

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROCEDUR**



Disusun Oleh :

Rakha Arbiyandanu / 2311102263

IF-11-6

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Prosedur adalah blok kode yang dapat dipanggil untuk melakukan tugas tertentu. Ini sering digunakan untuk mengatur kode dengan cara yang lebih modular dan terstruktur. Dalam bahasa pemrograman Go (Golang), konsep prosedur direpresentasikan oleh suatu fungsi.

Perbedaan Function dan Procedure dalam Go

Function:

- Menghasilkan atau mengembalikan nilai.
- Nama fungsi biasanya menggambarkan kata benda atau sifat.
- Memiliki tipe data yang jelas.
- Mengandung kata kunci return.
- Parameter dapat digunakan sebagai acuan secara langsung.

Procedure:

- Membentuk suatu instruksi baru
- Nama fungsi membentuk kata kerja
- Tidak memiliki tipe data
- Tidak terdapat "return"
- Tidak dapat menggunakan parameter secara langsung

Perbedaan parameter aktual dengan parameter formal

1. Parameter aktual

Merupakan variabel atau ekspresi dalam parameter yang digunakan untuk memanggil sebuah sub program. Misal:
`raise_salary(emp_nim.amount)`

2. Parameter formal

Merupakan variabel yang ada didalam parameter list pada sebuah program. Misal: `raise_salary emp_id INTEGER, amount REAL`

I. GUIDED

Soal Studi Case

Buatlah sebuah program beserta prosedur yang digunakan untuk menampilkan suatu pesan error, warning atau Informasi berdasarkan masukan dari user. Masukan terdiri dari sebuah bilangan bulat flag (0 s.d. 2) dan sebuah string pesan M. Keluaran berupa string pesan M beserta jenis pesannya, yaitu error, warning atau informasi berdasarkan nilai flag 0, 1 dan 2 secara berturut-turut.

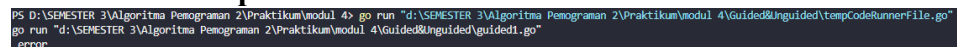
Sourcecode

```
package main
import "fmt"

func main(){
    var bilangan int
    var pesan string
    fmt.Scan(&bilangan, &pesan)
    cetakPesan(pesan, bilangan)
}

func cetakPesan(M string, flag int){
    var jenis string = ""
    if flag == 0 {
        jenis = "error"
    }else if flag == 1{
        jenis = "warning"
    }else if flag ==2 {
        jenis = "informasi"
    }
    fmt.Println(M, jenis)
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4\Guided&Unguided\tempCodeRunnerFile.go"
go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4\Guided&Unguided\guided1.go"
error
```

Deskripsi Program

Kode Go yang dikasih mendefinisikan fungsi utama main, yang minta pengguna untuk memasukkan sebuah bilangan bulat dan sebuah string. Kode ini juga mendefinisikan fungsi cetakPesan, yang menerima dua parameter: sebuah string dan sebuah integer (flag). Fungsi ini menginisialisasi variabel jenis dengan "Hello Word" dan mengubah menjadi "error", "warning", atau "informasi"

II. GUIDED

Soal Studi Case

Mencoba mengirim pesan atau notification

Sourcecode

```
package main

import "fmt"

func sendEmailNotification(email string) {
    fmt.Printf("Mengirim email ke %s: Pendaftaran berhasil.\n", email)
}

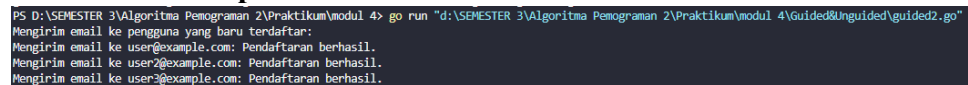
func main() {

    emails := []string{"user@example.com",
"user2@example.com", "user3@example.com"}

    fmt.Println("Mengirim email ke pengguna yang baru terdaftar:")

    for _, email := range emails {
        sendEmailNotification(email)
    }
}
```

