

```
package main

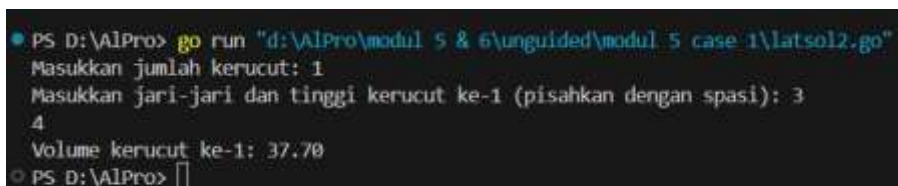
import (
    "fmt"
    "math"
)

func main() {
    var n int
    fmt.Print("Masukkan jumlah kerucut: ")
    fmt.Scan(&n)

    for i := 0; i < n; i++ {
        var r, t float64
        fmt.Printf("Masukkan jari-jari dan tinggi kerucut ke-%d\n", i+1)
        (pisahkan dengan spasi): ", i+1)
        fmt.Scan(&r, &t)

        // Rumus volume kerucut: (1/3) * π * r^2 * t
        volume := (1.0 / 3.0) * math.Pi * r * r * t
        fmt.Printf("Volume kerucut ke-%d: %.2f\n", i+1, volume)
    }
}
```

Screenshoot Output



```
PS D:\AIPro> go run "d:\AIPro\modul 5 & 6\unguided\modul 5 case 1\latsol2.go"
Masukkan jumlah kerucut: 1
Masukkan jari-jari dan tinggi kerucut ke-1 (pisahkan dengan spasi): 3
4
Volume kerucut ke-1: 37.70
PS D:\AIPro>
```

Deskripsi Program

Input:

- Baris pertama: bilangan bulat n (jumlah kerucut).
- n baris berikutnya: panjang jari-jari r dan tinggi t dari kerucut, masing-masing untuk setiap kerucut.

Proses:

- Program akan menerima input n , lalu untuk setiap kerucut, menerima input nilai r (jari-jari) dan t (tinggi).
- Menggunakan rumus volume kerucut: $\text{volume} = \frac{1}{3} \times \pi \times r^2 \times t$
- Hasil dari perhitungan volume disimpan dan dicetak untuk setiap kerucut.

Output:

- Program mencetak volume untuk setiap kerucut yang dimasukkan, dengan format desimal dua angka di belakang koma.

Algoritma dan Cara Kerja:

1. **Input Jumlah Kerucut:**
 - Program meminta pengguna memasukkan nilai n sebagai jumlah kerucut yang akan dihitung volumenya.
2. **Input Data Kerucut:**
 - Untuk setiap kerucut (loop dari 0 hingga $n-1$), program meminta input nilai jari-jari r dan tinggi t .
3. **Proses Perhitungan:**
 - Program menghitung volume kerucut menggunakan rumus $V = \frac{1}{3} \times \pi \times r^2 \times t$
4. **Output:**
 - Hasil volume dicetak dengan dua angka desimal.

Contoh Cara Kerja:

Jika pengguna memasukkan nilai 2 untuk n , kemudian memasukkan $r = 3$ dan $t = 4$ untuk kerucut pertama, serta $r = 5$ dan $t = 7$ untuk kerucut kedua, program akan menghitung dan menampilkan volume masing-masing kerucut.

Soal Studi Case 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.

Sourcecode

```
package main

import "fmt"

func main() {
    var base, exponent int

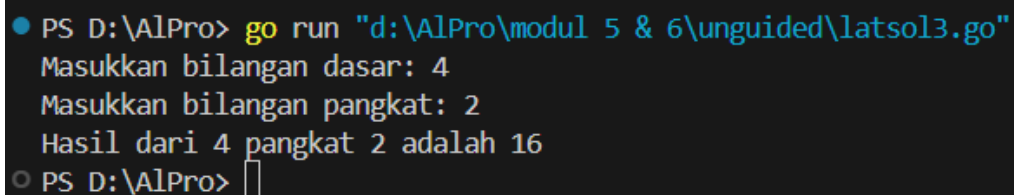
    // Input bilangan dasar dan pangkat
    fmt.Print("Masukkan bilangan dasar: ")
    fmt.Scan(&base)
    fmt.Print("Masukkan bilangan pangkat: ")
    fmt.Scan(&exponent)

    // Inisialisasi hasil dengan nilai 1 (karena perkalian identitas)
    result := 1

    // Perulangan untuk menghitung pemangkatan
    for i := 0; i < exponent; i++ {
        result *= base
    }

    // Output hasil pemangkatan
    fmt.Printf("Hasil dari %d pangkat %d adalah %d\n", base, exponent, result)
}
```

Screenshoot Output



```
PS D:\AlPro> go run "d:\AlPro\modul 5 & 6\unguided\latsol3.go"
Masukkan bilangan dasar: 4
Masukkan bilangan pangkat: 2
Hasil dari 4 pangkat 2 adalah 16
PS D:\AlPro>
```

Deskripsi Program

Input:

- Program meminta pengguna memasukkan dua bilangan bulat positif, yaitu bilangan dasar (*base*) dan pangkat (*exponent*).

