

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 4
PROSEDUR**



Oleh:

TRI PANJI UTOMO

2311102213

IF – 11- 02

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

Prosedur adalah unit modular dari kode yang dapat dipanggil untuk melakukan tugas tertentu tetapi tidak mengembalikan nilai seperti fungsi di modul sebelumnya. Prosedur pada dasarnya adalah fungsi dengan pernyataan return yang opsional, dan sering digunakan untuk mengelompokkan kode yang perlu dieksekusi berulang kali. Contohnya seperti dibawah ini

Notasi Algoritma	
1	procedure <nama procedure> (<params>)
2	kamus
3	{deklarasi variabel lokal dari procedure}
4	...
5	algoritma
6	{badan algoritma procedure}
7	...
8	endprocedure
Notasi dalam bahasa Go	
9	func <nama procedure> <(params)> {
10	/* deklarasi variabel lokal dari procedure */
11	...
12	/* badan algoritma procedure */
13	...
14	}

Di dalam pemanggilan prosedur suatu prosedur hanya akan dieksekusi apabila dipanggil baik secara langsung atau tidak langsung oleh program utama. Tidak langsung di sini maksudnya adalah prosedur dipanggil oleh program utama melalui perantara subprogram yang lain.

Notasi Algoritma	
1	program contohprosedur
2	kamus
3	x : integer
4	algoritma
5	x ← 5
6	cetakNFibo(x) {cara pemanggilan #1}
7	cetakNFibo(100) {cara pemanggilan #2}
8	endprogram
Notasi dalam bahasa Go	
9	func main() {
10	var x int
11	x = 5
12	cetakNFibo(x) {cara pemanggilan #1}
13	cetakNFibo(100) {cara pemanggilan #2}
14	}

II. GUIDED

1. Guided 1 Source Code

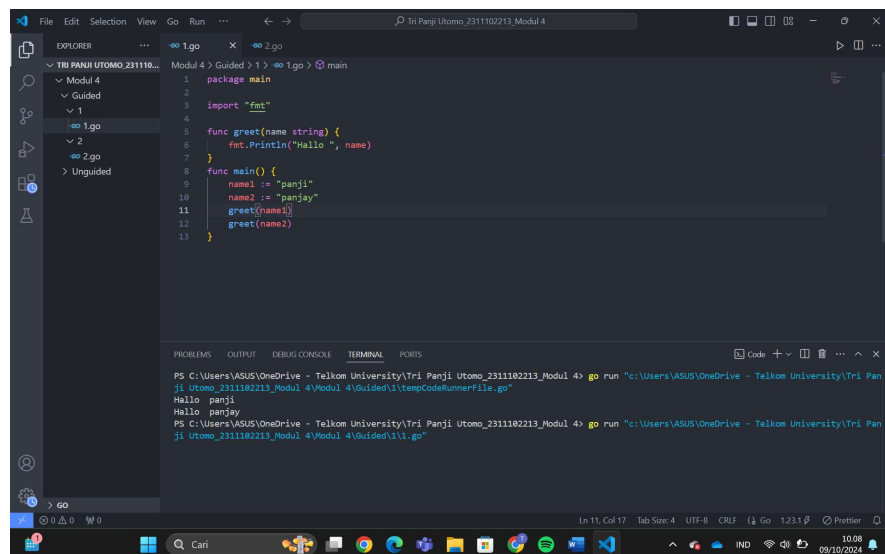
```
package main

import "fmt"

func greet(name string) {
    fmt.Println("Hallo ", name)
}

func main() {
    name1 := "panji"
    name2 := "panjay"
    greet(name1)
    greet(name2)
}
```

Screenshot



Deskripsi

Program ini memiliki tujuan untuk menyapa dua nama yang berbeda dengan menggunakan fungsi greet.

2. Guided 2

Source Code

```
package main

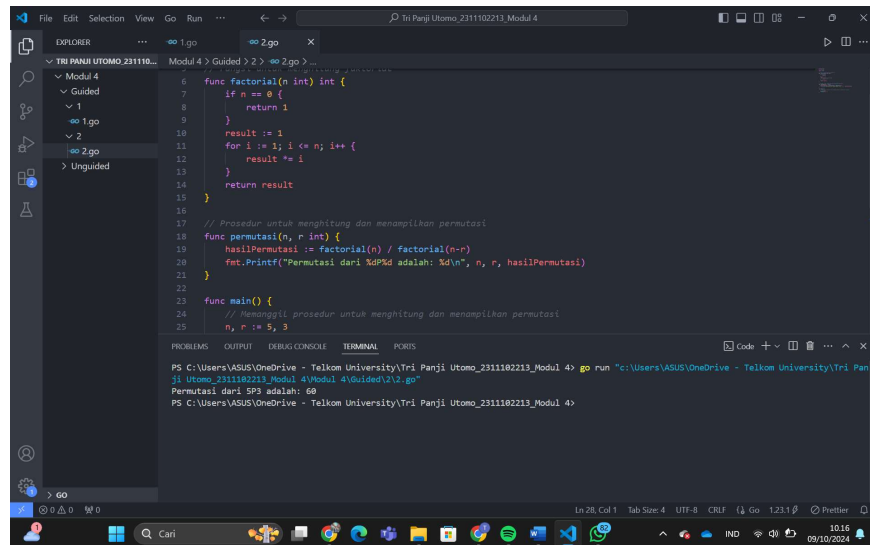
import "fmt"

// Fungsi untuk menghitung faktorial
func factorial(n int) int {
    if n == 0 {
        return 1
    }
    result := 1
    for i := 1; i <= n; i++ {
        result *= i
    }
    return result
}

// Prosedur untuk menghitung dan menampilkan permutasi
func permutasi(n, r int) {
    hasilPermutasi := factorial(n) / factorial(n-r)
    fmt.Printf("Permutasi dari %dP%d adalah: %d\n", n, r,
        hasilPermutasi)
}

func main() {
    // Memanggil prosedur untuk menghitung dan menampilkan
    permutasi
    n, r := 5, 3
    permutasi(n, r)
}
```

