

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

MODUL 3

FUNGSI



**Telkom
University**

Oleh:

Muhammad Faris Rachmadi

103112400079

IF12-01

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

I. DASAR TEORI

3.1 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

3.2 Deklarasi Function

Deklarasi fungsi sama dengan prosedur, yaitu berada pada blok yang terpisah dengan program utama. Pada bagian deklarasi terlihat setelah parameter terdapat **tipe data dari nilai** yang dikembalikan, sedangkan pada bagian badan fungsi terdapat **return dari nilai** yang dikembalikan.

3.3 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

Code:

```
103112400079_guided1 > go guided1.go > permutasi
1  package main
2
3  import "fmt"
4
5  func main() {
6      var a, b int
7      fmt.Scan(&a, &b)
8
9      if a >= b {
10         fmt.Println(permutasi(a, b))
11     } else {
12         fmt.Println(permutasi(b, a))
13     }
14 }
15
16
17 func faktorial(n int) int {
18     hasil := 1
19     for i := 1; i <= n; i++ {
20         hasil *= i
21     }
22     return hasil
23 }
24
25 func permutasi(n, r int) int {
26     if r > n {
27         return 0
28     }
29     return faktorial(n) / faktorial(n-r)
30 }
31
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3\103112400079_guided1\guided1.go"
7 4
840
```

Deskripsi:

Program Go ini bertujuan untuk menghitung nilai permutasi dari dua bilangan yang diberikan oleh pengguna. Permutasi dihitung menggunakan rumus:

$$P(n, r) = \frac{n!}{(n - r)!}$$

Guided 2

Code:

```
10311240079_guided2 > go guided2.go & main
1  package main
2
3  import (
4      "fmt"
5  )
6
7  // Fungsi untuk mengonversi suhu dari Celsius ke Fahrenheit
8  func celsiusToFahrenheit(celsius float64) float64 {
9      return (9.0/5.0)*celsius + 32
10 }
11
12 func main() {
13     var N int
14     fmt.Print("Masukkan jumlah data: ")
15     _, err := fmt.Scan(&N)
16     if err != nil || N <= 0 {
17         fmt.Println("Input tidak valid, pastikan memasukkan angka positif.")
18         return
19     }
20
21     temperatures := make([]float64, N)
22
23     // Membaca suhu dalam Celsius
24     fmt.Println("Masukkan suhu dalam Celsius:")
25     for i := 0; i < N; i++ {
26         _, err := fmt.Scan(&temperatures[i])
27         if err != nil {
28             fmt.Println("Input tidak valid, pastikan memasukkan angka.")
29             return
30         }
31     }
32
33     // Mengonversi ke Fahrenheit dan mencetak hasil
34     fmt.Println("Suhu dalam Fahrenheit:")
35     for _, temp := range temperatures {
36         fmt.Printf("%.2f\n", celsiusToFahrenheit(temp))
37     }
38 }
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3\10311240079_guided2\guided2.go"
Masukkan jumlah data: 3
Masukkan suhu dalam Celsius:
36 37 40
Suhu dalam Fahrenheit:
96.80
98.60
104.00
```

Deskripsi:

Program Go ini meminta pengguna untuk memasukkan jumlah data suhu dalam Celsius, kemudian mengonversi setiap suhu yang dimasukkan ke dalam Fahrenheit dan menampilkan hasilnya. Pertama, program meminta jumlah data suhu yang akan dimasukkan. Setelah itu, pengguna diminta untuk memasukkan suhu dalam Celsius satu per satu. Program akan mengonversi suhu tersebut menggunakan rumus konversi Celsius ke Fahrenheit dan menampilkan hasil konversinya dengan format dua angka desimal. Jika input yang diberikan tidak valid, program akan memberikan pesan kesalahan yang sesuai.