Proses:

- Program menggunakan perulangan untuk mengalikan bilangan dasar (*base*) dengan dirinya sendiri sebanyak nilai pangkat (*exponent*).
- Variabel *result* diinisialisasi dengan nilai 1 dan setiap kali terjadi perulangan, nilai *base* dikalikan ke dalam *result*.

Output:

- Program menampilkan hasil pemangkatan, yaitu bilangan dasar dipangkatkan dengan bilangan pangkat.

Algoritma dan Cara Kerja:

1. **Input Data:**
 - Program meminta dua bilangan bulat positif dari pengguna, yaitu bilangan dasar *base* dan bilangan pangkat *exponent*.
2. **Inisialisasi Variabel:**
 - Variabel *result* diinisialisasi dengan nilai 1, karena dalam perkalian, nilai awal harus 1 (bukan 0) agar tidak mempengaruhi hasil akhir.
3. **Perulangan Pemangkatan:**
 - Perulangan dilakukan sebanyak nilai *exponent*. Pada setiap iterasi, *result* dikalikan dengan *base*.
4. **Output Hasil:**
 - Setelah perulangan selesai, program mencetak hasil akhir dari operasi pemangkatan dalam format "Hasil dari *base* pangkat *exponent* adalah *result*".

Contoh Cara Kerja:

Jika pengguna memasukkan bilangan dasar 2 dan bilangan pangkat 3:

1. Program membaca input *base* = 2 dan *exponent* = 3.
2. Program melakukan perulangan 3 kali:
 - Iterasi 1: $result = 1 * 2 = 2$
 - Iterasi 2: $result = 2 * 2 = 4$
 - Iterasi 3: $result = 4 * 2 = 8$
3. Program mencetak hasil: "Hasil dari 2 pangkat 3 adalah 8".

Soal Studi Case 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.

Sourcecode

```
package main

import "fmt"

func main() {
    var n int

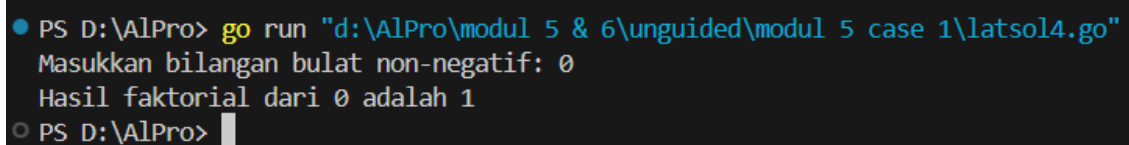
    // Input bilangan non-negatif
    fmt.Print("Masukkan bilangan bulat non-negatif: ")
    fmt.Scan(&n)

    // Inisialisasi hasil faktorial
    result := 1

    // Perulangan untuk menghitung faktorial
    for i := 1; i <= n; i++ {
        result *= i
    }

    // Output hasil faktorial
    fmt.Printf("Hasil faktorial dari %d adalah %d\n", n, result)
}
```

Screenshoot Output



```
PS D:\AlPro> go run "d:\AlPro\modul 5 & 6\unguided\modul 5 case 1\latsol4.go"
Masukkan bilangan bulat non-negatif: 0
Hasil faktorial dari 0 adalah 1
PS D:\AlPro>
```


Deskripsi Program

Input:

- Program meminta pengguna memasukkan bilangan bulat non-negatif n .

Proses:

- Program menghitung faktorial dari n menggunakan perulangan. Faktorial dari n adalah hasil perkalian semua bilangan bulat positif dari 1 hingga n .
- Faktorial didefinisikan sebagai: $n! = n \times (n-1) \times \dots \times 1$
 - Jika $n = 0$, maka faktorial dari 0 didefinisikan sebagai 1.

Output:

- Program menampilkan hasil faktorial dari bilangan n .

Algoritma dan Cara Kerja:

1. **Input Data:**
 - Program meminta bilangan bulat non-negatif n dari pengguna.
2. **Inisialisasi Variabel:**
 - Variabel `result` diinisialisasi dengan nilai 1, karena dalam perkalian, nilai awal harus 1 agar hasil faktorial benar.
3. **Proses Faktorial:**
 - Program melakukan perulangan dari 1 hingga n , dan pada setiap iterasi, `result` dikalikan dengan bilangan iterasi i .
4. **Output Hasil:**
 - Setelah perulangan selesai, program mencetak hasil akhir dari faktorial n dalam format "Hasil faktorial dari n adalah `result`".

Contoh Cara Kerja:

Jika pengguna memasukkan $n = 5$:

1. Program membaca input $n = 5$.
2. Program melakukan perulangan:
 - Iterasi 1: `result = 1 * 1 = 1`
 - Iterasi 2: `result = 1 * 2 = 2`
 - Iterasi 3: `result = 2 * 3 = 6`
 - Iterasi 4: `result = 6 * 4 = 24`
 - Iterasi 5: `result = 24 * 5 = 120`
3. Program mencetak hasil: "Hasil faktorial dari 5 adalah 120".