

LAPORAN
PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 3
FUNGSI



Oleh:

NAMA: NUFAIL ALAUDDIN TSAQIF

NIM: 103112400084

KELAS: IF-12-01

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. DASAR TEORI

1. Definisi Function

Fungsi merupakan satu kesatuan rangkaian instruksi yang memberikan atau menghasilkan suatu nilai dan biasanya memetakan input ke suatu nilai yang lain. Oleh karena itu, fungsi selalu menghasilkan/mengembalikan nilai. Suatu subprogram dikatakan fungsi apabila:

1. Ada deklarasi tipe nilai yang dikembalikan, dan
2. Terdapat kata kunci return dalam badan subprogram.

Maka fungsi digunakan jika suatu nilai biasanya diperlukan, seperti:

- Assignment nilai ke suatu variabel
- Bagian dari ekspresi
- Bagian dari argumen suatu subprogram, dsb.

Karena itu selalu pilih nama fungsi yang menggambarkan nilai, seperti kata benda dan kata sifat. Contoh nama-nama fungsi: median, rerata, nilaiTerbesar, ketemu, selesai, ...

2. Cara Pemanggilan Function

Sama halnya dengan prosedur, pemanggilan fungsi cukup dilakukan dengan penulisan nama fungsi beserta argumen yang diminta oleh parameter dari fungsi. Perbedaannya dengan prosedur adalah fungsi bisa di-assign ke suatu variabel, menjadi bagian dari ekspresi, dan argumen dari suatu subprogram.

II. GUIDED

GUIDED 1

```
guided1 > go guided1.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var a, b int
7      fmt.Scan(&a, &b)
8      if a >= b {
9          fmt.Println(permutasi(a, b))
10     } else {
11         fmt.Println(permutasi(b, a))
12     }
13 }
14
15 func faktorial(n int) int {
16     hasil := 1
17     for i := 1; i <= n; i++ {
18         hasil *= i
19     }
20     return hasil
21 }
22 func permutasi(n, r int) int {
23     if r > n {
24         return 0
25     }
26     return faktorial(n) / faktorial(n-r)
27 }
28
```

OUTPUT:

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\
2 4
12
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> |
```

DEKSRIPSI:

Program ini menghitung permutasi dua angka yang diberikan oleh pengguna, yaitu a dan b. Jika a lebih besar atau sama dengan b, program akan memanggil fungsi permutasi(a, b); jika tidak, fungsi permutasi(b, a) dipanggil. Fungsi permutasi menggunakan rumus faktorial, yang dihitung melalui fungsi faktorial dengan mengalikan angka dari 1 hingga n untuk menghasilkan hasil yang digunakan dalam rumus permutasi.

GUIDED 2

SURCE CODE

```
guided2 > go guided2.go > ...
1 package main
2
3 import "fmt"
4
5 func celciusToFahrenheit(celcius float64) float64 {
6     return (celcius * 9.0 / 5.0) + 32
7 }
8
9 func main() {
10     var N int
11     fmt.Print("Masukan jumlah data: ")
12     _, err := fmt.Scan(&N)
13     if err != nil || N <= 0 {
14         fmt.Println("Input tidak valid, masukan angka positif.")
15         return
16     }
17     temperatures := make([]float64, N)
18
19     fmt.Println("Masukan suhu dalam celcius: ")
20     for i := 0; i < N; i++ {
21         _, err = fmt.Scan(&temperatures[i])
22         if err != nil {
23             fmt.Println("Input tidak valid, pastikan masukan angka.")
24         }
25     }
26
27     fmt.Println("Suhu dalam Fahrenheit: ")
28     for _, temp := range temperatures {
29         fmt.Printf("%.2f\n", celciusToFahrenheit(temp))
30     }
31 }
32
```

OUTPUT

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\
Masukan jumlah data: 3
Masukan suhu dalam celcius:
16 22 26
Suhu dalam Fahrenheit:
60.80
71.60
78.80
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>
```