Guided 3

Code:

```
103112400079_guided3 > go guided3.go > ...
1 package main
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.Println("Masukkan jari-jari tabung: ")
23     _, errR := fmt.Scan(&r)
24     fmt.Println("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("=====")
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3> go run "c:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3\
Masukkan jari-jari tabung: 12
Masukkan tinggi tabung: 7
=====
Luas Permukaan Tabung: 1432.57 satuan²
Volume Tabung: 3166.73 satuan³
=====
```

Deskripsi:

Program Go ini menghitung luas permukaan dan volume tabung berdasarkan input jari-jari dan tinggi yang dimasukkan oleh pengguna. Program pertama-tama meminta pengguna untuk memasukkan nilai jari-jari dan tinggi tabung, kemudian memeriksa apakah input valid dan apakah kedua nilai tersebut lebih besar dari nol. Jika input valid, program akan menghitung luas permukaan dan volume tabung menggunakan rumus matematika yang sesuai. Hasil perhitungan luas permukaan dan volume akan ditampilkan dengan format dua angka desimal. Jika ada kesalahan input atau nilai yang tidak valid, program akan memberikan pesan kesalahan yang sesuai.

III. UNGUIDED

Unguided 1

Code:

```
10311240079_unguided1 > go run unguided1.go > ...
1  package main
2
3  // Muhammad Faris Rachmadi
4  // 10311240079
5
6  import "fmt"
7
8  func fact(n int) int {
9      if n == 0 {
10         return 1
11     }
12     r := 1
13     for i := 1; i <= n; i++ {
14         r *= i
15     }
16     return r
17 }
18
19 func permu(n, r int) int {
20     return fact(n) / fact(n-r)
21 }
22
23 func combi(n, r int) int {
24     return fact(n) / (fact(r) * fact(n-r))
25 }
26
27 func main() {
28     var a, b, c, d int
29     fmt.Print("a b c d: ")
30     fmt.Scan(&a, &b, &c, &d)
31
32     if a < c || b < d {
33         fmt.Println("Syarat salah")
34         return
35     }
36
37     fmt.Println(permu(a, c), combi(a, c))
38     fmt.Print(permu(b, d), combi(b, d))
39 }
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3> go run "c:\Users\Faris\Documents
a b c d: 5 10 3 10
60 10
3628800 1
```

Deskripsi:

Program Go ini menghitung permutasi dan kombinasi berdasarkan input yang diberikan oleh pengguna. Pengguna diminta untuk memasukkan empat bilangan bulat, yaitu a, b, c, dan d. Program kemudian memeriksa apakah syarat $a \geq c$ dan $b \geq d$ terpenuhi; jika tidak, program akan menampilkan pesan kesalahan "Syarat salah" dan berhenti. Jika syarat valid, program akan menghitung dan menampilkan nilai permutasi dan kombinasi untuk pasangan (a, c) dan (b, d) menggunakan rumus matematika permutasi dan kombinasi. Nilai permutasi dihitung dengan membagi faktorial a dengan faktorial (a-c), sementara nilai kombinasi dihitung dengan membagi faktorial a dengan hasil perkalian faktorial c dan (a-c).

Unguided 2

Code:

```
103112400079_unguided2 > go unguided2.go > ...
1  package main
2
3  // Muhammad Faris Rachmadi
4  // 103112400079
5
6  import "fmt"
7
8  func f(n int) int {
9      return n * n
10 }
11
12 func g(n int) int {
13     return n - 2
14 }
15
16 func h(n int) int {
17     return n + 1
18 }
19
20 func fogoh(n int) int {
21     return f(g(h(n)))
22 }
23
24 func gohof(n int) int {
25     return g(h(f(n)))
26 }
27
28 func hofog(n int) int {
29     return h(f(g(n)))
30 }
31
32 func main() {
33     var a, b, c int
34     fmt.Scan(&a, &b, &c)
35
36     fmt.Println(fogoh(a))
37     fmt.Println(gohof(b))
38     fmt.Println(hofog(c))
39 }
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3> go run "c:\Users\Faris\Documents\
7 2 10
36
3
65
```

Deskripsi:

