

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV
PROCEDURE**



Disusun Oleh :

Muhammad Djoko Susilo / 2311102212

IF 11-06

Dosen Pengampu :

ABEDNEGO ADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Prosedur

Prosedur

Prosedur adalah subprogram yang tidak mengembalikan nilai dan tidak menggunakan kata kunci `return`. Prosedur digunakan untuk mengurangi kompleksitas kode dan memberikan efek langsung saat dipanggil. Nama prosedur biasanya berupa kata kerja yang mewakili proses, seperti `cetak`, `hitung`, atau `belok`.

Pemanggilan Prosedur

Prosedur hanya akan dieksekusi jika dipanggil, baik langsung dari program utama atau dari subprogram lain. Pemanggilannya cukup dengan menulis nama prosedur beserta argumen yang diminta.

Parameter Prosedur

Parameter memungkinkan komunikasi antara subprogram dan pemanggilnya. Parameter bisa berupa parameter formal (dideklarasikan dalam subprogram) dan parameter aktual (diberikan saat pemanggilan).

Saat memanggil sebuah fungsi atau prosedur, kita seringkali perlu memberikan data sebagai input. Cara data ini dikirimkan ke fungsi atau prosedur disebut dengan mekanisme passing parameter. Ada dua mekanisme utama:

Pass by value: Dalam metode ini, nilai dari variabel yang kita kirimkan akan disalin ke dalam variabel baru di dalam fungsi atau prosedur. Perubahan pada variabel baru ini tidak akan mempengaruhi nilai variabel asli di luar fungsi atau prosedur. Ini seperti membuat fotokopi dari sebuah dokumen. Perubahan pada fotokopi tidak akan mengubah dokumen aslinya.

Pass by reference: Berbeda dengan pass by value, dalam metode ini, kita tidak mengirimkan nilai variabel, tetapi kita mengirimkan alamat memori dari variabel tersebut. Artinya, fungsi atau prosedur akan langsung bekerja pada data asli. Jika kita mengubah data di dalam fungsi atau prosedur, maka perubahan ini juga akan terlihat pada data asli di luar fungsi atau prosedur. Ini seperti memberikan seseorang kunci rumah kita. Mereka bisa langsung masuk dan mengubah apapun di dalam rumah kita.

II. GUIDED

1. Soal Studi Case

Buatlah sebuah program yang bisa menampilkan output ketika input bernilai 1 maka "Error", 2 maka "Warning" dan 3 maka "Informasi" menggunakan prosedur.

Sourcecode

```
package main

import (
    "fmt"
)

func cetakPesan(pesan string, flag int) {
    var jenis string
    if flag == 0 {
        jenis = "Error"
    } else if flag == 1 {
        jenis = "Warning"
    } else if flag == 2 {
        jenis = "Informasi"
    }
    fmt.Printf("%s: %s\n", jenis, pesan)
}

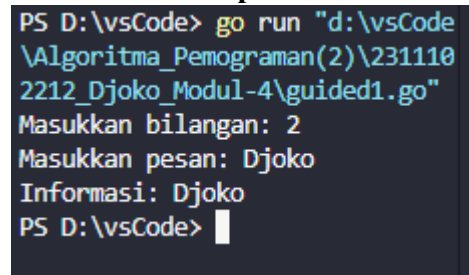
func main() {
    var bilangan int
    var pesan string

    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&bilangan)

    fmt.Print("Masukkan pesan: ")
    fmt.Scan(&pesan)

    cetakPesan(pesan, bilangan)
}
```

Screenshoot Output



```
PS D:\vsCode> go run "d:\vsCode
\Algoritma_Pemograman(2)\231110
2212_Djoko_Modul-4\guided1.go"
Masukkan bilangan: 2
Masukkan pesan: Djoko
Informasi: Djoko
PS D:\vsCode> |
```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan bilangan bulat dari 1 sampai 3 dan permintaan kedua adalah memasukkan kalimat. Program akan memanggil prosedur "cetakPesan" agar dapat menampilkan pesan yang dipanggil

2. Soal Studi Case

Buatlah sebuah program yang dapat mengirimkan pesan terkirim kepada email yang terdaftar dengan menggunakan prosedur dan juga array.

