

**LAPORAN**  
**PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 2**  
**MODUL 4**  
**PROSEDUR**



Oleh:

NAMA: MOHAMMAD REYHAN ARETHA FATIN

NIM: 103112400078

KELAS: IF-12-01

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## **I. DASAR TEORI**

### **1. Definisi Procedure**

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi baru yang dibuat untuk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar. Prosedur akan menghasilkan suatu akibat atau efek langsung pada program ketika dipanggil pada program utama. Suatu subprogram dikatakan prosedur apabila:

- a. Tidak ada deklarasi tipe nilai yang dikembalikan, dan
- b. Tidak terdapat kata kunci return dalam badan subprogram.

Kedudukannya prosedur sama seperti instruksi dasar yang sudah ada sebelumnya (assignment) dan/atau instruksi yang berasal dari paket (fmt), seperti fmt.Scan dan fmt.Print. Karena itu selalu pilih nama prosedur yang berbentuk kata kerja atau sesuatu yang merepresentasikan proses sebagai nama dari prosedur. Contoh: cetak, hitungRerata, cariNilai, belok, mulai, ...

### **2. Deklarasi dan Pemanggilan Prosedur**

Dalam pemrograman terstruktur, mendeklarasikan dan memanggil prosedur adalah praktik dasar. Prosedur dideklarasikan dengan parameter tertentu yang mendefinisikan nilai input yang dibutuhkan untuk eksekusi. Sebagai contoh, dalam kode Go yang disediakan, prosedur cetakNFibo menghasilkan n nilai pertama dalam deret Fibonacci, dan dapat dipanggil baik dengan variabel atau dengan nilai numerik langsung.

### **3. Parameter dalam Prosedur**

Parameter adalah data yang dikirim dari program utama ke prosedur. Ada dua jenis parameter:

- a. Parameter Formal: Ditulis saat mendeklarasikan prosedur.
- b. Parameter Aktual: Nilai yang diberikan saat memberikan prosedur.

Berdasarkan cara memori dialokasikan, ada dua tipe parameter:

- a. Pass by Value: Hanya menyalin nilai, lalu perubahan di dalam prosedur tidak memengaruhi nilai aslinya.
- b. Pass by Reference: Menggunakan alamat memori, perubahan yang ada didalam prosedur memengaruhi nilai aslinya.

## II. GUIDED

### GUIDED 1

SOURCE CODE:

```
guided1 > go guided1.go > ...
1 package main
2 |
3 import "fmt"
4
5 func hitungGaji(nama string, gajiPokok float64, jamLembur int) {
6     bonusLembur := float64(jamLembur) * 50000
7     totalGaji := gajiPokok + bonusLembur
8
9     fmt.Println("\n=== Slip Gaji ===")
10    fmt.Println("Nama Karyawan:", nama)
11    fmt.Printf("Gaji Pokok : Rp%.2f\n", gajiPokok)
12    fmt.Printf("Bonus Lembur: Rp%.2f (%d jam x Rp50,000)\n", bonusLembur, jamLembur)
13    fmt.Printf("Total Gaji : Rp%.2f\n", totalGaji)
14 }
15
16 func main() {
17     var nama string
18     var gajiPokok float64
19     var jamLembur int
20
21     fmt.Print("Masukkan Nama Karyawan: ")
22     fmt.Scanln(&nama)
23     fmt.Print("Masukkan Gaji Pokok: ")
24     fmt.Scanln(&gajiPokok)
25     fmt.Print("Masukkan Jam Lembur: ")
26     fmt.Scanln(&jamLembur)
27
28     hitungGaji(nama, gajiPokok, jamLembur)
29 }
30
```

### OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MCDUL4> go run "d:\ALGORITMA PROGRAMING
Masukkan Nama Karyawan: Aretha
Masukkan Gaji Pokok: 8000000
Masukkan Jam Lembur: 3

=== Slip Gaji ===
Nama Karyawan: Aretha
Gaji Pokok : Rp8000000.00
Bonus Lembur: Rp150000.00 (3 jam x Rp50,000)
Total Gaji : Rp8150000.00
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MCDUL4> |
```

**DEKSRIPSI:**

Program yang digunakan untuk menghitung gaji karyawan. Program ini meminta input dari pengguna berupa nama karyawan, gaji pokok, dan jumlah jam lembur yang dilakukan oleh karyawan tersebut. Dengan menggunakan informasi jam lembur, program akan menghitung bonus lembur berdasarkan tarif Rp50.000 per jam lembur. Kemudian, program akan menampilkan slip gaji yang mencakup gaji pokok, bonus lembur, dan total gaji yang diterima oleh karyawan setelah bonus lembur dihitung. Output hasil perhitungan gaji ditampilkan dengan format yang jelas dan terstruktur.

## GUIDED 2

### SOURCE CODE:

