

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1**

MODUL 5&6

Tugas Pendahuluan Modul 5&6



Disusun Oleh :

Reza Sahrul Nuramdani/ 103112400265

IF-12-05

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. Program Untuk Menjumlahkan Sekumpulan Bilangan

```
66 //SOAL SATU MODUL 5
67 //MENJUMLAHKAN SEKUMPULAN BILANGAN
68 //INPUT bilangan positif n
69 //OUTPUT bilangan hasil penjumlahan dari 1 sampai n.
70
71 package main
72
73 import "fmt"
74
75 func main() {
76     var n, sum int
77
78     fmt.Scan(&n)
79     sum = 0
80     for i := 1; i <= n; i++ {
81         sum += i
82     }
83     fmt.Println(sum)
84 }
```

Soal Studi Case

Buatlah program untuk menjumlahkan sekumpulan bilangan.

INPUT bilangan positif n

OUTPUT bilangan hasil penjumlahan dari 1 sampai n

Sourcecode

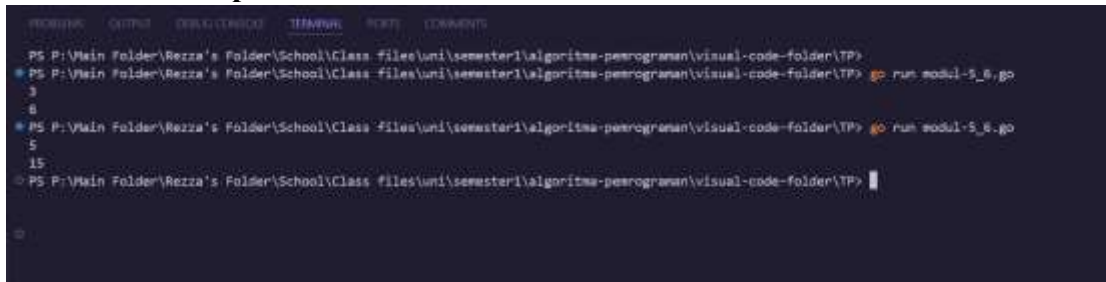
```
package main

import "fmt"

func main() {
    var n, sum int

    fmt.Scan(&n)//membaca input n, jumlah iterasi yang akan
                //dilakukan
    sum = 0 //menginisialisasi nilai sum dengan 0
    for i := 1; i <= n; i++ {
        sum += i //akan menambahkan nilai I dalam sum
    }
    fmt.Println(sum)//jika iterasi sudah sebanyak n, sesuai
                    //kondisi loop, maka akan dilakukan print
}
```

Screenshot Output



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma pemrograman\visual-code-folder\TP>
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma pemrograman\visual-code-folder\TP> go run modul-5_8.go
3
6
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma pemrograman\visual-code-folder\TP> go run modul-5_8.go
5
15
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma pemrograman\visual-code-folder\TP> |
```

Deskripsi Program

(//membaca input n, jumlah iterasi yang akan dilakukan
//menginisialisasi nilai sum dengan 0
//akan menambahkan nilai I dalam sum
//jika iterasi sudah sebanyak n, sesuai kondisi loop, maka akan dilakukan print)

B. Menghitung Volume Kerucut Sebanyak n

```
package main

import (
    "fmt"
    "math"
)

Codeium: Refactor | Explain | Generate GoDoc | X
func main() {
    var n int
    fmt.Scan(&n) //membaca input n jumlah kerucut yang akan dihitung

    for i := 0; i < n; i++ {
        var r, h float64
        fmt.Scan(&r, &h) //membaca input jari-jari alas dan tinggi kerucut
        //sesuai dengan jumlah n yang diinginkan.

        volume := (1.0 / 3.0) * math.Pi * r * r * h //rumus volume yang akan
        fmt.Println(volume) //menampilkan hasil s
    }
}
```

Soal Studi Case

Menghitung sejumlah volume n kerucut, dengan diketahui jari-jari alas, dan tinggi kerucut.

INPUT baris pertama, n, selanjutnya masing-masing panjang jari-jari dan tinggi kerucut.

OUTPUT Beberapa baris yang menyatakan volume dari n kerucut.

Sourcecode