Sourcecode

```
package main

import (
    "fmt"
)

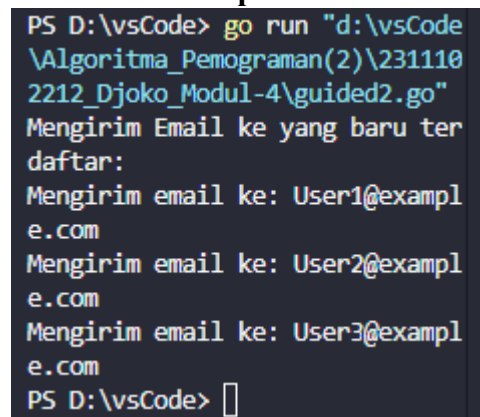
func sendEmailNotification(email string) {
    fmt.Printf("Mengirim email ke: %s\n", email)
}

func main() {
    emails := []string{"User1@example.com",
        "User2@example.com", "User3@example.com"}

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

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

Screenshoot Output



```
PS D:\vsCode> go run "d:\vsCode
\Algoritma_Pemograman(2)\231110
2212_Djoko_Modul-4\guided2.go"
Mengirim Email ke yang baru ter
daftar:
Mengirim email ke: User1@exampl
e.com
Mengirim email ke: User2@exampl
e.com
Mengirim email ke: User3@exampl
e.com
PS D:\vsCode> 
```

Deskripsi Program

Ini adalah sebuah program yang dapat mengirimkan pesan terkirim kepada email yang terdaftar, disini program memanggil prosedur menggunakan fungsi `forEach` untuk mengirim satu persatu email yang terdaftar didalam array ke prosedur "sendEmailNotification".

3. Soal Studi Case

Buatlah sebuah program yang dapat menghitung rumus ini $2 * x - 0.5 * y + 3.0$.

Sourcecode

```
package main

import (
    "fmt"
)

func f1(x, y int) float64 {
    return float64(2*x) - 0.5*float64(y) + 3.0
}

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.Println("Enter two integers:")
    fmt.Scan(&a, &b)

    f2(a, b, &c)
    fmt.Printf("Result from f2 (stored in c): %.2f\n", c)

    resultF1 := f1(b, a)
    fmt.Printf("Result from f1: %.2f\n", resultF1)
}
```

Screenshoot Output

```
PS D:\vsCode> go run "d:\vsCode
\Algoritma_Pemograman(2)\231110
2212_Djoko_Modul-4\guided3.go"
Enter two integers:
4 5
Result from f2 (stored in c): 8
.50
Result from f1: 11.00
PS D:\vsCode> |
```

Deskripsi Program

Program yang dirancang sesuai dengan soal yang diberikan. Program ini dapat menghitung nilai inputan pengguna menggunakan prosedur f1 dan f2 serta dapat ditampilkan di terminal.

III. UNGUIDED

1. Soal Studi Case

Buatlah program yang bisa untuk menghiung nilai permutasian dan kombinasi menggunakan prosedur.

Sourcecode

```
package main

import (
    "fmt"
)

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

func permutation(a, b int) int {
    kiri := factorial(a)
    kanan := factorial(a - b)
    return kiri / kanan
}

// Combination function
func combination(a, b int) int {
    kiri := factorial(a)
    kanan1 := factorial(b)
    kanan2 := factorial(a - b)
    return kiri / (kanan1 * kanan2)
}

func taskDiskrit(a, b, c, d int) {
    fmt.Printf("Permutasian dari (%d,%d): %d\n", a, c,
    permutation(a, c))
    fmt.Printf("Kombinasi dari (%d, %d): %d\n", a, c,
    combination(a, c))
    fmt.Printf("Permutasian dari (%d,%d): %d\n", b, d,
    permutation(b, d))
    fmt.Printf("Kombinasi dari (%d, %d): %d\n", b, d,
    combination(b, d))
}

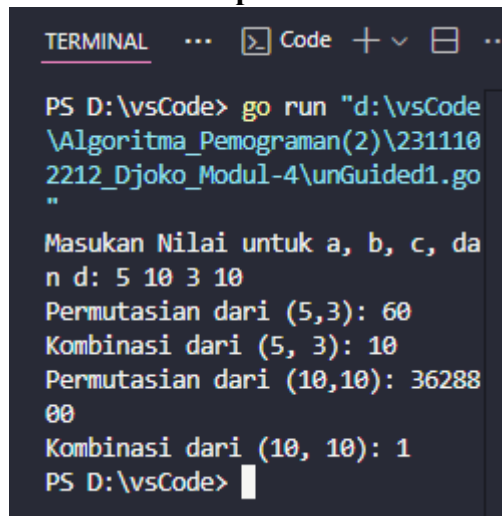
func main() {
    var a, b, c, d int
    fmt.Print("Masukan Nilai untuk a, b, c, dan d: ")
    fmt.Scan(&a, &b, &c, &d)
```

