

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL IV

PROSEDUR



Oleh:

AHMADAN SYARIDIN

2311102038

S1-IF11-02

S1 TEKNIK INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1. Dasar teori

Prosedur merupakan potongan beberapa intruksi program menjadi suatu intruksi 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 sub program dikatakan prosedur apabila

1. tidak ada deklarasi tipe nilai yang dikembalikan
2. tidak terdapat kata kunci return Dalam badan subprogram

Berikut ini adalah cara penulisan Deklarasi prosedur pada notasi pseudocode dan golang

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	}

Penulisan berada di luar blok dari program utama atau fungsi main() pada suatu program go, dan bisa ditulis sebelum atau sesudah blok program utama

Deklarasi mencetak n nilai pertama dari deret Fibonacci

	Notasi Algoritma
1	procedure cetakNFibo(in n : integer)
2	kamus
3	f1, f2, f3, i : integer
4	algoritma
5	f2 ← 0
6	f3 ← 1
7	for i ← 1 to n do
8	output(f3)
9	f1 ← f2
10	f2 ← f3
11	f3 ← f1 + f2
12	endfor
13	endprocedure
	Notasi dalam bahasa Go
14	func cetakNFibo(n int) {
15	var f1, f2, f3 int
16	f2 = 0
17	f3 = 1
18	for i := 1; i <= n; i++ {
19	fmt.Println(f3)
20	f1 = f2
21	f2 = f3
22	f3 = f1 + f2
23	}
24	}

Prosedur hanya akan di eksekusi bila dipanggil secara langsung atau tidak langsung oleh program utama . tidak langsung yang di maksud adalah prosedur dipanggil oleh program utama melalui perantara subprogram yang lain.

Pemanggilannya sendiri cukup mudah, yaitu dengan menuliskan nama beserta parameter atau argument yang diminta dari suatu prosedur.

	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	}

2.Guided

Source code

```
package main

import "fmt"

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

func main() {
    name1 := "ahmadan"
    name2 := "dimas"
    name3 := "agha"
    greet(name1)
    greet(name2)
    greet(name3)
}
```

Screenshot

```
PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4> go run "c:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_mo
Hallo ahmadan
Hallo dimas
Hallo agha
PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4>
```

Deskripsi

Program ini adalah program Go sederhana yang menggunakan fungsi bernama greet untuk mencetak sapaan kepada beberapa nama. Fungsi greet menerima satu parameter berupa string dan mencetak pesan "Hallo" diikuti oleh nama tersebut. Di dalam fungsi utama (main), tiga variabel string (name1, name2, name3) dideklarasikan dengan nilai masing-masing "ahmadan", "dimas", dan "agha". Fungsi greet kemudian dipanggil tiga kali untuk setiap variabel, sehingga menghasilkan keluaran berupa sapaan untuk setiap nama yang telah didefinisikan.

3.Unguided

1.Unguided 1

Sourcode

```
package main

import "fmt"

func factorial(n int) int {
    if n == 0 || n == 1 {
        return 1
    }
    result := 1
    for i := 2; i <= n; i++ {
        result *= i
    }
}
```

```

return result
}

// Fungsi buat menghitung permutasi
func permutation(n, r int) int {
    return factorial(n) / factorial(n-r)
}

// Fungsi buat menghitung kombinasi
func combination(n, r int) int {
    return factorial(n) / (factorial(r) * factorial(n-r))
}

func main() {
    // Input 4 bilangan
    var a, b, c, d int

    fmt.Println("Masukkan bilangan a, b, c, d (dengan spasi): ")
    fmt.Scanf("%d %d %d %d", &a, &b, &c, &d)

    // Cek syarat a >= c dan b >= d
    if a >= c && b >= d {
        // Menghitung permutasi dan kombinasi a dan c
        permutasiAC := permutation(a, c)
        kombinasiAC := combination(a, c)

        // Menghitung permutasi dan kombinasi b dan d
        permutasiBD := permutation(b, d)
        kombinasiBD := combination(b, d)

        // Output hasil
        fmt.Println("Permutasi(a, c) dan Kombinasi(a, c):", permutasiAC, kombinasiAC)
    }
}

