

LAPORAN
PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 3
FUNGSI



Oleh:

NAMA: SETYO NUGROHO

NIM: 103112400024

KELAS: IF-12-01

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

I. DASAR TEORI

1. Definisi Function

Fungsi adalah sekumpulan instruksi yang memberikan atau menghasilkan nilai. Fungsi biasanya digunakan untuk memetakan input ke output tertentu. Fungsi memiliki dua karakteristik utama:

- a. Deklarasi tipe nilai yang dikembalikan: Menentukan jenis nilai yang akan dikembalikan oleh fungsi, seperti integer, real, atau string.
- b. Perintah return: Merupakan bagian dari badan fungsi yang mengembalikan hasil perhitungan atau output dari fungsi tersebut.

Fungsi digunakan dalam berbagai situasi seperti:

- a. Penugasan nilai ke variabel
- b. Bagian dari ekspresi
- c. Argumen untuk subprogram lain

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.

3. Pemanggilan Fungsi

Pemanggilan fungsi dilakukan dengan menyebutkan nama fungsi dan memberikan argumen yang sesuai. Fungsi dapat dipanggil untuk melakukan berbagai operasi seperti perhitungan, manipulasi data, dan bahkan sebagai bagian dari ekspresi lain. Fungsi dapat menghasilkan nilai yang bisa disimpan dalam variabel atau langsung ditampilkan.

II. GUIDED

```
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\103112400024_MODUL3> go run "d:\ALGORITMA
4 6
360
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3>
```

DEKSRIPSI:

Program menerima dua input integer, a dan b. Jika a lebih besar atau sama dengan b, program menghitung permutasi a dan b dengan memanggil permutasi(a, b), jika tidak, memanggil permutasi(b, a). di mana faktorial dihitung dengan fungsi faktorial, yang mengalikan angka dari 1 hingga n.

GUIDED 2

```
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

```
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112480024_MODUL3> go run "d:\ALGORITMA PROGRA
Masukan jumlah data: 3
Masukan suhu dalam celcius:
16 18 22
Suhu dalam Fahrenheit:
60.80
64.40
71.60
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112480024_MODUL3>
```

DEKSRIPI

Program meminta jumlah suhu dalam Celsius yang akan dikonversi ke Fahrenheit. Jika input tidak valid atau ≤ 0 , muncul pesan kesalahan. Pengguna memasukkan suhu sesuai jumlah yang ditentukan, disimpan dalam array. Program mengonversi setiap suhu ke Fahrenheit dengan rumus: $(C \times 9/5) + 32$ ($C \times 9/5 + 32$) menggunakan fungsi `celciusToFahrenheit`, lalu menampilkan hasilnya.

GUIDED 3

```
guided3 > -o 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




```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODULE3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODULE3\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\103112400024_MODULE3> 
```

DEKSRISI

Program meminta pengguna memasukkan jari-jari dan tinggi tabung, lalu memeriksa apakah input valid dan bernilai positif. Jika input tidak valid atau kurang dari atau sama dengan nol, program menampilkan pesan kesalahan dan berhenti. Jika valid, program menghitung luas permukaan dan volume tabung menggunakan rumus yang sesuai. Setelah perhitungan selesai, hasil luas permukaan dan volume ditampilkan dalam format yang telah ditentukan.

III. UNGUIDED

UNGUIDED 1

```
unguided1 >  unguided1.go > ...
1 //Setyo Nugroho
2 //103112400024
3 package main
4
5 import "fmt"
6
7 func main() {
8     var a, b, c, d int
9
10    fmt.Scan(&a, &b, &c, &d)
11
12    p1:=permutasi(a, c)
13    c1 := kombinasi(a, c)
14    fmt.Printf("%d %d\n", p1, c1)
15
16    p2 :=permutasi(b, d)
17    c2 := kombinasi(b, d)
18    fmt.Printf("%d %d\n", p2, c2)
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\103112400024_MODULE3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\10
5 10 3 10
60 10
3628800 1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODULE3> |
```

DEKSRIPSI

Program menghitung permutasi dan kombinasi berdasarkan input pengguna. Dimulai dengan fungsi faktorial untuk menghitung faktorial, yang digunakan dalam hitungPermutasi dan hitungKombinasi sesuai rumus matematis. Program menerima empat input: dua angka pertama untuk perhitungan pertama dan dua angka berikutnya untuk perhitungan kedua. Jika kondisi $a \geq b$ dan $c \geq d$ terpenuhi, hasil perhitungan ditampilkan. Jika tidak, program menampilkan pesan kesalahan.