```
guided2 > -go guided2.go > ...
1  package main
2
3  import "fmt"
4
5  // Prosedur untuk menghitung rata-rata dan menentukan kelulusan
6  func hitungRataRata(nama string, nilai1, nilai2, nilai3 float64) {
7      ratarata := (nilai1 + nilai2 + nilai3) / 3
8      status := "Tidak Lulus"
9      if ratarata >= 60 {
10         status = "Lulus"
11     }
12
13     fmt.Println("\n=== Hasil Akademik ===")
14     fmt.Println("Nama Mahasiswa :", nama)
15     fmt.Printf("Nilai 1 : %.2f\n", nilai1)
16     fmt.Printf("Nilai 2 : %.2f\n", nilai2)
17     fmt.Printf("Nilai 3 : %.2f\n", nilai3)
18     fmt.Printf("Rata-rata : %.2f\n", ratarata)
19     fmt.Println("Status:", status)
20 }
21 func main() {
22     var nama string
23     var nilai1, nilai2, nilai3 float64
24
25     // Input dari pengguna
26     fmt.Print("Masukkan Nama Mahasiswa: ")
27     fmt.Scanln(&nama)
28
29     fmt.Print("Masukkan Nilai 1: ")
30     fmt.Scanln(&nilai1)
31
32     fmt.Print("Masukkan Nilai 2: ")
33     fmt.Scanln(&nilai2)
34
35     fmt.Print("Masukkan Nilai 3: ")
36     fmt.Scanln(&nilai3)
37
38     hitungRataRata(nama, nilai1, nilai2, nilai3)
39
40 }
41
```

## OUTPUT:

```
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL4> go run "d:\ALGORITMA PROGRAMING\PM
Masukkan Nama Mahasiswa: Aretha
Masukkan Nilai 1: 89
Masukkan Nilai 2: 88
Masukkan Nilai 3: 85

=== Hasil Akademik ===
Nama Mahasiswa : Aretha
Nilai 1 : 89.00
Nilai 2 : 88.00
Nilai 3 : 85.00
Rata-rata : 87.33
Status: Lulus
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL4> |
```

## DEKSRIPSI:

Program ini digunakan untuk menghitung rata-rata nilai mahasiswa dan menentukan kelulusan berdasarkan nilai tersebut. Program meminta input dari pengguna berupa nama mahasiswa dan tiga nilai ujian. Setelah nilai dimasukkan, program menghitung rata-rata dari ketiga nilai tersebut. Jika rata-rata lebih dari atau sama dengan 60, status mahasiswa akan ditampilkan sebagai "Lulus", sebaliknya statusnya "Tidak Lulus". Program kemudian menampilkan hasil berupa nama mahasiswa, nilai masing-masing ujian, rata-rata nilai, dan status kelulusan.

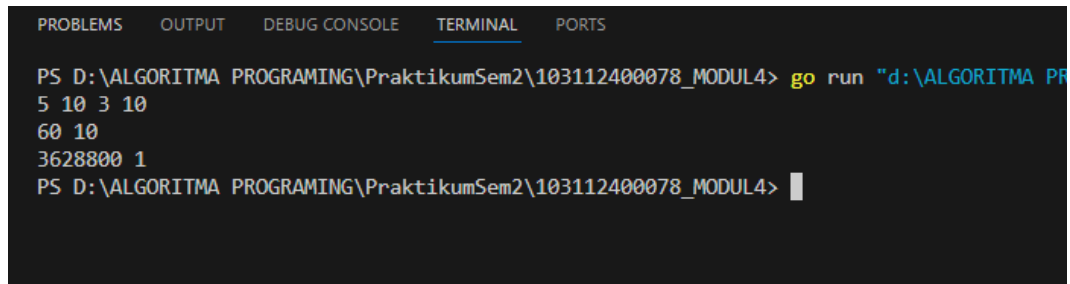
### III. UNGUIDED

#### UNGUIDED 1

#### SOURCE CODE

