

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROSEDUR**



Disusun Oleh :

Yoga Hogantara / 2311102153

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Prosedur merupakan suatu dari program yang digunakan untuk menjalankan suatu tugas tertentu yang letaknya terpisah dari bagian program yang menggunakannya. Suatu prosedur digunakan dengan tujuan khusus, yaitu untuk mengerjakan suatu tugas tertentu. Keuntungan menggunakan prosedur yaitu dapat memudahkan dalam menulis dan menemukan kesalahan pada program, dan juga menghemat ukuran program.

Notasi dalam bahasa GO

```
func <nama function> (<params>) {  
/* deklarasi variable local dari prosedur */  
...  
/* badan algoritma prosedur*/  
...  
}
```

Parameter dibagi menjadi 2 yaitu parameter formal dan actual

1. Parameter Formal

Parameter formal merupakan suatu parameter yang ditulis ketika deklarasi suatu subprogram yang berfungsi sebagai petunjuk argumen

2. Parameter Aktual

Parameter aktual merupakan argumen yang digunakan pada bagian parameter ketika melakukan pemanggilan subprogram.

II. GUIDED

1. Sourcecode

```
package main

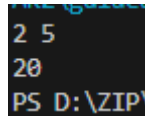
import "fmt"

func main(){
    var a, b int
    var result int
    fmt.Scan(&a,&b)
    if a >= b {
        permutasi(a,b,&result)
        fmt.Print(result)
    }else{
        permutasi(b,a,&result)
        fmt.Print(result)
    }
}

func faktorial(n int,hasil *int) {
    *hasil= 1
    for i :=1; i<=n; i++){
        *hasil = *hasil *i
    }
}

func permutasi(n,r int, hasil *int) {
    var fakn, fakr int
    faktorial(n, &fakn)
    faktorial(n-r, &fakr)
    *hasil = fakn/fakr
}
```

Screenshoot Output



```
2 5
20
PS D:\ZIP'
```

Deskripsi Program

Program tersebut menggunakan prosedur yang berfungsi untuk menghitung nilai faktorial dan permutasi dari inputan yang di masukkan oleh user. Dengan menggunakan perulangan untuk mencari faktorial dan percabangan untuk sebuah kondisi inputan. Dengan menggunakan func faktorial dan func permutasi untuk menghitung faktorial dan permutasi inputan dari user.

2. Sourcecode

```
package main

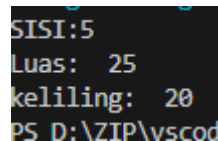
import "fmt"

func luas(s int) {
    var hasil int
    hasil=s*s
    fmt.Println("LUAS:", hasil)
}

func keliling(s int) {
    var hasil int
    hasil=s*4
    fmt.Println("KELILING:", hasil)
}

func main(){
    var s int
    fmt.Print("SISI:")
    fmt.Scan(&s)
    luas(s)
    keliling(s)
}
```

Screenshoot Output



```
SISI:5
Luas: 25
keliling: 20
PS D:\ZTP\vscode
```

Deskripsi Program

Program tersebut menggunakan prosedur yang berfungsi untuk menghitung sebuah luas dan keliling sebuah persegi, dimana sisi nya sebagai inputan yang diisi oleh user. Dengan menggunakan func luas dan func keliling untuk sebuah rumus mencari luas dan keliling untuk mempersingkat program.

III. UNGUIDED

1. Sourcecode

```
package main

import "fmt"

func faktorial(n int, hasil *int) {
    *hasil = 1
    for i := 1; i <= n; i++ {
        *hasil = *hasil * i
    }
}

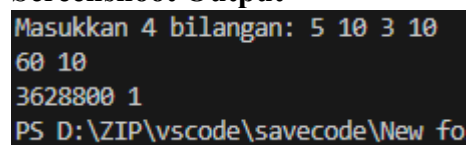
func permutasi(n, r int, hasil *int) {
    var nfaktorial, nrfakrorial int
    faktorial(n, &nfaktorial)
    faktorial(r, &nrfakrorial)
    faktorial(n-r, &nrfakrorial)
    *hasil = nfaktorial / nrfakrorial
}

func kombinasi(n, r int, hasil *int) {
    var nfaktorial, rfaktorial, nrfakrorial int
    faktorial(n, &nfaktorial)
    faktorial(r, &rfaktorial)
    faktorial(n-r, &nrfakrorial)
    *hasil = nfaktorial / (rfaktorial * nrfakrorial)
}

func main() {
    var a, b, c, d, p1, p2, c1, c2 int
    fmt.Print("Masukkan 4 bilangan: ")
    fmt.Scan(&a, &b, &c, &d)

    permutasi(a, c, &p1)
    kombinasi(a, c, &c1)
    fmt.Printf("%d %d\n", p1, c1)
    permutasi(b, d, &p2)
    kombinasi(b, d, &c2)
    fmt.Printf("%d %d\n", p2, c2)
}
```

Screenshoot Output



```
Masukkan 4 bilangan: 5 10 3 10
60 10
3628800 1
PS D:\ZIP\vscode\savecode\New fo
```

Deskripsi Program

Program tersebut menggunakan prosedur yang berfungsi untuk mencari sebuah permutasi dan kombinasi dari 4 bilangan yang di inputkan user pada output baris pertama merupakan hasil dari permutasi dan kombinasi a terhadap c, sedangkan yang kedua merupakan hasil dari permutasi dan kombinasi b terhadap d.