Screenshot



The screenshot shows a Go program in a VS Code editor. The Explorer pane on the left shows a project structure with 'Modul 4' containing 'Guided' and 'Unguided' subfolders. The '2.go' file is selected. The code defines a 'factorial' function, a 'permutasi' function, and a 'main' function. The 'main' function calls 'permutasi(5, 3)'. The terminal at the bottom shows the command 'go run' and the output 'Permutasi dari 5P3 adalah: 60'.

```
6 func factorial(n int) int {
7     if n == 0 {
8         return 1
9     }
10    result := 1
11    for i := 1; i <= n; i++ {
12        result *= i
13    }
14    return result
15 }
16
17 // Prosedur untuk menghitung dan menampilkan permutasi
18 func permutasi(n, r int) {
19     hasilPermutasi := factorial(n) / factorial(n-r)
20     fmt.Printf("Permutasi dari %dP%d adalah: %d\n", n, r, hasilPermutasi)
21 }
22
23 func main() {
24     // Memanggil prosedur untuk menghitung dan menampilkan permutasi
25     n, r := 5, 3
26 }
```

```
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 4> go run "c:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 4\Guided\2.go"
Permutasi dari 5P3 adalah: 60
PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 4>
```

Deskripsi

Program ini memiliki dua fungsi utama yaitu menghitung faktorial dan menghitung permutasi, serta menampilkan hasilnya. fungsi ini memanggil prosedur permutasi dengan argumen $n = 5$ dan $r = 3$ lalu menjalankan prosedur untuk menghitung dan menampilkan permutasi dari 5P3.

III. UNGUIDED

1. Unguided 1

Source Code

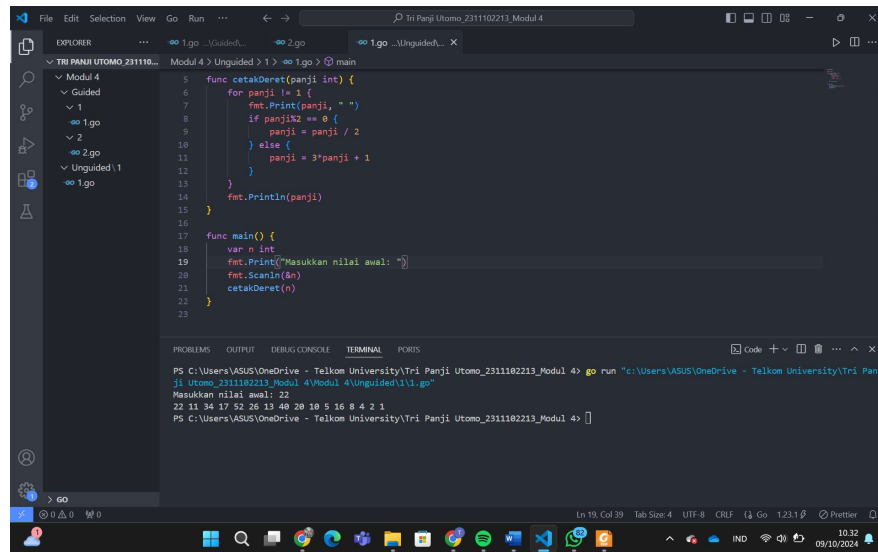
```
package main

import "fmt"

func cetakDeret(panji int) {
    for panji != 1 {
        fmt.Print(panji, " ")
        if panji%2 == 0 {
            panji = panji / 2
        } else {
            panji = 3*panji + 1
        }
    }
    fmt.Println(panji)
}

func main() {
    var n int
    fmt.Print("Masukkan nilai awal: ")
    fmt.Scanln(&n)
    cetakDeret(n)
}
```

Screenshot



```
5 func cetakDeret(panji int) {
6     for panji != 1 {
7         fmt.Print(panji, " ")
8         if panji%2 == 0 {
9             panji = panji / 2
10        } else {
11            panji = 3*panji + 1
12        }
13    }
14    fmt.Println(panji)
15}

16
17 func main() {
18     var n int
19     fmt.Print("Masukkan nilai awal: ")
20     fmt.Scanln(&n)
21     cetakDeret(n)
22 }
23
```

PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 4> go run "c:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 4\Modul 4\Unguided\1\1.go"

Masukkan nilai awal: 22

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

PS C:\Users\ASUS\OneDrive - Telkom University\Tri Panji Utomo_2311182213_Modul 4>

Deskripsi

Program yang akan mencetak deret angka dari nilai awal yang dimasukkan user hingga mencapai nilai 1 dengan menerima satu parameter yang bertipe integer dan menggunakan loop for, fungsi ini terus mencetak nilai selama tidak sama dengan 1. Jika parameter adalah bilangan genap, nilai dibagi 2, jika parameter adalah bilangan ganjil, nilai menjadi $3 \times \text{parameter} + 1$.