

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**Muhammad Djoko Susilo / 2311102212**

**11-06**

**Dosen Pengampu :**

**ABEDNEGO DWI SEPTIADI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. DASAR TEORI

### Dasar Teori

#### Tipe Data dan Variabel

Variabel adalah tempat menyimpan nilai dalam bahasa pemrograman. Tipe data adalah tipe yang digunakan untuk menentukan tipe apa yang ingin disimpan dalam variabel, ada integer, real, boolean, karakter dan string.

Operasi yang dapat dilakukan terhadap tipe data variabel ada bermacam macam, contohnya seperti operasi aritmatika, seperti tambah "+" yang dapat digunakan untuk tipe data string, integer dan real.

Bahasa Go memiliki suatu tipe data yang bersifat konstanta atau nilai yang tidak dapat diubah(konstan), berfungsi agar memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut. Seperti PI untuk mempresentasikan konstanta  $\pi$

Didalam bahasa Go, untuk ada berbagai perintah contohnya adalah print untuk memunculkan nilai, scan untuk menerima nilai. Dalam print pun ada bermacam macam perintah yang memiliki fungsinya tersendiri seperti Printf, Println, dan print, juga ada beberapa perintah scan seperti ScanF, Scanln dan scan.

#### Struktur Kontrol Perulangan

Dalam bahasa Go **for** menjadi kata kunci untuk semua jenis perulangan yang dapat dipelajari dalam notasi algoritma. Dua bentuk yang digunakan disini adalah struktur **while-loop** dan **repeat-until**.

Dalam struktur perulangan biasanya diberikan batas atas dan batas bawah agar bisa membuat perulangan berhenti, juga bisa diberhentikan dengan menggunakan **break**.

#### Bentuk While-Loop

Struktur ini memungkinkan perulangan untuk memenuhi kondisi yang harus terpenuhi terlebih dahulu, dan akan selesai jika kondisi bernilai **false**/salah.

#### Bentuk Repeat-Until

Struktur ini memiliki sifat yang berbeda yaitu program akan dijalankan terlebih dahulu lalu dicek apakah kondisi bernilai **true** atau **false**, struktur ini memungkinkan program untuk menjalankan program setidaknya satu kali.

## II. GUIDED

### III. Soal 1

#### Soal Studi Case

Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silahkan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

#### Sourcecode

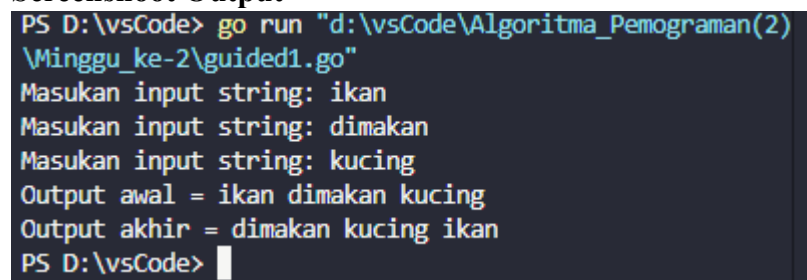
```
package main
import "fmt"

func main() {
    var (
        satu, dua, tiga string
        temp string
    )

    fmt.Print("Masukan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&tiga)

    fmt.Println("Output awal = " + satu + " " + dua + " "
+ tiga)
    temp = satu
    satu = dua
    dua = tiga
    tiga = temp
    fmt.Println("Output akhir = " + satu + " " + dua + " "
+ tiga)
}
```

#### Screenshoot Output



```
PS D:\vsCode> go run "d:\vsCode\Algoritma_Pemograman(2)
\Minggu_ke-2\guided1.go"
Masukan input string: ikan
Masukan input string: dimakan
Masukan input string: kucing
Output awal = ikan dimakan kucing
Output akhir = dimakan kucing ikan
PS D:\vsCode> 
```

#### Deskripsi Program

Program diatas adalah suatu program yang bisa mengubah nilai dari yang urutan pertama menjadi terakhir dengan menggunakan variabel temp sebagai tempat sementara untuk menyimpan nilai yang nanti akan diambil kembali suatu waktu.

