

LAPORAN
PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 4
PROSEDUR



Oleh:

NAMA: SETYO NUGROHO

NIM: 103112400024

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.

2. Deklarasi dan Pemanggilan Prosedur

Penulisan deklarasi ini berada di luar blok yang dari program utama atau func main() pada suatu program Go, dan bisa ditulis sebelum atau setelah dari blok program utama tersebut.

Pemanggilan suatu prosedur cukup mudah, yaitu dengan hanya menuliskan nama beserta parameter atau argumen yang diminta dari suatu prosedur. Sebagai contoh prosedur cetakNFibo di atas dipanggil dengan menuliskan namanya, kemudian sebuah variabel atau nilai integer tertentu sebagai argument untuk paramter n.

3. Parameter dalam Prosedur

Prosedur dapat menerima parameter yang berfungsi sebagai data input yang diperlukan untuk pemrosesan. Parameter ini dapat dikategorikan menjadi dua jenis:

- a. Parameter formal: Parameter yang dideklarasikan dalam prosedur dan berfungsi sebagai tempat untuk nilai yang akan diberikan selama pemanggilan.
- b. Parameter aktual: Nilai yang diberikan saat memanggil prosedur.

4. Pass by Value

Hanya menyalin nilai, lalu perubahan di dalam prosedur tidak memengaruhi nilai aslinya.

5. Pass by Reference

Menggunakan alamat memori, perubahan yang ada didalam prosedur memengaruhi nilai aslinya.

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\PRAKTIKUM\103112400024_MODUL4> go run "d:\ALGORITMA PROGRAMING\
Masukkan Nama Karyawan: Setyo
Masukkan Gaji Pokok: 9000000
Masukkan Jam Lembur: 4

=== Slip Gaji ===
Nama Karyawan: Setyo
Gaji Pokok : Rp9000000.00
Bonus Lembur: Rp200000.00 (4 jam x Rp50,000)
Total Gaji : Rp9200000.00
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4> |
```

DEKSRIPSI:

Program ini digunakan untuk menghitung gaji karyawan dengan meminta input berupa nama karyawan, gaji pokok, dan jumlah jam lembur. Bonus lembur dihitung berdasarkan tarif Rp50.000 per jam, dan hasilnya ditambahkan pada gaji pokok. Program kemudian menampilkan slip gaji yang mencakup gaji pokok, bonus lembur, dan total gaji 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\PRAKTIKUM\103112400024_MODUL4> go run "d:\ALGORITMA PROGRAMING\
Masukkan Nama Mahasiswa: Setyo
Masukkan Nilai 1: 80
Masukkan Nilai 2: 75
Masukkan Nilai 3: 88

=== Hasil Akademik ===
Nama Mahasiswa : Setyo
Nilai 1 : 80.00
Nilai 2 : 75.00
Nilai 3 : 88.00
Rata-rata : 81.00
Status: Lulus
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4> |
```

DEKSRIPSI:

Program ini menghitung rata-rata nilai mahasiswa dari tiga ujian dan menentukan kelulusan berdasarkan rata-rata tersebut. Program meminta input nama mahasiswa dan tiga nilai ujian, kemudian menghitung rata-rata nilai. Jika rata-rata ≥ 60 , statusnya "Lulus"; jika kurang dari 60, statusnya "Tidak Lulus". Hasilnya, termasuk nama mahasiswa, nilai ujian, rata-rata, dan status kelulusan, akan ditampilkan.