Screenshot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4\Guided&Unguided\guided2.go"
Mengirim email ke pengguna yang baru terdaftar:
Mengirim email ke user@example.com: Pendaftaran berhasil.
Mengirim email ke user2@example.com: Pendaftaran berhasil.
Mengirim email ke user3@example.com: Pendaftaran berhasil.
```

Deskripsi Program

sendEmailNotification menerima alamat email sebagai parameter dan mencetak pesan bahwa email telah dikirim, menandakan pendaftaran berhasil. Dalam fungsi main, sebuah slice berisi tiga alamat email dideklarasikan. Program menghasilkan pesan pengantar sebelum menggunakan loop untuk iterasi melalui setiap alamat email dalam slice, memanggil fungsi sendEmailNotification untuk masing-masing email.

III. UNGUIDED

Soal Studi Case

Sourcecode

```
package main

import "fmt"

// Fungsi f1 mengembalikan hasil bertipe float64
func f1(x, y int) float64 {
    var hasil float64
    hasil = float64(2*x) - 0.5*float64(y) + 3.0
    return hasil
}

// Fungsi f2 menggunakan pointer untuk menyimpan hasil di 'hasil'
func f2(x, y int, hasil *float64) {
    *hasil = float64(2*x) - 0.5*float64(y) + 3.0
}

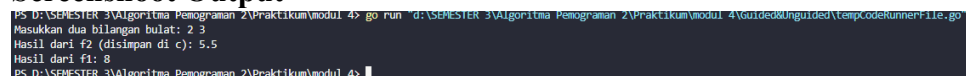
func main() {
    var a, b int
    var c float64

    fmt.Print("Masukkan dua bilangan bulat: ")
    fmt.Scan(&a, &b)

    // Memanggil f2 untuk menyimpan hasil di c
    f2(a, b, &c)
    fmt.Println("Hasil dari f2 (disimpan di c):", c)

    // Memanggil f1 dan menyimpan hasil di resultF1
    resultF1 := f1(b, a)
    fmt.Println("Hasil dari f1:", resultF1)
}
```

Screenshot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4\unguided\unguided\tempcodekunner\file.go"
Masukkan dua bilangan bulat: 2 3
Hasil dari f2 (disimpan di c): 5.5
Hasil dari f1: 8
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4>
```

Deskripsi Program

Program di atas mengandung dua f1 dan f2. Fungsi f1 mengembalikan hasil perhitungan dalam bentuk float64, sehingga berfungsi seperti fungsi biasa yang menghasilkan nilai. Sebaliknya, f2 menggunakan pointer untuk menyimpan hasil perhitungan di variabel yang diakses dari luar, tanpa

mengembalikan nilai langsung, sehingga mendekati konsep prosedur. Program ini meminta input dua bilangan bulat dari pengguna, lalu menggunakan kedua fungsi tersebut untuk menghitung dan mencetak hasil berdasarkan input yang diberikan.

IV. UNGUIDED

Soal Studi Case

1) Minggu ini, mahasiswa Fakultas Informatika mendapatkan tugas dari mata kuliah matematika

laman 44 | Modul Praktikum Algoritma dan Pemrograman 2

diskrit untuk mempelajari kombinasi dan permutasi. Jonas salah seorang mahasiswa, iseng untuk mengimplementasikannya ke dalam suatu program. Oleh karena itu bersediakah kalian membantu Jonas? (tidak tentunya ya :p)

Masukan terdiri dari empat buah bilangan asli a, b, c , dan d yang dipisahkan oleh spasi, dengan syarat $a \geq c$ dan $b \geq d$.

Keluaran terdiri dari dua baris. Baris pertama adalah hasil permutasi dan kombinasi a terhadap c , sedangkan baris kedua adalah hasil permutasi dan kombinasi b terhadap d .