```
package main

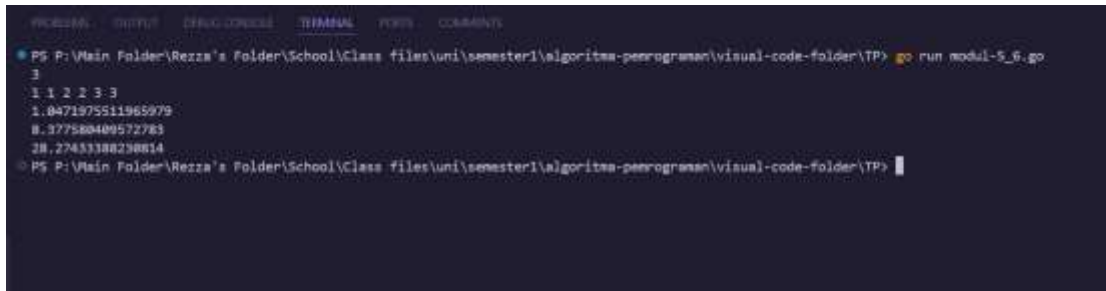
import (
    "fmt"
    "math"
)

func main() {
    var n int
    fmt.Scan(&n) //membaca input n jumlah kerucut yang akan
    dihitung

    for i := 0; i < n; i++ {
        var r, h float64
        fmt.Scan(&r, &h) //membaca input jari-jari alas
        dan tinggi kerucut, tempatnya di dalam operation for loop karena
        biar bisa menyesuaikan jumlah input
        //sesuai dengan jumlah n yang diinginkan.

        volume := (1.0 / 3.0) * math.Pi * r * r * h
        //rumus volume yang akan dieksekusi setiap eksekusi for loop
        sehabis menghitung input r dan h
        fmt.Println(volume)
        //menampilkan hasil setiap eksekusi for loop
    }
}
```

Screenshoot Output



```
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-programan\visual-code-folder\TP> go run modul-5_6.go
3
1.0471975511965979
8.377580409572783
38.27453388238814
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-programan\visual-code-folder\TP>
```

Deskripsi Program

- //membaca input n jumlah kerucut yang akan dihitung
- //membaca input jari-jari alas dan tinggi kerucut, tempatnya di dalam operation for loop karena biar bisa menyesuaikan jumlah input sesuai dengan jumlah n yang diinginkan.
- //rumus volume yang akan dieksekusi setiap eksekusi for loop sehabis menghitung input r dan h menampilkan hasil setiap eksekusi for loop.

C. Program Pemangkatan Dua Bilangan Dengan Perkalian dan Looping

```
118 package main
119
120 import (
121     "fmt"
122 )
123
124 func main() {
125     var a, n int
126     fmt.Scan(&a, &n) //membaca input a dan n (jumlah pangkat)
127
128     hasil := 1 //inisialisasi nilai hasil dengan 1
129     for i := 0; i < n; i++ {
130         hasil *= a //hasil dikali dengan nilai a, lalu value di simpa
131     }
132     fmt.Println(hasil)
133 }
134
```

Soal Studi Case

Menghitung hasil pemangkatan dari dua buah bilangan. Terdiri dari perkalian dan struktur kontrol perulangan.

INPUT dua bilangan bulat

OUTPUT suatu bilangan yang menyatakan hasil dari bilangan pertama dipangkatkan dengan bilangan kedua.

Sourcecode

```

package main

import (
    "fmt"
)

func main() {
    var a, n int
    fmt.Scan(&a, &n) //membaca input a dan n (jumlah
pangkat)

    hasil := 1 //inisialisasi nilai hasil dengan 1
    for i := 0; i < n; i++ {
        hasil *= a //hasil dikali dengan nilai a, lalu
value di simpan dalam hasil, dan dioperasikan lagi dalam iterasi
selanjutnya sebanyak n
    }
    fmt.Println(hasil)
}

```

Screenshoot Output

