**LAPORAN PRAKTIKUM**

**ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV**

**PROSEDUR**

****

**DISUSUN OLEH :**

**WILDAN MAULANA ZIDAN**

**2311102162**

**IF 11 02**

**S1 TEKNIK INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

1. **DASAR TEORI**
   1. **Definisi Procedure**

Prosedur dapat dianggap sebagai potongan beberapa instruksi program menjadi suatu instruksi baru yang dibuat unutk mengurangi kerumitan dari kode program yang kompleks pada suatu program yang besar, dikatakan prosedur apabila tidak ada deklarasi tipe nilai yang dikembalikan dan tidak terdapat kata kunci return dalam badan subprogram.

* 1. **Deklarasi Procedure**

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

* 1. **Cara pemanggilan Procedure**

Pemanggilan suatau procedure cukup mudah, yaitu dengan hanya

menuliskamnama beserta parameter atau argument yang diminta dari suatu prosedur.

Sebagai contoh prosedur cetak NFibo di atas dipanggil dengan menuliskan Namanya,

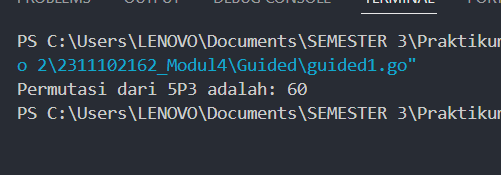
kemudian sebuah variabel atau nilai int tertentu sebagai argument parameter n.

Pemanggilan suatau procedure cukup mudah, yaitu dengan hanya menuliskam nama beserta parameter atau argument yang diminta dari suatu prosedur. Sebagai contoh prosedur cetak NFibo di atas dipanggil dengan menuliskan Namanya, kemudian sebuah variabel atau nilai int tertentu sebagai argument parameter n.

1. **GUIDED**

Guided 1

|  |
| --- |
| 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)  } |

   
  
  
  
  
  
  
  
  
Kode di atas menghitung dan menampilkan nilai modifikasi dari dua angka "n" dan "r". Pertama, fungsi 'faktor' digunakan untuk menghitung faktor suatu bilangan. Metode yang dimodifikasi kemudian menghitung penilaian dengan menghitung n! /(n-r)!`, menampilkan hasil dalam format yang ditentukan. Pada fungsi utama, program memanggil metode 'permutasi' dengan nilai 'n=5' dan 'r=3', sehingga menampilkan hasil permutasi '5P3'.

1. **UNGUIDED**

Unguided 3

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 1/2n, 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.

|  |
| --- |
| package main  import "fmt"  func hasilkanDeret(bilangan int) {  for bilangan != 1 {  fmt.Printf("%d ", bilangan)  if bilangan%2 == 0 {  bilangan /= 2  } else {  bilangan = 3\*bilangan + 1  }  }  fmt.Println(1)  }  func main() {  var bilangan int  fmt.Print("Masukkan bilangan : ")  fmt.Scan(&bilangan)  if bilangan > 1000000 {  fmt.Println("Bilangan harus kurang dari 1000000 !!!")  } else {  hasilkanDeret(bilangan)  }  } |

kode di atas menghasilkan urutan berdasarkan angka yang dimasukkan pengguna menggunakan algoritma Collatz (atau 3n+1). Ketika pengguna memasukkan angka, program akan memeriksa apakah angka tersebut lebih besar dari 1.000.000. Jika demikian, program akan menampilkan pesan kesalahan. Jika tidak, fungsi 'generateSeries' akan dijalankan dan mencetak seri tersebut. Cara kerjanya berurutan sesuai aturan: jika bilangan genap, bagi dengan 2. Jika ganjil, kalikan dengan 3 dan tambahkan 1.