DEKSRIPSI

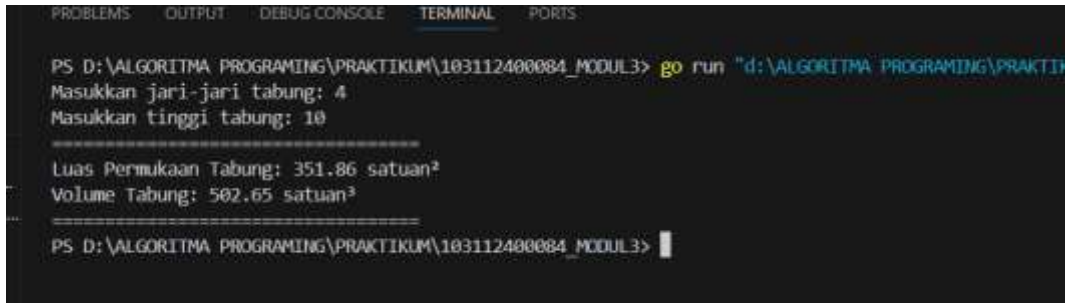
Program ini menerima input suhu dalam Celsius dari pengguna dan mengonversinya ke Fahrenheit. Setelah meminta jumlah data suhu, program akan meminta input suhu dan menyimpannya dalam array. Setiap suhu kemudian dikonversi menggunakan rumus $(\text{Celsius} * 9.0 / 5.0) + 32$ dan hasilnya ditampilkan. Jika input jumlah data tidak valid, program menampilkan pesan kesalahan.

GUIDED 3

SOURCE CODE

```
guided3 > guided3.go > ...
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 // Fungsi untuk menghitung luas permukaan tabung
9 func luasPermukaanTabung(r, t float64) float64 {
10     return 2 * math.Pi * r * (r + t)
11 }
12
13 // Fungsi untuk menghitung volume tabung
14 func volumeTabung(r, t float64) float64 {
15     return math.Pi * math.Pow(r, 2) * t
16 }
17
18 func main() {
19     var r, t float64
20
21     // Input jari-jari dan tinggi tabung dengan validasi
22     fmt.Print("Masukkan jari-jari tabung: ")
23     _, errR := fmt.Scan(&r)
24     fmt.Print("Masukkan tinggi tabung: ")
25     _, errT := fmt.Scan(&t)
26
27     // Memeriksa apakah input valid
28     if errR != nil || errT != nil {
29         fmt.Println("Input tidak valid! Harap masukkan angka yang benar.")
30         return
31     }
32
33     // Memeriksa apakah jari-jari dan tinggi bernilai positif
34     if r <= 0 || t <= 0 {
35         fmt.Println("Jari-jari dan tinggi tabung harus lebih dari nol.")
36         return
37     }
38
39     // Menghitung luas permukaan dan volume
40     luas := luasPermukaanTabung(r, t)
41     volume := volumeTabung(r, t)
42
43     // Menampilkan hasil
44     fmt.Println("=====")
45     fmt.Printf("Luas Permukaan Tabung: %.2f satuan²\n", luas)
46     fmt.Printf("Volume Tabung: %.2f satuan³\n", volume)
47     fmt.Println("=====")
48 }
49
```

OUTPUT

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active. The prompt is 'PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>'. The user enters 'go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3\main.go"'. The program prompts for 'Masukkan jari-jari tabung: 4' and 'Masukkan tinggi tabung: 10'. It then outputs 'Luas Permukaan Tabung: 351.86 satuan²' and 'Volume Tabung: 502.65 satuan³'. The prompt returns to 'PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3\main.go"
Masukkan jari-jari tabung: 4
Masukkan tinggi tabung: 10
=====
Luas Permukaan Tabung: 351.86 satuan²
Volume Tabung: 502.65 satuan³
=====
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>
```

DEKSRISI

Program ini menghitung luas permukaan dan volume tabung berdasarkan input jari-jari dan tinggi tabung dari pengguna. Jika input valid (positif), program menghitung luas permukaan dan volume menggunakan rumus yang sesuai dan menampilkan hasilnya. Jika input tidak valid, program menampilkan pesan kesalahan.

III. UNGUIDED

UNGUIDED 1

SOURCE CODE

```
1 package main
2 // MUFAIL ALAUDDIN TSAQIF
3 //103112400084
4 import "fmt"
5
6 func main() {
7     var a, b, c, d int
8
9     fmt.Scan(&a, &b, &c, &d)
10
11     p1:=permutasi(a, c)
12     c1 := kombinasi(a, c)
13     fmt.Printf("%d %d\n", p1, c1)
14
15     p2 :=permutasi(b, d)
16     c2 := kombinasi(b, d)
17     fmt.Printf("%d %d\n", p2, c2)
18 }
19
20 func faktorial(n int) int {
21     if n <= 1 {
22         return 1
23     }
24     return n * faktorial(n-1)
25 }
26
27 func permutasi(n, r int) int {
28     return faktorial(n) / faktorial(n-r)
29 }
30
31 func kombinasi(n, r int) int {
32     return faktorial(n) / (faktorial(r) * faktorial(n-r))
33 }
```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROG
5 10 3 10
60 10
3628800 1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> |
```

DEKSRIPSI

Program ini menghitung permutasi dan kombinasi dengan menggunakan fungsi faktorial. Setelah menerima empat input, program memeriksa apakah kondisi $a \geq b$ dan $c \geq d$ terpenuhi. Jika ya, hasil permutasi dan kombinasi ditampilkan; jika tidak, program menampilkan pesan "Input tidak sesuai".