Cara kerja program adalah program meminta pengguna untuk memasukan/menginputkan string, permintaan di minta sebanyak tiga kali dan permintaan pertama akan disimpan ke dalam variabel satu dan begitu seterusnya sampai tiga kali. Setelah itu program akan memunculkan output nilai yang sudah diinputkan dan ditampilkan ke terminal/layar, program mulai untuk menukar posisi inputan dan terakhir program akan memunculkan output berupa nilai yang sudah ditukar.

#### IV. Soal 2

##### Soal Studi Case

Tahun kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat(true) atau bukan (false).

##### Sourcecode

```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Masukkan tahun: ")
    fmt.Scanln(&tahun)

    if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {
        fmt.Printf("%d adalah tahun kabisat: true\n",
tahun)
    } else {
        fmt.Printf("%d bukan tahun kabisat: false\n",
tahun)
    }
}
```

### Screenshoot Output

```
PS D:\vsCode> go run "d:\vsCode\Algoritma_Pemograman(2)\Minggu_ke-2\guided2.go"
Masukkan tahun: 2000
2000 adalah tahun kabisat: true
PS D:\vsCode> go run "d:\vsCode\Algoritma_Pemograman(2)\Minggu_ke-2\guided2.go"
Masukkan tahun: 2018
2018 bukan tahun kabisat: false
PS D:\vsCode> |
```

### Deskripsi Program

Program diatas berfungsi untuk menghitung apakah tahun yang dimasukan/diinputkan merupakan tahun kabisat, tahun kabisat bisa diperoleh jika tahun yang diinputkan merupakan tahun yang habis dibagi 400 dan jika tahun habis dibagi 4 tetapi tidak habis dibagi 100, jika kedua kondisi tersebut terpenuhi maka program akan mencetak jika tahun tersebut adalah true dan jika tidak akan mencetak false.

## V. Soal 3

### Soal Studi Case

Buat program **Bola** yang menerima input jari-jari suatu bola (bilangan bulat). Tampilkan Volume dan Luas kulit bola.  $Volume_{bola} = \frac{4}{3}\pi r^3$  dan  $luas_{bangunan} = \pi r^2$  ( $\pi \approx 3.1415926535$ )

### Sourcecode

```
package main

import "fmt"

func main() {
    var r float64
    const pi = 3.1415926535

    fmt.Print("Masukkan jari-jari bola: ")
    fmt.Scanln(&r)

    r2 := r * r
    r3 := r * r * r

    volume := (4.0 / 3.0) * pi * r3

    luas := 4 * pi * r2
```

```
        fmt.Printf("Bola dengan jejari 5 memiliki volume %.2f  
        dan luas kulit %.2f", volume, luas)  
    }
```

### Screenshoot Output

```
PS D:\vsCode> go run "d:\vsCode\Algoritma_Pemograman(2)  
Minggu_ke-2\guided3.go"  
Masukkan jari-jari bola: 5  
Bola dengan jejari 5 memiliki volume 523.60 dan luas ku  
lit 314.16  
PS D:\vsCode>
```

### Deskripsi Program

Program diatas dibuat untuk menghitung volume dan luas kulit bola dengan menginputkan jari jarinya. , Rumus yang digunakan adalah  $Volume_{bola} = \frac{4}{3}\pi r^3$  dan  $luas_{bangunan} = \pi r^2$  ( $\pi \approx 3.1415926535$ ). Hasil dari perhitungan volume dan luas bola dicetak dengan format dua angka desimal.