```

```

        fmt.Println("Permutasi(b, d) dan Kombinasi(b, d):", permutasiBD, kombinasiBD)

    } else {

        fmt.Println("Syarat a >= c dan b >= d tidak terpenuhi.")

    }

}

```

Screenshot

```

PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4> go run "c:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4\main.go"
Masukkan bilangan a, b, c, d (dengan spasi):
19 12 10 05
Permutasi(a, c) dan Kombinasi(a, c): 335221286400 92378
Permutasi(b, d) dan Kombinasi(b, d): 95040 792
PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4>

```

Deskripsi

Program di atas menghitung permutasi dan kombinasi dari dua pasang angka. Pertama, pengguna diminta memasukkan empat angka: a, b, c, dan d. Program lalu memeriksa apakah a lebih besar atau sama dengan c, dan b lebih besar atau sama dengan d. Jika ya, program akan menghitung permutasi dan kombinasi untuk pasangan angka a dengan c, serta b dengan d, dan menampilkan hasilnya. Jika tidak, program hanya akan memberi tahu bahwa syarat tidak terpenuhi.

2.Unguided 2

Source code

```

package main

import "fmt"

func cetakDeret(n int) {

    for n != 1 {

        fmt.Printf("%d ", n)
    }
}

```

```

    if n%2 == 0 {
        n = n / 2
    } else {
        n = 3*n + 1
    }
}

fmt.Println(n)
}

func main() {
    var n int

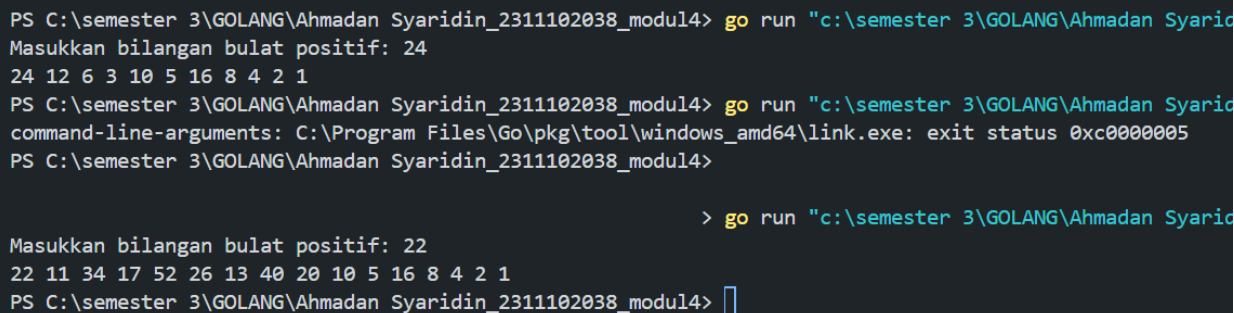
    fmt.Print("Masukkan bilangan bulat positif: ")

    fmt.Scan(&n)

    cetakDeret(n)
}

```

Screenshot



```

PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4> go run "c:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4\main.go"
Masukkan bilangan bulat positif: 24
24 12 6 3 10 5 16 8 4 2 1
PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4> go run "c:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4\main.go"
command-line-arguments: C:\Program Files\Go\pkg\tool\windows_amd64\link.exe: exit status 0xc0000005
PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4>
> go run "c:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4\main.go"
Masukkan bilangan bulat positif: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS C:\semester 3\GOLANG\Ahmadan Syaridin_2311102038_modul4>

```


Deskripsi

Program Go tersebut mencetak deret berdasarkan aturan tertentu yang ditentukan oleh nilai awal n yang dimasukkan pengguna. Fungsi `cetakDeret(n int)` menggunakan loop untuk mencetak nilai n hingga mencapai 1, dengan mengikuti dua aturan: jika n genap, nilai berikutnya adalah $n/2$, dan jika n ganjil, nilai berikutnya adalah $3n + 1$. Proses ini terus berlanjut hingga mencapai angka 1, yang juga dicetak sebagai bagian dari deret. Di dalam fungsi `main()`, program meminta pengguna untuk memasukkan bilangan bulat positif dan kemudian memanggil fungsi `cetakDeret(n)` untuk menampilkan deret yang dihasilkan dari input tersebut.