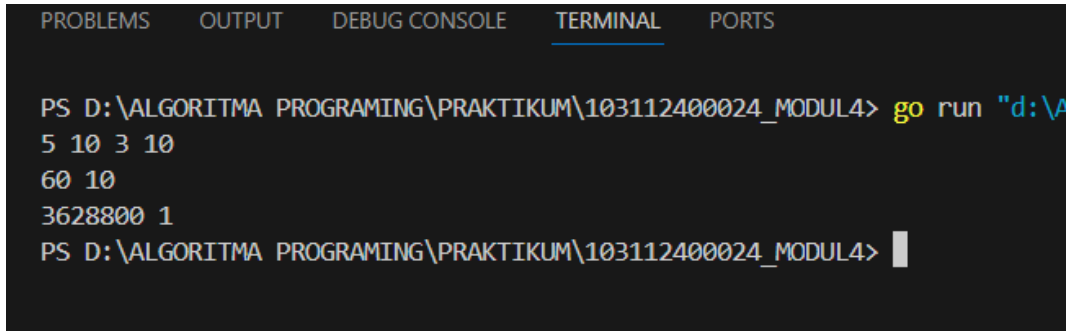
II. UNGUIDED

UNGUIDED 1

SOURCE CODE

```
guided1 > -go unguided1.go >...
1  package main
2  //SETYO NUGROHO
3  // 103112400024
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 Go IDE's terminal window. The terminal has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The prompt is 'PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4>'. The user has entered 'go run "d:\A' and the program has outputted three lines: '5 10 3 10', '60 10', and '3628800 1'. The prompt is now 'PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4>' with a cursor.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS


PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4> go run "d:\A
5 10 3 10
60 10
3628800 1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4> █
```

DEKSRIPSI

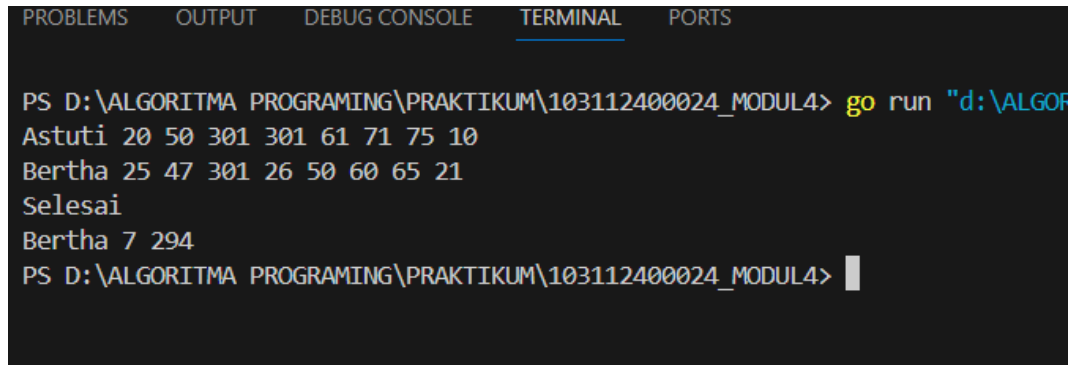
Program ini berfungsi untuk menghitung dan mencetak permutasi dan kombinasi dari tiga variabel yang diberikan oleh pengguna (a, b, c, dan d). Program ini dimulai dengan menerima inputan untuk variabel-variabel tersebut, kemudian menghitung permutasi dan kombinasi berdasarkan nilai-nilai yang dimasukkan. Fungsi permutasi() menghitung jumlah cara berbeda untuk menyusun elemen-elemen yang ada, sedangkan fungsi kombinasi() menghitung jumlah cara untuk memilih elemen-elemen tanpa memperhatikan urutan. Program ini juga menggunakan fungsi faktorial() untuk menghitung faktorial dari angka yang diberikan, yang menjadi dasar perhitungan untuk permutasi dan kombinasi. Setiap hasil dari perhitungan permutasi dan kombinasi dicetak di layar. Program ini memanfaatkan teknik rekursi untuk menghitung permutasi dan kombinasi secara lebih efisien.

UNGUIDED 2

SOURCE CODE

```
unguided2 >  unguided2.go > ...
1 package main
2 //103112400024
3 //SETYO NUGROHO
4 import "fmt"
5
6 func hitungSkor(soal int, totalwaktu *int, totalsoal *int) {
7     if soal <= 300 {
8         *totalsoal++
9         *totalwaktu += soal
10    }
11 }
12
13 func bandingkanPemain(nama1 string, totalsoal1, totalwaktu1 int, nama2 string, totalsoal2, totalwaktu2 int) (string, int, int) {
14     if totalsoal1 > totalsoal2 || (totalsoal1 == totalsoal2 && totalwaktu1 < totalwaktu2) {
15         return nama1, totalsoal1, totalwaktu1
16     }
17     return nama2, totalsoal2, totalwaktu2
18 }
19
20 func main() {
21     var nama1, nama2 string
22     var soal1, soal2 int
23     var totalwaktu1, totalwaktu2, totalsoal1, totalsoal2 int
24     var totalsoaltemp, totalwaktutemp int
25     var namatemp string
26
27     fmt.Scan(&nama1)
28     if nama1 != "Selesai" {
29         for i := 0; i < 8; i++ {
30             fmt.Scan(&soal1)
31             hitungSkor(soal1, &totalwaktu1, &totalsoal1)
32         }
33
34         for {
35             fmt.Scan(&nama2)
36             if nama2 == "Selesai" {
37                 break
38             }
39
40             totalwaktu2, totalsoal2 = 0, 0
41             for i := 0; i < 8; i++ {
42                 fmt.Scan(&soal2)
43                 hitungSkor(soal2, &totalwaktu2, &totalsoal2)
44             }
45
46             namatemp, totalsoaltemp, totalwaktutemp = bandingkanPemain(nama1, totalsoal1, totalwaktu1, nama2, totalsoal2, totalwaktu2)
47         }
48
49         fmt.Println(namatemp, totalsoaltemp, totalwaktutemp)
50     }
51 }
52
```