## VI. UNGUIDED

### VII. Soal 1

### VIII. Soal Studi Case

- Diberikan nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam Fahrenheit, Reamur, dan Kelvin menggunakan rumus berikut:

#### 1. Fahrenheit:

$$\text{Fahrenheit} = \left( \text{Celcius} * \frac{9}{5} \right) + 32$$

#### 2. Reamur:

$$\text{Reamur} = \text{Celcius} * \frac{4}{5}$$

#### 3. Kelvin:

$$\text{Kelvin} = \text{Celcius} + 273.15$$

#### Sourcecode

```
package main

import "fmt"

func main() {
    var celsius float64

    // Input temperatur dalam derajat Celsius
    fmt.Print("Masukkan temperatur dalam derajat Celsius: ")
    fmt.Scan(&celsius)

    // Menghitung Fahrenheit, Reamur, dan Kelvin
    fahrenheit := (celsius * 9 / 5) + 32
    reamur := celsius * 4 / 5
    kelvin := celsius + 273.15

    // Menampilkan hasil konversi
    fmt.Printf("Temperatur Celsius: %.2f\n", celsius)
    fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)
    fmt.Printf("Derajat Reamur: %.2f\n", reamur)
    fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)
}
```

#### Screenshoot Output



```
Masukkan temperatur dalam derajat Celsius: 50
Temperatur Celsius: 50.00
Derajat Fahrenheit: 122.00
Derajat Reamur: 40.00
Derajat Kelvin: 323.15
|
```

### Deskripsi Program

Program digunakan untuk membaca input integer dan karakter yang nanti integer akan diubah kedalam karakter ASCII dan karakter akan diubah menjadi huruf setelah karakter tersebut.

### Soal 2

#### Soal Studi Case

Tipe karakter sebenarnya hanya apa yang tampak dalam tampilan. Di dalamnya tersimpan dalam bentuk biner 8 bit (byte) atau 32 bit (rune) saja.

Buat program ASCII yang akan membaca 5 buah data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII).

**Masukan** terdiri dari dua baris. Baris pertama berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

**Keluaran** juga terdiri dari dua baris. Baris pertama berisi 5 buah representasi karakter dari data yang diberikan, yang berdampingan satu dengan yang lain, tanpa dipisahkan spasi. Baris kedua berisi 3 buah karakter (juga tidak dipisahkan oleh spasi).

### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var a, b, c, d, e int
    var chars string
```

```

    fmt.Print("Masukkan 5 data integer (32 - 127): ")
    fmt.Scanf("%d %d %d %d %d", &a, &b, &c, &d, &e)

    fmt.Printf("Output karakter: %c%c%c%c%c\n", a, b, c,
d, e)

    fmt.Print("Masukkan 3 karakter berdampingan: ")
    fmt.Scanf("%s", &chars)

    nextChars := string(chars[0]+1) + string(chars[1]+1) +
string(chars[2]+1)
    fmt.Printf("Karakter setelah: %s\n", nextChars)
}

```

### Screenshoot Output

```

go run 7tmp/ISHyJ2mSTb.go
Masukkan 5 data integer (32 - 127): 66 97 105 117 115
Output karakter: Baius
Masukkan 3 karakter berdampingan: SNO
Karakter setelah: TOP

```

### Deskripsi Program

Program digunakan untuk membaca input integer dan karakter yang nanti integer akan diubah kedalam karakter ASCII dan karakter akan diubah menjadi huruf setelah karakter tersebut.

### Soal 3

#### Soal Studi Case

Buatkan sebuah program yang menerima input berupa warna dari ke-4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan **true** apabila urutannya sesuai ("Merah, Kuning, Hijau dan Ungu") dan **false** apabila urutannya salah.

#### Sourcecode

```

package main

import "fmt"

func main() {
    var colors [5][4]string
    var allTrue = true

    for i := 0; i < 5; i++ {
        fmt.Printf("Masukan Warnanya %d:\n", i+1)
        for j := 0; j < 4; j++ {

```

```

        fmt.Scan(&colors[i][j])
    }
}

for i := 0; i < 5; i++ {
    if colors[i][0] != "Merah" || colors[i][1] !=
    "Kuning" || colors[i][2] != "Hijau" || colors[i][3] !=
    "Ungu" {
        allTrue = false
        break
    }
}

if allTrue {
    fmt.Println("Berhasil: true")
} else {
    fmt.Println("Berhasil: false")
}
}

```

### Screenshoot Output