```
unguided1 > go 1.go > faktorial
1  package main
2  // Mohammad Reyhan Aretha Fatin
3  // 103112400078
4  import "fmt"
5
6  func main() {
7      var a, b, c, d int
8      var permut, komb int
9      fmt.Scan(&a, &b, &c, &d)
10     if a >= c {
11         permutasi(a, c, &permut)
12         kombinasi(a, c, &komb)
13         fmt.Println(permut, komb)
14     } else {
15         permutasi(c, a, &permut)
16         kombinasi(c, a, &komb)
17         fmt.Println(permut, komb)
18     }
19     if b >= d {
20         permutasi(b, d, &permut)
21         kombinasi(b, d, &komb)
22         fmt.Println(permut, komb)
23     } else {
24         permutasi(d, b, &permut)
25         kombinasi(d, b, &komb)
26         fmt.Println(permut, komb)
27     }
28 }
29
30 func faktorial(n int, hasil *int) {
31     *hasil = 1
32     for i := 1; i <= n; i++ {
33         *hasil *= i
34     }
35 }
36
37 func permutasi(n, r int, hasil *int) {
38     var test1, test2 int
39     if r > n {
40         *hasil = 0
41     }
42     faktorial(n, &test1)
43     faktorial(n-r, &test2)
44     *hasil = test1 / test2
45 }
46
47 func kombinasi(n, r int, hasil *int) {
48     var test1, test2, test3 int
49     if r > n {
50         *hasil = 0
51     }
52     faktorial(n, &test1)
53     faktorial(n-r, &test2)
54     faktorial(r, &test3)
55     *hasil = test1 / (test2 * test3)
56 }
```

## OUTPUT

A screenshot of a terminal window with a dark background. At the top, there are tabs labeled 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is selected and underlined), and 'PORTS'. The terminal shows the following text: 'PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078\_MODUL4> go run "d:\ALGORITMA PR', '5 10 3 10', '60 10', '3628800 1', and 'PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078\_MODUL4>'.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL4> go run "d:\ALGORITMA PR
5 10 3 10
60 10
3628800 1
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL4>
```

## DEKSRIPSI

Program ini menghitung permutasi dan kombinasi dari beberapa angka menggunakan fungsi faktorial. Di dalam program, terdapat tiga fungsi utama: faktorial, permutasi, dan kombinasi. Fungsi faktorial digunakan untuk menghitung hasil faktorial dari suatu angka, sementara fungsi permutasi menghitung jumlah permutasi dari dua angka, dan fungsi kombinasi menghitung jumlah kombinasi. Program ini menerima input berupa dua angka, kemudian melakukan perbandingan untuk menentukan apakah input pertama lebih besar atau sama dengan input kedua, atau lebih kecil. Berdasarkan kondisi tersebut, program kemudian menampilkan hasil perhitungan permutasi dan kombinasi yang sesuai.



## UNGUIDED 2

## SOURCE CODE

```
guided1.go guided2.go 1.go 2.go X 3.go 10311240078_MODUL4.docx
unguided2 > 2.go > _
1 package main
2 //Mohammad Reyhan Aretha Fatin
3 //10311240078
4 import "fmt"
5
6 func hitungSkor(waktuSoal int, totalWaktu *int, totalSoal *int) {
7     if waktuSoal <= 300 {
8         *totalSoal += 1
9         *totalWaktu += waktuSoal
10    }
11 }
12
13 func tentukanPemenang(totalSoal1, totalWaktu1, totalSoal2, totalWaktu2 int, nama1, nama2 string) (string, int, int) {
14     if totalSoal1 < totalSoal2 {
15         return nama2, totalSoal2, totalWaktu2
16     } else if totalSoal1 > totalSoal2 {
17         return nama1, totalSoal1, totalWaktu1
18     } else {
19         if totalWaktu1 > totalWaktu2 {
20             return nama1, totalSoal1, totalWaktu1
21         }
22         return nama2, totalSoal2, totalWaktu2
23     }
24 }
25
26 func main() {
27     var nama1, nama2 string
28     var waktuSoal1, waktuSoal2 int
29     var totalWaktu1, totalWaktu2, totalSoal1, totalSoal2 int
30     var totalSoalSementara, totalWaktuSementara int
31     var namaSementara string
32
33     fmt.Scan(&nama1)
34     if nama1 != "Selesai" {
35
36         for i := 0; i < 8; i++ {
37             fmt.Scan(&waktuSoal1)
38             hitungSkor(waktuSoal1, &totalWaktu1, &totalSoal1)
39         }
40
41         for {
42             fmt.Scan(&nama2)
43             if nama2 == "Selesai" {
44                 break
45             }
46
47             for i := 0; i < 8; i++ {
48                 fmt.Scan(&waktuSoal2)
49                 hitungSkor(waktuSoal2, &totalWaktu2, &totalSoal2)
50             }
51
52             namaSementara, totalSoalSementara, totalWaktuSementara = tentukanPemenang(totalSoal1, totalWaktu1, totalSoal2, totalWaktu2, nama1, nama2)
53         }
54
55         fmt.Println(namaSementara, totalSoalSementara, totalWaktuSementara)
56     }
57 }
58 }
```