UNGUIDED 2

SOURCE CODE

```
unguided2 > -go Unguided2.go > ...
1  //Setyo Nugroho
2  //103112400024
3  package main
4
5  import "fmt"
6
7  func kuadrat(n int) int {
8      |   return n * n
9  }
10 func kurang(n int) int {
11     |   return n - 2
12 }
13 func tambah(n int) int {
14     |   return n + 1
15 }
16 func fogoh(n int) int {
17     |   return kuadrat(kurang(tambah(n)))
18 }
19 func gohof(n int) int {
20     |   return kurang(tambah(kuadrat(n)))
21 }
22 func hofog(n int) int {
23     |   return tambah(kuadrat(kurang(n)))
24 }
25 func main() {
26     |   var a, b, c int
27     |   fmt.Scan(&a, &b, &c)
28     |   fmt.Println(fogoh(a))
29     |   fmt.Println(gohof(b))
30     |   fmt.Println(hofog(c))
31 }
32
```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3\Unguided2.go"
7 2 10
36
3
65
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3\Unguided2.go"
5 5 5
16
24
10
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3> |
```

DEKSRIPSI

Program ini menghitung tiga fungsi komposisi berbeda menggunakan operasi matematika dasar: kuadrat, pengurangan 2, dan penambahan 1. Fungsi utama yang didefinisikan adalah fogoh, gohof, dan hofog, yang mengombinasikan ketiga operasi dalam urutan berbeda. Dalam fungsi main, program menerima tiga input (a, b, c), menghitung hasil komposisi untuk fogoh(a), gohof(b), dan hofog(c), lalu menampilkannya sebagai output.

UNGUIDED 3

```
unguided3 > go unguided3.go > main
1 //Setyo Nugroho
2 //103112400024
3 package main
4
5 import "fmt"
6
7 func jarak(a, b, c, d float64) float64 {
8     return ((a - c) * (a - c)) + ((b - d) * (b - d))
9 }
10 func didalam(cx, cy, r, x, y float64) bool {
11     return jarak(cx, cy, x, y) <= (r * r)
12 }
13 func main() {
14     var cx1, cy1, r1 float64
15     var cx2, cy2, r2 float64
16     var x, y float64
17     fmt.Scan(&cx1, &cy1, &r1)
18     fmt.Scan(&cx2, &cy2, &r2)
19     fmt.Scan(&x, &y)
20     dalamLingkaran1 := didalam(cx1, cy1, r1, x, y)
21     dalamLingkaran2 := didalam(cx2, cy2, r2, x, y)
22     if dalamLingkaran1 && dalamLingkaran2 {
23         fmt.Println("Titik di dalam lingkaran 1 dan 2")
24     } else if dalamLingkaran1 {
25         fmt.Println("Titik di dalam lingkaran 1")
26     } else if dalamLingkaran2 {
27         fmt.Println("Titik di dalam lingkaran 2")
28     } else {
29         fmt.Println("Titik di luar lingkaran 1 dan 2")
30     }
31 }
```

OUTPUT

```
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3\unguided3.go"
1 1 5
8 8 4
2 2
Titik di dalam lingkaran 1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3\unguided3.go"
1 2 3
4 5 6
7 8
Titik di dalam lingkaran 2
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL3>
```

DEKSRIPSI

Program memeriksa apakah sebuah titik (x, y) berada di dalam dua lingkaran berbeda. Fungsi jarak menghitung jarak Euclidean antara dua titik, sedangkan fungsi didalam menentukan apakah titik tersebut berada dalam lingkaran berdasarkan perbandingan jarak ke pusat dengan jari-jari lingkaran. Dalam main, program menerima input pusat dan jari-jari kedua lingkaran serta koordinat titik yang diuji. Hasilnya menunjukkan apakah titik berada di dalam kedua lingkaran, hanya satu lingkaran, atau di luar keduanya.

IV. KESIMPULAN

Program yang dibuat dalam modul ini menunjukkan bagaimana fungsi dapat digunakan untuk menyelesaikan berbagai perhitungan matematis, seperti faktorial, permutasi, kombinasi, komposisi fungsi, dan analisis posisi titik terhadap lingkaran. Dengan menggunakan pendekatan modular melalui fungsi, program menjadi lebih terstruktur, efisien, dan mudah dipahami.

Penggunaan fungsi faktorial, permutasi, dan kombinasi memungkinkan perhitungan yang lebih sistematis dalam menentukan nilai kombinatorial. Sementara itu, fungsi komposisi seperti fogoh, gohof, dan hofog menunjukkan bagaimana beberapa fungsi dapat dikombinasikan untuk membentuk operasi baru. Selain itu, penerapan konsep jarak Euclidean dalam fungsi jarak dan didalam membantu menentukan posisi titik relatif terhadap dua lingkaran.

Secara keseluruhan, modul ini menegaskan pentingnya fungsi dalam pemrograman untuk meningkatkan keterbacaan kode, memudahkan debugging, dan memungkinkan penggunaan ulang kode dalam berbagai kasus perhitungan.

REFERENSI

MODUL 3-FUNGSI-ALGORITMA PEMOGRAMAN 2