```

        if a >= c && b >= d {
            taskDiskrit(a, b, c, d)
        } else {
            fmt.Println("Invalid input: a harus >= c and b
harus >= d.")
        }
    }
}

```

Screenshoot Output



```

TERMINAL ... Code + v □ ...
PS D:\vsCode> go run "d:\vsCode
\Algoritma_Pemograman(2)\231110
2212_Djoko_Modul-4\unGuided1.go
"
Masukan Nilai untuk a, b, c, da
n d: 5 10 3 10
Permutasian dari (5,3): 60
Kombinasi dari (5, 3): 10
Permutasian dari (10,10): 36288
00
Kombinasi dari (10, 10): 1
PS D:\vsCode>

```

Deskripsi Program

Ini merupakan sebuah program yang dapat menghitung permutasian dan kombinasi, program meminta user untuk memasukan 4 bilangan bulat yang akan masuk kedalam variabel a,b,c dan d secara berurutan dan program akan melakukan pengkondisian dimana jika a kurang dari sama dengan c dan b kurang dari sama dengan d maka program akan memanggil prosedur “taskDiskrit”, tetapi ketika kondisi tidak terpenuhi maka akan terjadi pesan “invalid”.

Didalam taskDiskrit, prosedur ini juga memanggil prosedur lainnya seperti “permutation” dan “combination” yang dapat melakukan sebuah perhitungan sesuai dengan masing masing nama prosedur. Tidak lupa juga didalam prosedur permutasian dan kombinasi terdapat prosedur “factorial” untuk menghitung nilai faktorial

2. Soal Studi Case

Buatlah sebuah program yang dapat menampilkan nama pemenang beserta jumlah soal dan total skor dari sebuah kompetisi pemograman tingkat nasional yang memiliki 8 soal dan sebuah peraturan dimana jika

peserta tidak berhasil atau tidak mengirimkan jawaban maka akan dianggap menyelesaikannya dalam waktu 5 jam 1 menit (301).

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    // Kamus
    var nilai [2][8]int
    var nama1, nama2 string
    var i, j int

    // Algoritma
    for i = 0; i < 2; i++ {
        if i == 0 {
            fmt.Print("Masukkan nama peserta 1: ")
            fmt.Scan(&nama1)
        } else {
            fmt.Print("Masukkan nama peserta 2: ")
            fmt.Scan(&nama2)
        }

        for j = 0; j < 8; j++ {
            fmt.Printf("Masukkan nilai untuk peserta %d, soal ke-%d: ", i+1, j+1)
            fmt.Scan(&nilai[i][j])
            if nilai[i][j] == 0 {
                nilai[i][j] = 301
            }
        }
    }

    // Menghitung skor
    hitungSkor(nilai, 2, 8, nama1, nama2)
}

func hitungSkor(arr [2][8]int, jumlahOrang, banyakSkor int, peserta1, peserta2 string) {
    // Kamus
    var hasilSkor1, hasilSkor2 int
    var jumlahSoalTerjawab1, jumlahSoalTerjawab2 int

    // Algoritma
    for i := 0; i < jumlahOrang; i++ {
        for j := 0; j < banyakSkor; j++ {
            if i == 0 {
                hasilSkor1 += arr[i][j]
            }
        }
    }
}
```

```

        if arr[i][j] != 301 {
            jumlahSoalTerjawab1++
        }
    } else {
        hasilSkor2 += arr[i][j]
        if arr[i][j] != 301 {
            jumlahSoalTerjawab2++
        }
    }
}

// Menentukan pemenang
if jumlahSoalTerjawab1 == jumlahSoalTerjawab2 {
    if hasilSkor1 > hasilSkor2 {
        hasilSkor1 -= 301 * (banyakSkor -
jumlahSoalTerjawab1)
        fmt.Printf("Pemenang: %s, Jumlah soal
terjawab: %d, Total skor: %d\n", peserta1,
jumlahSoalTerjawab1, hasilSkor1)
    } else {
        hasilSkor2 -= 301 * (banyakSkor -
jumlahSoalTerjawab2)
        fmt.Printf("Pemenang: %s, Jumlah soal
terjawab: %d, Total skor: %d\n", peserta2,
jumlahSoalTerjawab2, hasilSkor2)
    }
} else if jumlahSoalTerjawab1 > jumlahSoalTerjawab2 {
    hasilSkor1 -= 301 * (banyakSkor -
jumlahSoalTerjawab1)
    fmt.Printf("Pemenang: %s, Jumlah soal terjawab:
%d, Total skor: %d\n", peserta1, jumlahSoalTerjawab1,
hasilSkor1)
} else {
    hasilSkor2 -= 301 * (banyakSkor -
jumlahSoalTerjawab2)
    fmt.Printf("Pemenang: %s, Jumlah soal terjawab:
%d, Total skor: %d\n", peserta2, jumlahSoalTerjawab2,
hasilSkor2)
}
}
}