Dengan menggunakan rumus $P(n,r) = \frac{n!}{(n-r)!}$, $C(n,r) = \frac{n!}{r!(n-r)!}$

Untuk menghitung permutasi dan kombinasi pada program

2. Source code

```
package main

import (
    "fmt"
)

func skor(soal [8]int, skor *int, waktu
*int) {
    *skor = 0
    *waktu = 0
    for i := 0; i < 8; i++ {
        if soal[i] <= 301 {
            *skor++
            *waktu += soal[i]
        }
    }
}

func main() {
    var peserta1, peserta2 string
    var soalselesai1, soalselesai2 [8]int
    var skor1, skor2, total1, total2 int

    fmt.Print("PESERTA: ")
    fmt.Scan(&peserta1)
    fmt.Println("WAKTU Pengerjaan:")
    for i := 0; i < 8; i++ {
        fmt.Scan(&soalselesai1[i])
    }
}
```

```

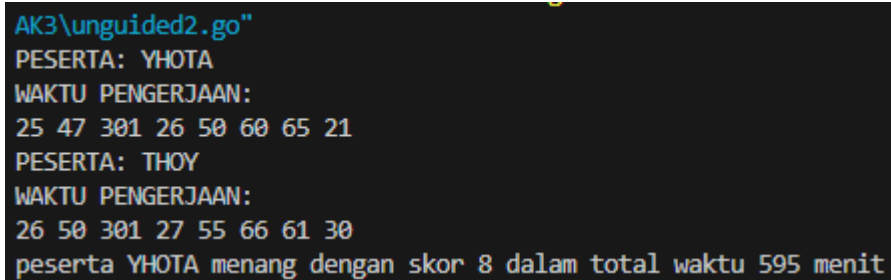
        fmt.Print("PESERTA: ")
        fmt.Scan(&peserta2)
        fmt.Println("WAKTU Pengerjaan:")
        for i := 0; i < 8; i++ {
            fmt.Scan(&soalselesai2[i])
        }

        skor(soalselesai1, &skor1, &total1)
        skor(soalselesai2, &skor2, &total2)

        if skor1 > skor2 || (skor1 == skor2 &&
total1 < total2) {
            fmt.Printf("peserta %s menang
dengan skor %d dalam total waktu %d
menit\n", peserta1, skor1, total1)
        } else if skor1 == skor2 {
            fmt.Printf("kedua peserta seri\n")
        } else {
            fmt.Printf("peserta %s menang
dengan skor %d dalam total waktu %d
menit\n", peserta2, skor2, total2)
        }
    }
}

```

Screenshoot Output



```

AK3\unguided2.go"
PESERTA: YHOTA
WAKTU Pengerjaan:
25 47 301 26 50 60 65 21
PESERTA: THOY
WAKTU Pengerjaan:
26 50 301 27 55 66 61 30
peserta YHOTA menang dengan skor 8 dalam total waktu 595 menit

```

Deskripsi Program

Pada program tersebut menggunakan prosedur yang berfungsi sebagai menghitung waktu pengerjaan dari peserta yang di inputkan user, dengan catatan maksimal waktu 301menit. Program tersebut mencari pemenang lomba berdasarkan waktu pengerjaan dan jumlah soal yang mana jika peserta mengerjakan soal paling banyak dan waktu paling singkat maka peserta itulah pemenangnya.

3. Source code

```
package main

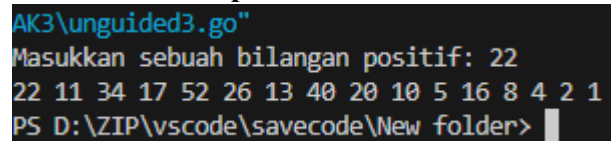
import "fmt"

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

func main() {
    var n int

    fmt.Print("Masukkan sebuah bilangan positif: ")
    fmt.Scan(&n)
    deret(n)
}
```

Screenshoot Output



```
AK3\unguided3.go"
Masukkan sebuah bilangan positif: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\ZIP\vscode\savecode\New folder>
```

Deskripsi Program

Program tersebut menggunakan prosedur yang berfungsi untuk menghitung deret yang di inputkan oleh user, dengan deret dimulai dengan bilangan bulat n , jika n genap maka suku berikutnya setengah n , namun jika ganjil maka suku berikutnya $3n+1$, Rumus tersebut digunakan hingga suku berikut bernilai 1