```

PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-pemrograman\visual-code-folder\TP> go run modul-5_6.go
modul-5_6.go:155:7: expected 'package', found 'EOF'
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-pemrograman\visual-code-folder\TP> go run modul-5_6.go
modul-5_6.go:155:7: expected 'package', found 'EOF'
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-pemrograman\visual-code-folder\TP> go run modul-5_6.go
4 2
16
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-pemrograman\visual-code-folder\TP>

```

Deskripsi Program

//membaca input a dan n (jumlah pangkat).

//inisialisasi nilai hasil dengan 1.

//hasil dikali dengan nilai a, lalu value di simpan dalam hasil, dan dioperasikan lagi dalam iterasi selanjutnya sebanyak n.

D. Menghitung Hasil Faktorial dari Suatu Bilangan

```
140 package main
141
142 import (
143     "fmt"
144 )
145
146 func main() {
147     var n int
148     fmt.Scan(&n) //membaca seberapa banyak n, yang menentukan jumlah
149
150     hasil := 1 //inisialisasi hasil dengan 1, karena faktorial 0
151     for i := 1; i <= n; i++ {
152         hasil *= i //setiap iterasi value dari hasil akan dikali
153     }
154     fmt.Println(hasil)
155 }
156
```

Codeium: Refactor | Explain | Generate GoDoc | X

Soal Studi Case

Menghitung hasil faktorial dari suatu bilangan

INPUT suatu bilangan bulat NON-negatif

OUTPUT hasil faktorial bilangan bulat.

Sourcecode

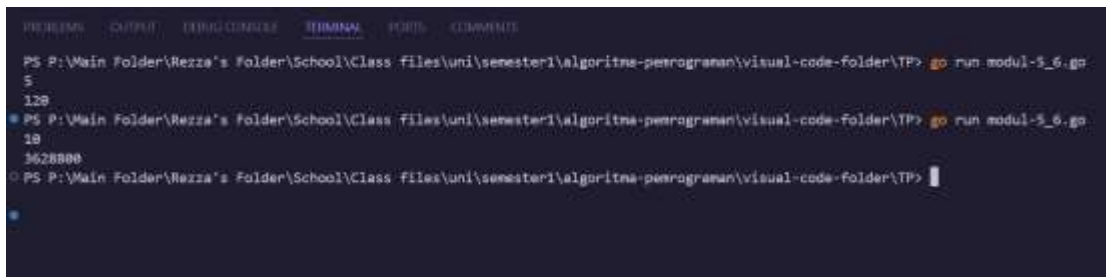
```
package main

import (
    "fmt"
)

func main() {
    var n int
    fmt.Scan(&n) //membaca seberapa banyak n, yang
    menentukan jumlah iterasi yang akan dilakukan

    hasil := 1 //inisialisasi hasil dengan 1, karena
    faktorial 0 adalah 1, dan perkalian harus 1 agar valid
    for i := 1; i <= n; i++ {
        hasil *= i //setiap iterasi value dari hasil akan
        dikalikan dengan i, yang akan naik terus nilainya sebanyak n
    }
    fmt.Println(hasil)
}
```

Screenshoot Output



```
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-pemrograman\visual-code-folder\TP> go run modul-5_6.go
5
120
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-pemrograman\visual-code-folder\TP> go run modul-5_6.go
18
3628800
PS P:\Main Folder\Rezza's Folder\School\Class files\uni\semester1\algoritma-pemrograman\visual-code-folder\TP>
```

Deskripsi Program

// membaca seberapa banyak n, yang menentukan jumlah iterasi yang akan dilakukan.
// inisialisasi hasil dengan 1, karena faktorial 0 adalah 1, dan perkalian harus 1 agar valid.
// setiap iterasi value dari hasil akan dikalikan dengan i, yang akan naik terus nilainya sebanyak n.