```

Screenshoot Output

```
PS D:\vsCode> go run "d:\vsCode\Algoritma_Pemograman(2)\2311102212_Djoko_Modul-4\Unguided2.go"

Masukkan nama peserta 1: Astuti
Masukkan nilai untuk peserta 1, soal ke-1:
20
Masukkan nilai untuk peserta 1, soal ke-2: 25
Masukkan nilai untuk peserta 1, soal ke-3: 0
Masukkan nilai untuk peserta 1, soal ke-4: 0
Masukkan nilai untuk peserta 1, soal ke-5: 61
Masukkan nilai untuk peserta 1, soal ke-6: 71
Masukkan nilai untuk peserta 1, soal ke-7: 75
Masukkan nilai untuk peserta 1, soal ke-8: 10
Masukkan nama peserta 2: Bertha
Masukkan nilai untuk peserta 2, soal ke-1: 25
Masukkan nilai untuk peserta 2, soal ke-2: 47
Masukkan nilai untuk peserta 2, soal ke-3: 0
Masukkan nilai untuk peserta 2, soal ke-4: 26
Masukkan nilai untuk peserta 2, soal ke-5: 50
Masukkan nilai untuk peserta 2, soal ke-6: 60
Masukkan nilai untuk peserta 2, soal ke-7: 65
Masukkan nilai untuk peserta 2, soal ke-8: 21
Pemenang: Bertha, Jumlah soal terjawab: 7, Total skor: 294
PS D:\vsCode> 
```

Deskripsi Program

Program yang dirancang untuk menentukan pemenang dari sebuah kompetisi pemrograman dibuat menggunakan beberapa teknik, antara lain if-else (kombinasi), for (percabangan), dan prosedur.

Cara kerja dari program ini adalah program akan meminta pengguna untuk memasukan 8 catatan waktu untuk peserta pertama dan selanjutnya 8 catatan waktu juga untuk peserta kedua. Program memanggil prosedur "hitungSkor" untuk menghitung jumlah soal yang dikerjakan dan jumlah catatan waktu yang diperoleh oleh masing masing peserta lalu ditentukan pemenangnya berdasarkan jumlah soal, tetapi ketika jumlah soal yang dikerjakan sama maka akan ditentukan berdasarkan jumlah catatan waktunya.

3. Soal Studi Case

Buatlah sebuah program **skiena** 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 main() {
    var sukuAwal int

    fmt.Print("Masukkan nilai dari 0 - 1000000: ")
    fmt.Scan(&sukuAwal)
    cetakDeret(sukuAwal)
}

func cetakDeret(n int) {

    var genap, ganjil int

    fmt.Print(n)
    for {
        if n%2 == 0 {
            genap = n / 2
            n = genap
            fmt.Print(" ", n)
        } else {
            ganjil = 3*n + 1
            n = ganjil
            fmt.Print(" ", n)
        }
        if n == 1 {
            break
        }
    }
    fmt.Println()
}
```

Screenshoot Output

```
PS D:\vsCode> go run "d:\vsCode\Algoritma_Pemograman(2)\2311102212_Djoko_Modul-4\unGuided3.go"

Masukkan nilai dari 0 - 1000000: 22
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
PS D:\vsCode> |
```

Deskripsi Program

Program ini dirancang untuk menghitung perhitungan skiena yang dapat menghitung nilai yang diinputkan menjadi deret yang berakhir di nilai 1, ketika nilai yang diinputkan adalah genap maka nilai akan dibagi setengah dan akan dicetak ke deret selanjutnya, ketika nilai yang diinputkan adalah ganjil maka nilai selanjutnya adalah $3n + 1$.

Cara kerja dari program adalah program meminta pengguna untuk memasukan nilai dari 1 sampai 1000000, lalu program akan memanggil prosedur "cetakDeret" yang mana didalam prosedur terdapat rumus yang sudah disiapkan dan kondisi kondisi yang diperlukan.