Catatan: permutasi (P) dan kombinasi (C) dari n terhadap r ($n \geq r$) dapat dihitung dengan menggunakan persamaan berikut!

$$P(n, r) = \frac{n!}{(n-r)!}, \text{ sedangkan } C(n, r) = \frac{n!}{r!(n-r)!}$$

Sourcecode

```
package main

import "fmt"

func main() {
    var a, b, c int

    fmt.Print("Masukkan 3 angka : ")
    fmt.Scan(&a, &b, &c)

    fmt.Println(fogoh(a))
    fmt.Println(gohof(b))
    fmt.Println(hofog(c))
}

func f(x int) int {
    return x * x
}

func g(x int) int {
    return x - 2
}

func h(x int) int {
    return x + 1
}

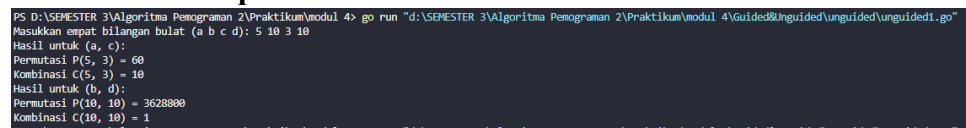
func fogoh(x int) int {
    return f(g(h(x)))
}

func gohof(x int) int {
```

```
        return g(h(f(x)))
    }

    func hofog(x int) int {
        return h(f(g(x)))
    }
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4\Guided8\unguided\unguided1.go"
Masukkan empat bilangan bulat (a b c d): 5 10 3 10
Hasil untuk (a, c):
Permutasi P(5, 3) = 60
Kombinasi C(5, 3) = 10
Hasil untuk (b, d):
Permutasi P(10, 10) = 3628800
Kombinasi C(10, 10) = 1
```

Deskripsi Program

Program diatas menghitung kombinasi dan permutasi dari dua pasang angka,Pengguna melakukan inputan 4 inputan program memiliki funtion 3 untuk factorial,satu untuk permutasi,dan satu lagi untuk kombinasi

V. UNGUIDED

Soal Studi Case

Kompetisi pemrograman tingkat nasional berlangsung ketat. Setiap peserta diberikan 8 soal yang harus dapat diselesaikan dalam waktu 5 jam saja. Peserta yang berhasil menyelesaikan soal paling banyak dalam waktu paling singkat adalah pemenangnya.

Buat program **gema** yang mencari pemenang dari daftar peserta yang diberikan. Program harus dibuat modular, yaitu dengan membuat prosedur `hitungSkor` yang mengembalikan total soal dan total skor yang dikerjakan oleh seorang peserta, melalui parameter formal. Pembacaan nama peserta dilakukan di program utama, sedangkan waktu pengerjaan dibaca di dalam prosedur.

Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

func hitungSkor(waktu []int, soal *int, skor *int) {
    *soal = 0
    *skor = 0
    for _, t := range waktu {
        if t < 301 {
            *soal++
            *skor += t
        }
    }
}

func main() {
    var pemenang string
    var soalMax, skorMin int
    soalMax = 0
    skorMin = 999999

    for {
        var nama string
        var waktu [8]int
```

```

        fmt.Print("Masukkan nama peserta dan waktu
penyelesaian soal (ketik 'Selesai' untuk berhenti): ")
        fmt.Scan(&nama)

        if strings.ToLower(nama) == "selesai" {
            break
        }

        for i := 0; i < 8; i++ {
            fmt.Scan(&waktu[i])
        }

        var soal, skor int
        hitungSkor(waktu[:], &soal, &skor)

        if soal > soalMax || (soal == soalMax && skor <
skorMin) {
            pemenang = nama
            soalMax = soal
            skorMin = skor
        }
    }

    if pemenang != "" {
        fmt.Printf("Pemenang: %s dengan %d soal
diselesaikan dan total waktu %d menit\n", pemenang, soalMax,
skorMin)
    } else {
        fmt.Println("Tidak ada peserta.")
    }
}

```

Screenshoot Output