Program Go ini mendefinisikan beberapa fungsi matematika dan komposisinya untuk dihitung berdasarkan input yang diberikan oleh pengguna. Fungsi-fungsi yang ada adalah $f(n)$ yang mengembalikan kuadrat dari n , $g(n)$ yang mengurangi n dengan 2, dan $h(n)$ yang menambahkan 1 pada n . Program kemudian mendefinisikan tiga komposisi fungsi: $\text{fogoh}(n)$ yang mengkomposisikan fungsi f , g , dan h secara berurutan, $\text{gohof}(n)$ yang mengkomposisikan g , h , dan f , serta $\text{hofog}(n)$ yang mengkomposisikan h , f , dan g . Pengguna diminta untuk memasukkan tiga bilangan, dan program akan menghitung dan menampilkan hasil dari masing-masing komposisi fungsi untuk bilangan yang dimasukkan.

Unguided 3

Code:

```
103112400079_unguided3 > -o unguided3.go > ...
1  package main
2
3  // Muhammad Faris Rachmadi
4  // 103112400079
5
6  import "fmt"
7
8  func DalamLingkaran(cx, cy, r, x, y float64) bool {
9      jarakKuadrat := (x-cx)*(x-cx) + (y-cy)*(y-cy)
10     return jarakKuadrat <= r*r
11 }
12
13 func main() {
14     var cx1, cy1, r1 float64
15     var cx2, cy2, r2 float64
16     var x, y float64
17
18     fmt.Scan(&cx1, &cy1, &r1)
19     fmt.Scan(&cx2, &cy2, &r2)
20     fmt.Scan(&x, &y)
21
22     diDalamLingkaran1 := DalamLingkaran(cx1, cy1, r1, x, y)
23     diDalamLingkaran2 := DalamLingkaran(cx2, cy2, r2, x, y)
24
25     if diDalamLingkaran1 && diDalamLingkaran2 {
26         fmt.Println("Titik di dalam lingkaran 1 dan 2")
27     } else if diDalamLingkaran1 {
28         fmt.Println("Titik di dalam lingkaran 1")
29     } else if diDalamLingkaran2 {
30         fmt.Println("Titik di dalam lingkaran 2")
31     } else {
32         fmt.Println("Titik di luar lingkaran 1 dan 2")
33     }
34 }
```

Output:

```
PS C:\Users\Faris\Documents\ALGORITMA PEMROGRAMAN 2\Modul 3> go run "c:\Users\Faris\Documents\
1 1 5
8 8 4
15 20
Titik di luar lingkaran 1 dan 2
```

Deskripsi:

Program Go ini mengecek apakah sebuah titik berada di dalam salah satu dari dua lingkaran yang diberikan. Program pertama-tama meminta pengguna untuk memasukkan koordinat pusat dan jari-jari dari dua lingkaran, serta koordinat titik yang akan diperiksa. Fungsi DalamLingkaran digunakan untuk menghitung jarak kuadrat antara titik dan pusat lingkaran, kemudian membandingkannya dengan kuadrat jari-jari lingkaran untuk menentukan apakah titik tersebut berada di dalam lingkaran. Program kemudian memeriksa apakah titik tersebut berada di dalam salah satu atau kedua lingkaran, dan menampilkan hasilnya, yaitu apakah titik berada di dalam lingkaran 1, lingkaran 2, atau keduanya, atau di luar kedua lingkaran.

IV. KESIMPULAN

Melalui latihan dengan berbagai program, peserta dapat menguasai konsep dasar penggunaan fungsi dalam pemrograman, khususnya dalam bahasa Go. Praktikum ini mencakup implementasi berbagai fungsi yang berguna untuk menyelesaikan masalah matematis dan logika, seperti perhitungan permutasi, konversi suhu, volume dan luas permukaan tabung, serta komposisi fungsi. Selain itu, praktikum ini juga mengajarkan bagaimana cara memvalidasi input dan menangani kesalahan, serta pentingnya memahami alur pemrograman yang efisien melalui penggunaan fungsi. Secara keseluruhan, praktikum ini membantu peserta untuk meningkatkan keterampilan dalam menyusun dan memahami fungsi dalam konteks pemrograman yang lebih luas.

V. REFERENSI

MODUL 3 PRAKTIKUM ALGORITMA PEMROGRAMAN 2 - FUNGSI