## OUTPUT

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\10311240078_MODUL4> go run "d:\ALGORITMA PROG
Astuti 20 50 301 301 61 71 75 10
Bertha 25 47 301 26 50 60 65 21
Selesai
Bertha 7 294
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\10311240078_MODUL4> |
```

## **DEKSRIPSI**

Program yang digunakan untuk menghitung skor dan menentukan pemenang antara dua peserta dalam sebuah perlombaan berdasarkan waktu dan soal yang diberikan. Program ini dimulai dengan mendeklarasikan beberapa variabel untuk menyimpan data nama peserta, waktu, dan soal yang dihadapi. Fungsi `hitungSkor` digunakan untuk menghitung skor berdasarkan waktu yang ditempuh oleh peserta, dan fungsi `tentukanPemenang` untuk menentukan siapa yang memenangkan perlombaan berdasarkan total skor dan waktu yang dicapai oleh masing-masing peserta. Dalam fungsi utama (`main`), program melakukan pemindaian input secara berulang untuk setiap peserta, kemudian menghitung skor dan menentukan pemenangnya. Program berhenti jika input nama peserta adalah "Selesai."

### UNGUIDED 3

#### SOURCE CODE

```
unguided3 > go 3.go > ...
1  package main
2  // Mohammad Reyhan Aretha Fatin
3  //103112400078
4  import "fmt"
5
6  func cetakDeret(n int) {
7      for n != 1 {
8          fmt.Print(n, " ")
9          if n % 2 == 0 {
10             n = n / 2
11         } else {
12             n = 3 * n + 1
13         }
14     }
15     fmt.Print(n)
16 }
17
18 func main() {
19     var n int
20     fmt.Scan(&n)
21     if n < 1000000 {
22         cetakDeret(n)
23     } else {
24         fmt.Println("Error")
25     }
26 }
27
```

#### OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL4> go run "d:\ALGORITMA PRO
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\103112400078_MODUL4> |
```

## **DEKSRIPSI**

Program yang menghasilkan deret angka berdasarkan aturan tertentu yang dikenal dengan Collatz Conjecture. Program menerima input angka  $n$  dan akan mencetak deret angka sesuai dengan aturan berikut: jika angka  $n$  genap, maka angka tersebut dibagi 2, jika ganjil, maka angka tersebut dikalikan 3 dan ditambah 1. Proses ini diulang sampai angka  $n$  mencapai 1. Fungsi cetakDeret digunakan untuk menghitung dan mencetak deret angka tersebut. Fungsi main memvalidasi bahwa input yang diberikan kurang dari 1 juta, dan jika valid, akan menjalankan fungsi cetakDeret. Jika input lebih besar atau sama dengan 1 juta, program akan menampilkan pesan "Error".

## **IV. KESIMPULAN**

Kesimpulan dari modul ini yang membahas tentang prosedur dalam pemrograman adalah bahwa prosedur merupakan konsep penting dalam pengembangan perangkat lunak yang memungkinkan pemrogram untuk menyusun kode dengan cara yang lebih modular dan terstruktur. Prosedur adalah sekumpulan instruksi atau langkah-langkah yang dirancang untuk melakukan tugas tertentu. Dalam konteks pemrograman, prosedur dapat dipanggil berkali-kali untuk menyelesaikan suatu masalah atau mencapai tujuan yang diinginkan tanpa perlu menulis ulang kode yang sama. Ini membuat pemrograman lebih efisien, karena kode yang telah diprogram dalam bentuk prosedur dapat digunakan kembali di berbagai bagian program atau bahkan dalam program lain, sehingga mengurangi redundansi.

Salah satu manfaat utama prosedur adalah memudahkan pemeliharaan dan pengembangan kode. Dengan membagi kode menjadi bagian-bagian kecil yang independen dan terpisah, program menjadi lebih mudah dikelola, diperbaiki, atau diperbarui tanpa mempengaruhi bagian lain dari program. Prosedur juga meningkatkan keterbacaan kode, karena setiap prosedur memiliki tujuan dan fungsinya sendiri yang jelas. Hal ini memungkinkan programmer lain yang bekerja dalam proyek yang sama untuk lebih cepat memahami alur program dan memperbaiki atau mengembangkan kode yang ada.

Dalam prosedur, kita dapat menentukan parameter sebagai input yang memungkinkan prosedur untuk menerima data eksternal dan melakukan pengolahan berdasarkan data tersebut. Selain itu, prosedur juga dapat mengembalikan hasil atau output setelah menjalankan instruksi yang ditentukan. Hal ini memberikan fleksibilitas dalam penggunaan prosedur, karena prosedur tidak hanya dapat digunakan untuk melaksanakan tugas, tetapi juga memberikan informasi atau hasil kepada bagian lain dari program yang memerlukannya.

Secara keseluruhan, penggunaan prosedur dalam pemrograman tidak hanya meningkatkan efisiensi dan pengorganisasian kode, tetapi juga memberikan fondasi bagi pengembangan perangkat lunak yang lebih terstruktur dan dapat dikelola dengan baik. Dengan memanfaatkan prosedur, programmer dapat menyelesaikan masalah dengan lebih sistematis dan meningkatkan kualitas perangkat lunak secara keseluruhan.

## **REFERENSI**

MODUL 4. PROSEDUR ALGORITMA PEMOGRAMAN 2