```

PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4\Guided&Unguided\unguided\unguided2.go"
Masukkan nama peserta dan waktu penyelesaian soal (ketik 'Selesai' untuk berhenti): Rakha
20 50 301 301 61 71 75 10
Masukkan nama peserta dan waktu penyelesaian soal (ketik 'Selesai' untuk berhenti): Bertha 25 47 301 26 50 60 65 21
Masukkan nama peserta dan waktu penyelesaian soal (ketik 'Selesai' untuk berhenti): Selesai
Pemenang: Bertha dengan 7 soal diselesaikan dan total waktu 294 menit
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4>

```

Deskripsi Program

Program ini digunakan untuk pemenang dari kompetisi pemrograman berdasarkan waktu penyelesaian soal. Peserta memasukkan nama mereka diikuti oleh waktu yang diperlukan untuk menyelesaikan 8 soal. Program menghitung jumlah soal yang diselesaikan (waktu < 301 menit) dan total waktu yang dibutuhkan untuk soal yang berhasil diselesaikan. Pemenang ditentukan berdasarkan jumlah soal terbanyak yang diselesaikan. Jika jumlah soal yang diselesaikan sama, pemenang adalah peserta dengan total waktu tercepat. Program akan mencetak nama pemenang beserta jumlah soal yang diselesaikan dan total waktu yang dibutuhkan.

VI. UNGUIDED

Soal Studi Case

Buat program **sklena** yang akan mencetak setiap suku dari deret yang dijelaskan di atas untuk nilai suku awal yang diberikan. Pencetakan deret harus dibuat dalam prosedur cetakDeret yang mempunyai 1 parameter formal, yaitu nilai dari suku awal.

Sourcecode

```
package main

import (
    "fmt"
)

func cetakDeret(n int) {
    deret := []int{}

    deret = append(deret, n)

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

        deret = append(deret, n)
    }

    for i, v := range deret {
        if i == len(deret)-1 {
            fmt.Print(v)
        } else {
            fmt.Print(v, " ")
        }
    }
    fmt.Println()
}

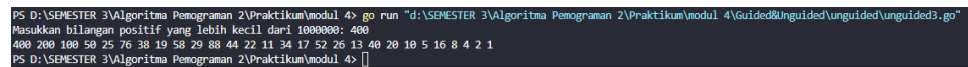
func main() {
    var n int

    fmt.Print("Masukkan bilangan positif yang lebih kecil dari
1000000: ")
}
```

```
        fmt.Scan(&n)

        if n > 0 && n < 1000000 {
            cetakDeret(n)
        } else {
            fmt.Println("Input tidak valid. Pastikan bilangan positif dan
kurang dari 1000000.")
        }
    }
}
```

Screenshoot Output



```
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4> go run "d:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4\Guided\Unguided\unguided3.go"
Masukkan bilangan positif yang lebih kecil dari 1000000: 400
400 399 398 397 396 395 394 393 392 391 390 389 388 387 386 385 384 383 382 381 380 379 378 377 376 375 374 373 372 371 370 369 368 367 366 365 364 363 362 361 360 359 358 357 356 355 354 353 352 351 350 349 348 347 346 345 344 343 342 341 340 339 338 337 336 335 334 333 332 331 330 329 328 327 326 325 324 323 322 321 320 319 318 317 316 315 314 313 312 311 310 309 308 307 306 305 304 303 302 301 300 299 298 297 296 295 294 293 292 291 290 289 288 287 286 285 284 283 282 281 280 279 278 277 276 275 274 273 272 271 270 269 268 267 266 265 264 263 262 261 260 259 258 257 256 255 254 253 252 251 250 249 248 247 246 245 244 243 242 241 240 239 238 237 236 235 234 233 232 231 230 229 228 227 226 225 224 223 222 221 220 219 218 217 216 215 214 213 212 211 210 209 208 207 206 205 204 203 202 201 200 199 198 197 196 195 194 193 192 191 190 189 188 187 186 185 184 183 182 181 180 179 178 177 176 175 174 173 172 171 170 169 168 167 166 165 164 163 162 161 160 159 158 157 156 155 154 153 152 151 150 149 148 147 146 145 144 143 142 141 140 139 138 137 136 135 134 133 132 131 130 129 128 127 126 125 124 123 122 121 120 119 118 117 116 115 114 113 112 111 110 109 108 107 106 105 104 103 102 101 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
PS D:\SEMESTER 3\Algoritma Pemrograman 2\Praktikum\modul 4>
```

Deskripsi Program

Program ini dirancang untuk mencetak deret bilangan berdasarkan aturan tertentu. Deret mulai dengan sebuah bilangan positif nnn yang lebih kecil dari 1.000.000. Proses ini berlanjut hingga mencapai suku terakhir yang bernilai 1. Program akan menerima input dari pengguna, validasi input tersebut, dan mencetak semua suku dari deret dalam satu baris dengan spasi sebagai pemisah.