UNGUIDED 2

SOURCE CODE

```
guided2 > Uunguided2.go > ...
1  package main
2  //NUFAIL ALAUDDIN TSAQIF
3  //103112400084
4  import "fmt"
5
6  func kuadrat(n int) int {
7      |   return n * n
8  }
9  func kurang(n int) int {
10     |   return n - 2
11 }
12 func tambah(n int) int {
13     |   return n + 1
14 }
15 func fogoh(n int) int {
16     |   return kuadrat(kurang(tambah(n)))
17 }
18 func gohof(n int) int {
19     |   return kurang(tambah(kuadrat(n)))
20 }
21 func hofog(n int) int {
22     |   return tambah(kuadrat(kurang(n)))
23 }
24 func main() {
25     |   var a, b, c int
26     |   fmt.Scan(&a, &b, &c)
27     |   fmt.Println(fogoh(a))
28     |   fmt.Println(gohof(b))
29     |   fmt.Println(hofog(c))
30 }
31
```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3\Uunguided2.go"
7 2 10
36
3
65
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3\Uunguided2.go"
5 5 5
16
24
10
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> |
```

DEKSRIPSI

Program ini menghitung nilai dari tiga fungsi komposisi yang melibatkan operasi kuadrat, pengurangan 2, dan penambahan 1. Tiga fungsi komposisi utama, yaitu fogoh, gohof, dan hofog, mengkombinasikan ketiga operasi ini dalam urutan yang berbeda. Program menerima tiga input (a, b, dan c), lalu menghitung dan menampilkan hasil komposisi masing-masing fungsi.

UNGUIDED 3

SOURCE CODE

```
1 package main
2 // NUFAIL ALAUDOIN TSAQIF
3 // 103112400084
4 import "fmt"
5
6 func jarak(a, b, c, d float64) float64 {
7     return ((a - c) * (a - c)) + ((b - d) * (b - d))
8 }
9
10 func didalam(cx, cy, r, x, y float64) bool {
11     return jarak(cx, cy, x, y) <= (r * r)
12 }
13
14 func main() {
15     var cx1, cy1, r1 float64
16     var cx2, cy2, r2 float64
17     var x, y float64
18     fmt.Scan(&cx1, &cy1, &r1)
19     fmt.Scan(&cx2, &cy2, &r2)
20     fmt.Scan(&x, &y)
21     dalamLingkaran1 := didalam(cx1, cy1, r1, x, y)
22     dalamLingkaran2 := didalam(cx2, cy2, r2, x, y)
23     if dalamLingkaran1 && dalamLingkaran2 {
24         fmt.Println("Titik di dalam lingkaran 1 dan 2")
25     } else if dalamLingkaran1 {
26         fmt.Println("Titik di dalam lingkaran 1")
27     } else if dalamLingkaran2 {
28         fmt.Println("Titik di dalam lingkaran 2")
29     } else {
30         fmt.Println("Titik di luar lingkaran 1 dan 2")
31     }
32 }
```

OUTPUT

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3\main.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3\main.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400084_MODUL3>
```

DEKSRIPSI

Program ini memeriksa apakah sebuah titik (x, y) berada di dalam dua lingkaran. Fungsi jarak menghitung jarak antara titik dan pusat lingkaran menggunakan rumus Euclidean, sementara fungsi didalam memeriksa apakah titik berada dalam lingkaran dengan membandingkan jarak titik dengan jari-jari lingkaran. Program meminta input untuk pusat, jari-jari kedua lingkaran, dan koordinat titik, kemudian menentukan posisi titik apakah berada di dalam kedua lingkaran, satu lingkaran, atau di luar keduanya.

IV. KESIMPULAN

Kesimpulan dari laporan praktikum ini adalah bahwa pemahaman tentang fungsi dalam bahasa Go sangat penting untuk memecahkan berbagai masalah komputasi secara terstruktur dan efisien. Fungsi memungkinkan pemrogram untuk menyusun program dengan cara yang modular, meningkatkan keterbacaan, dan memudahkan pemeliharaan kode. Praktikum ini menunjukkan penerapan fungsi dalam berbagai konteks, seperti menghitung permutasi, konversi suhu, dan perhitungan geometri. Melalui penggunaan fungsi-fungsi dasar seperti faktorial, komposisi fungsi, dan pengujian kondisi, praktikum ini memberikan pemahaman yang lebih mendalam tentang cara mendefinisikan, memanggil, dan mengembalikan nilai dari fungsi dalam pemrograman Go.

REFERENSI

MODUL 3-FUNGSI-ALGORITMA PEMOGRAMAN 2