```

Ungu
Masukan Warnanya 2:
Merah
Kuning
Hijau
Ungu
Masukan Warnanya 3:
Merah
Kuning
Hijau
Ungu
Masukan Warnanya 4:
Merah
Kuning
Hijau
Ungu
Masukan Warnanya 5:
Merah
Kuning
Hijau
Ungu
Berhasil: true

```

```
Hijau
Ungu
Masukan Warnanya 2:
Merah
Kuning
Hijau
Ungu
Masukan Warnanya 3:
Merah
Kuning
Hijau
Ungu
Masukan Warnanya 4:
Ungu
Hijau
Kuning
Merah
Masukan Warnanya 5:
Merah
Kuning
Hijau
Ungu
Berhasil: false
|
```

### Deskripsi Program

Program digunakan mengecek apakah pengguna sudah melakukan uji coba warna yang sesuai dengan ketentuannya yaitu semua percobaan (5 percobaan) sudah sesuai dengan aturannya (harus berurut mulai dari ("Merah, Kuning, Hijau, Ungu")).

Program ini menggunakan perulangan untuk mengulang perintah agar pengguna dapat memasukkan hasil dari percobaan dan dapat di cek, percabangan digunakan untuk mengecek apakah pengguna sudah benar dalam percobaan jika benar maka akan menghasilkan nilai akhir true jika gagal akan false.

### Soal 4

#### Soal Studi Case

Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program akan meminta input berupa nama

bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita

#### Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var N int
    var bunga, pita string
    var count int

    // Meminta input jumlah bunga
    fmt.Print("Masukkan jumlah bunga (N): ")
    fmt.Scan(&N)

    // Proses input nama bunga sebanyak N kali atau hingga 'SELESAI'
    for i := 1; i <= N; i++ {
        fmt.Printf("Bunga %d: ", i)
        fmt.Scan(&bunga)

        // Jika input 'SELESAI', hentikan input
        if strings.ToUpper(bunga) == "SELESAI" {
            break
        }

        // Gabungkan nama bunga ke pita
        if pita == "" {
            pita = bunga
        } else {
            pita = pita + " - " + bunga
        }
        count++
    }

    // Menampilkan hasil pita dan jumlah bunga yang dimasukkan
    fmt.Printf("Pita: %s\n", pita)
    fmt.Printf("Banyak bunga: %d\n", count)
}
```

#### Screenshoot Output

```
Masukkan jumlah bunga (N): 3
Bunga 1: Mawar
Bunga 2: Selesai
Pita: Mawar
Banyak bunga: 1
```

### Deskripsi Program

Program ini meminta input berupa bilangan bulat positif, lalu program akan melakukan pengulangan sebanyak yang diminta oleh user, namun jika user mengetikkan "Selesai" sebelum mencapai yang diminta user maka akan langsung berhenti pengulangannya. Hasil akhirnya adalah setiap nama bunga yang diinputkan oleh user akan tampil di output jika terdapat lebih dari satu maka akan ada tanda "-" sebagai pemisah, dan terdapat juga jumlah bunga yang diinputkan oleh user.

### Soal 5

#### Soal Studi Case

Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri dan kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong tidak lebih dari 9 kg.

Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi lebih dari 9 kg atau lebih.

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKiri, beratKanan float64

    for {
        // Meminta input berat barang di kantong kiri
        dan kanan
        fmt.Print("Masukkan berat belanjaan di kantong
        kiri: ")
```

```

        fmt.Scan(&beratKiri)
        fmt.Print("Masukkan berat belanjaan di kantong
kanan: ")
        fmt.Scan(&beratKanan)

        // Menghitung selisih berat antara kedua kantong
        selisihBerat := math.Abs(beratKiri - beratKanan)

        // Tentukan apakah sepeda motor oleng (true)
        atau tidak oleng (false)
        oleng := selisihBerat > 9

