

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

**MODUL 4
PROCEDURE**



Oleh:

WILDAN DAFFA' HAKIM PUTRA ANTARA

2311102055

S1IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

Sedikit berbeda dengan fungsi, procedure merupakan potongan instruksi yang digunakan untuk mengurangi kerumitan program namun tidak mengembalikan nilai.

Dalam deklarasi, fungsi dan prosedur dipisah dengan blok utamanya dan pada penamaan procedure biasanya diberikan nama sesuai dengan prosesnya cetak, hitungRerata, cariNilai,

Berikut merupakan cara deklarasi procedure

Notasi Algoritma

```
procedure <nama procedure> (<params>) kamus
    {deklarasi variabel lokal dari procedure}
    ...
algoritma
    {badan algoritma procedure}
    ...
endprocedure
```

Notasi dalam Bahasa GO

```
func <nama procedure> <(params)> {
    /* deklarasi variabel lokal dari procedure */
    ...
    /* badan algoritma procedure */
    ...
}
```

Cara pemanggilan procedure hampir sama dengan fungsi hanya saja kita tinggal menuliskan nama beserta parameter saja tanpa perlu menyimpan nilainya karena procedure tidak mengembalikan nilai melalui return. Berikut contoh pemanggilan procedure

Notasi Algoritma

```
program ContohProsedur
kamus
    s real
algoritma
    s ← 4.8;
    luasPersegi(s)
    luasPersegi(3)
endprogram
```

Notasi dalam Bahasa GO

```
func main() {
    var s float64
    s = 4.8
    luasPersegi (s)
    luasPersegi(22)
}
```

II. GUIDED

1. Buatlah sebuah program menggunakan prosedur dan fungsi yang digunakan untuk menghitung nilai permutasi.

```
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)
}
```

```
MacBook-Air-Daffa Alpro2 → go run "/Users/daffaha
go"
Permutasi dari 5P3 adalah: 60
MacBook-Air-Daffa Alpro2 →
```

Program ini digunakan untuk menghitung permutasi dari dua bilangan, pada program ini terdapat sub program yaitu fungsi faktorial dan prosedur permutasi. Pada fungsi faktorial, dilakukan perhitungan nilai faktorial dengan melakukan perulangan sebanyak nilai variabel formal n dan disetiap perulangan nilai i akan dikalikan nilai variabel result dan nantinya akan mengembalikan nilai dari variable result. Pada fungsi permutasi akan dilakukan pembagian nilai n faktorial dengan n-r faktorial di mana nilai ini didapat dari pemanggilan fungsi faktorial dengan nilai parameter aktual n dan n-r dan disimpan pada variabel hasilPermutasi, karena tidak mengembalikan nilai, maka hasil dicetak ke terminal dengan printf. Pada fungsi utamanya dideklarasikan variabel n, r dengan nilai 5, 3 lalu memanggil prosedur permutasi.

III. UNGUIDED

1. Skiena dan Revilla dalam Programming Challenges mendefinisikan sebuah deret bilangan. Deret dimulai dengan sebuah bilangan bulat n. Jika bilangan n saat itu genap, maka suku berikutnya adalah $\frac{1}{2}n$, tetapi jika ganjil maka suku berikutnya bernilai $3n+1$. Rumus yang sama digunakan terus menerus untuk mencari suku berikutnya. Deret

berakhir ketika suku terakhir bernilai 1. (Soal lengkap terdapat di modul)

```
package main

import "fmt"

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

func main() {
    var n int
    fmt.Scan(&n)
    if n > 1000000 {
        fmt.Println("Bilangan harus <=
1000000")
    } else {
        cetakDeret(n)
    }
}
```

```
λ MacBook-Air-Daffa Alpro2 → go run "/Users/daffahakim/Documents/Kuliah
ded3.go"
22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
λ MacBook-Air-Daffa Alpro2 → go run "/Users/daffahakim/Documents/Kuliah
ded3.go"
35
35 106 53 160 80 40 20 10 5 16 8 4 2 1
```

Program ini akan menghasilkan suatu deret yang selalu berakhir satu jika masukkan masih di bawah 1000000. Terdapat satu sub program yang merupakan prosedur yaitu cetakDeret di mana prosedur ini mencetak nilai n lalu mengecek jika n dimodulus 2 habis (menandakan genap) maka prosedur akan memanggil dirinya sendiri dengan parameter aktual $n/2$ (rekursif). Jika ini tidak terpenuhi dan n bukan merupakan 1 maka prosedur cetakDeret akan dipanggil lagi dengan parameter aktual $(3*n+1)$ jika semuanya tidak memenuhi maka akan mencetak baris baru (untuk menghilangkan “%” di akhir, kondisi else ini bisa dihilangkan) dan prosedur ini berakhir. Pada fungsi utama,

program akan meminta masukkan 1 angka yang akan disimpan pada variabel `n` jika nilainya lebih dari 1000000 maka akan dihasilkan output “Bilangan harus \leq 1000000” namun jika tidak maka prosedur `cetakDeret` akan dipanggil dengan parameter aktual nilai `n` yang telah dimasukkan.