OUTPUT

A screenshot of a Go IDE terminal window. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS. The command prompt shows the user is in the directory D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4. The command 'go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4\main.go"' has been executed. The output of the program is displayed as follows:

```
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4\main.go"
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\PRAKTIKUM\103112400024_MODUL4>
```

DEKSRIPSI

Program ini digunakan untuk membandingkan skor antara dua pemain dalam suatu permainan. Setiap pemain menjawab serangkaian soal, dan program menghitung total skor yang diperoleh oleh masing-masing pemain berdasarkan waktu yang dibutuhkan untuk menjawab soal-soal tersebut. Program ini memanfaatkan fungsi `hitungSkor()` untuk menghitung skor dengan menambahkan poin setiap kali pemain berhasil menjawab soal, dan kemudian menambahkan waktu yang dibutuhkan. Fungsi `bandingkanPemain()` digunakan untuk membandingkan hasil total skor dan waktu antara dua pemain. Hasil perbandingan akan mencetak nama pemain yang menang berdasarkan total skor lebih tinggi atau, jika skor sama, berdasarkan waktu yang lebih cepat. Program ini menggunakan perulangan untuk memproses soal-soal dan memastikan bahwa input diberikan dengan benar sebelum melanjutkan ke perbandingan.

SOURCE CODE

```
unguided3 > -go unguided3.go > ...
1  package main
2  //103112400024
3  //SETYO NUGROHO
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("Input Tidak Valid")
25     }
26 }
```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4> go run "d:\ALGORITMA PROGRAMI
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\103112400024_MODUL4> |
```

DEKSRIPSI

Program ini menghasilkan deret angka berdasarkan aturan yang disebut sebagai *Collatz conjecture*. Pengguna memasukkan angka dan program akan terus mencetak angka berikutnya dalam deret tersebut. Jika angka genap, angka tersebut dibagi dua, dan jika angka ganjil, angka tersebut dikalikan dengan tiga lalu ditambah satu. Proses ini terus berlangsung hingga angka mencapai 1. Program hanya menerima input yang lebih kecil dari 100,000 dan mencetak deret angka sampai angka 1 tercapai. Jika input tidak valid (lebih dari 100,000), program akan mencetak pesan "Input Tidak Valid."

III. KESIMPULAN

Kesimpulan dari laporan ini dapat disimpulkan bahwa prosedur dalam pemrograman memiliki peran penting dalam menyederhanakan program yang kompleks dengan membagi instruksi menjadi potongan-potongan kecil yang lebih mudah dipahami dan dikelola. Prosedur berfungsi untuk mengurangi duplikasi kode dan meningkatkan struktur program. Selain itu, prosedur juga memungkinkan penggunaan parameter untuk memberikan fleksibilitas dalam pemrosesan data, dengan pilihan antara pass by value atau pass by reference. Kedua metode ini memiliki perbedaan signifikan dalam cara data dikirim dan diproses dalam prosedur, mempengaruhi hasil akhir dari program.

Selain itu, contoh implementasi prosedur pada program yang telah dilakukan dalam laporan ini, seperti menghitung gaji karyawan, menentukan kelulusan mahasiswa, hingga menghitung permutasi dan kombinasi, menunjukkan bagaimana prosedur dapat diaplikasikan dalam berbagai macam perhitungan matematis dan logika. Dengan menggunakan prosedur, program dapat lebih modular dan mudah di-maintain, serta mengurangi risiko kesalahan atau pengulangan kode yang tidak perlu. Penggunaan prosedur yang tepat akan sangat membantu dalam pengembangan aplikasi yang lebih efisien dan terstruktur dengan baik.

REFERENSI

MODUL 4 PROSEDUR ALGORITMA PEMOGRAMAN 2