        // Tampilkan hasil pengecekan
        if oleng {
            fmt.Println("Sepeda motor Pak Andi akan
oleng: true")
        } else {
            fmt.Println("Sepeda motor Pak Andi tidak
akan oleng: false")
        }

        totalBarang := math.Abs(beratKiri + beratKanan)

        if totalBarang > 150 {
            fmt.Println("Program Selesai")
            break
        }
    }
}

```

### Screenshoot Output

```

Masukkan berat belanjaan di kantong kiri: 5
Masukkan berat belanjaan di kantong kanan: 10
Sepeda motor Pak Andi tidak akan oleng: false
Masukkan berat belanjaan di kantong kiri: 55.6
Masukkan berat belanjaan di kantong kanan: 70.2
Sepeda motor Pak Andi akan oleng: true
Masukkan berat belanjaan di kantong kiri: 59
Masukkan berat belanjaan di kantong kanan: 98
Sepeda motor Pak Andi akan oleng: true
Program Selesai
|

```

### Deskripsi Program

Program ini telah dimodifikasi untuk menggunakan variabel boolean `oleng`, yang diisi dengan nilai `true` jika selisih berat antara kedua kantong lebih dari 9 kg, dan `false` jika tidak. Penggunaan boolean ini membuat logika lebih jelas: program langsung mengatur status `oleng` berdasarkan hasil perhitungan.

Jika `oleng == true`, program akan menampilkan bahwa sepeda motor akan `oleng`, dan jika `oleng == false`, program akan memberi tahu bahwa sepeda motor tidak akan `oleng`.

Program akan selesai jika suatu inputan memiliki berat lebih dari 150.

## Soal 6

### Soal Studi Case

Diberikan sebuah persamaan sebagai berikut:

$$f(K)=f(K) = \frac{4K+2}{(4K+1)(4K+3)}$$

1. Buatlah sebuah program yang menerima input sebuah bilangan sebagai ( $K$ ), kemudian menghitung dan menampilkan nilai  $f(K)$  sesuai persamaan di atas.
2. Modifikasi program sebelumnya yang menerima input integer  $K$  dan menghitung aproksimasi  $\sqrt{2}$  untuk  $K$  tersebut. Aproksimasi  $\sqrt{2}$  diberikan oleh rumus:

$$\sqrt{2} \approx \frac{1}{K} \left( \frac{4K + 2}{(4K + 1)(4K + 3)} \right)$$

Tampilkan hasil perhitungan  $\sqrt{2}$  dengan presisi 10 angka di belakang koma.

### Sourcecode

```
package main

import (
    "fmt"
)

// Fungsi untuk menghitung aproksimasi akar 2 menggunakan
rumus produk
func hitungAkar2(n int) float64 {
    hasil := 1.0
```



```

        // Loop untuk menghitung produk dari suku-suku
        for k := 0; k < n; k++ {
            atas := (4*float64(k) + 2) * (4*float64(k) + 2)
            bawah := (4*float64(k) + 1) * (4*float64(k) + 3)
            hasil *= atas / bawah
        }

        return hasil
    }

    func main() {
        var n int

        // Meminta input jumlah suku yang akan dihitung
        fmt.Print("Nilai K: ")
        fmt.Scan(&n)

        // Menghitung aproksimasi akar 2
        akar2 := hitungAkar2(n)

        // Menampilkan hasil aproksimasi
        fmt.Printf("Nilai akar 2 = %.10f\n", akar2)
    }

```

### Screenshoot Output

```

Nilai K: 10
Nilai akar 2 = 1.4054086752

```

### Deskripsi Program

Program pertama menerima input nilai  $K$ , kemudian menghitung nilai  $f(K)$  menggunakan persamaan yang diberikan dan menampilkan hasilnya dengan presisi 10 angka di belakang koma. Program kemudian menghitung aproksimasi nilai  $\sqrt{2}$  menggunakan rumus yang diberikan dan menampilkan hasilnya dengan tingkat ketelitian yang sama.