

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMOGRAMAN II**  
**MODUL II**  
**REVIEW STRUKTUR KONTROL**



Oleh:

NAMA : DWI HESTI ARIANI

NIM : 2311102094

KELAS : 11- IF -02

**S1 TEKNIK INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## I. DASAR TEORI

### a. Golang

Golang adalah Bahasa pemrograman procedural. Bahasa ini dikembangkan pada tahun 2007 oleh Robert Griesemer, Rob Pike , dan Ken Thompson di Google, tetapi diluncurkan pada tahun 2009 sebagai Bahasa pemrograman sumber terbuka. Program disusun dengan menggunakan paket, untuk manajemen dependensi yang efisien. Bahasa ini juga mendukung pola adopsi lingkungan yang mirip dengan bahasa dinamis. Misalnya, inferensi tipe (`y := 0` adalah deklarasi valid dari variabel `y` bertipe `int`).

Dalam Bahasa pemrograman Go, program utama selalu mempunyai dua komponen yaitu :

- **Package main** merupakan penanda bahwa file ini berisi program utama
- **Func main()** berisi kode utama dari sebuah program Go.

Beberapa utilitas Go yang terkait dalam proses bahasa Go yaitu sebagai berikut:

- **go build** : mengkompilasi program, sumber daya yang ada dalam folder menjadi sebuah program.
- **go build file.go** : mengkompilasi program sumber file.go saja.
- **go fmt** : membaca semua program sumber dalam folder dan mereformat penulisannya agar sesuai dengan standar penulisan program sumber Go.
- **go clean** : membersihkan file-file dalam folder sehingga tersisa program sumber nya saja. \

### b. Jenis Tipe Data

- Integer
- Boolean
- Real
- String
- Character

## II. GUIDED

### Source Code

```
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)
}
// Dwi Hesti Ariani_2311102094
```

### Output Program

```
go run /tmp/A4skbwJ19G.go
Masukan input string: satu
Masukan input string: dua
Masukan input string: tiga
Output awal = satu dua tiga
Output akhir = dua tiga satu
```

## Deskripsi Program

Program diatas ialah program Go sederhana untuk menukar nilai dari tiga input string yang akan menyimpan nilai input. User diminta untuk memasukkan tiga string , nilai input akan disinpan di 'satu' 'dua' 'tiga' lalu program akan menukar nilai dari 'satu' 'dua' dan 'tiga' menggunakan variable sementara yaitu " temp" . " temp " mengambil nilai dari 'satu' kemudian 'satu' mengambil nilai dari ' dua' dst. Setelah proses menukar nilai selesai program akan mencetak output akhir yaitu berupa hasil proses menukar nilai.

## Guided 2

### Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var r float64

    fmt.Print("Jejari = ")
    fmt.Scan(&r)

    volumeBola := (4.0 / 3.0) * math.Pi * math.Pow(r, 3)
    luasBola := 4 * math.Pi * math.Pow(r, 2)

    fmt.Printf("Bola dengan jejari %.0f memiliki volume %.4f dan\nluas kulit %.4f\n", r, volumeBola, luasBola)
}
```

## Output Program

```
go run /tmp/KuC1VxfFdY.go
Jejari = 21
Bola dengan jejari 21 memiliki volume 38792.3861 dan luas kulit 5541
.7694
```

## Deskripsi Program

Program diatas ialah program yang digunakan untuk menghitung volume dan luas kulit bola berdasarkan jejari yang diinput oleh pengguna, kemudian , program menghitung nilai volume dan luas kulit bola menggunakan rumus yang tertera pada program , lalu setelah menghitung program akan menampilkan output yang berisi satu baris nilai volume dan luas , masing-masing dengan empat digit di belakang koma.

## Guided 3

### Source Code

```
package main

import (
    "fmt"
)

func main() {
    var celsius float64

    fmt.Print("Temperatur Celsius: ")
    fmt.Scan(&celsius)

    fahrenheit := (celsius * 9 / 5) + 32

    reamur := celsius * 4 / 5

    kelvin := celsius + 273.15

    fmt.Printf("Derajat Reamur: %.2f\n", reamur)
    fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)
    fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)
}
```



### Output Program

```
go run /tmp/A3zhqJUeZf.go
Temperatur Celsius: 25
Derajat Reamur: 20.00
Derajat Fahrenheit: 77.00
Derajat Kelvin: 298.15
```

### Deskripsi Program

Program diatas ialah sebuah program Go yang digunakan untuk mengkonversi suhu dari Celcius ke beberapa satuan suhu lainnya, yaitu Fahrenheit, Reamur, dan Kelvin. Program ini meminta user untuk input suhu dalam Celcius , kemudian program menghitung nilai suhu dalam satuan lainnya menggunakan rumus konversi pada program diatas. Lalu setelah menghitung nilai suhu dalam satuan lainnya , program akan menampilkan hasilnya yang berisi nilai suhu Fahrenheit, Reamur dan Kelvin.

### III. UNGUIDED

- Latihan Modul 2 B

#### Source Code

##### Unguided 1

```
package main

import (
    "fmt"
)

func main() {
    var warna1, warna2, warna3, warna4 string
    var berhasil bool = true

    fmt.Println("Masukkan warna percobaan (merah/kuning/hijau/ungu):")
    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        fmt.Scanln(&warna1, &warna2, &warna3, &warna4)

        if warna1 != "merah" || warna2 != "kuning" || warna3 != "hijau" || warna4 != "ungu" {
            berhasil = false
        }
    }

    fmt.Println("BERHASIL:", berhasil)
}

\\ Dwi Hesti Ariani_2311102094
```

#### Output Program

```
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true

=== Session Ended. Please Run the code again ===
```

## Deskripsi Program

Program diatas ialah program yang meminta user untuk memasukkan kombinasi warna percobaan sebanyak 5 kali kemudian memeriksa apakah kombinasi warna yang di masukkan oleh user benar atau tidak. Perulangan for digunakan untuk melakukan percobaan sebanyak 5 kali. Variabel Berhasil (Boolean) digunakan untuk menyimpan nilai true dan false .Program akan melakukan pemeriksaan apakah kombinasi yang diinputkan oleh user sudah benar atau tidak. Jika semua kombinasi sudah sesuai program akan mencetak output true , namun jika salah satu warna tidak sesuai program akan mencetak output false.

## Unguided 2

### Source Code

```
package main

import "fmt"

func main() {
    var pita string
    var jumlahBunga int

    fmt.Println("Masukkan bunga yang anda inginkan")
    for {
        var namaBunga string
        fmt.Print("Nama bunga ke-", jumlahBunga+1, ": ")
        fmt.Scanln(&namaBunga)

        if namaBunga == "selesai" { // input selesai jika anda
            ingin menghentikan
            break
        }

        if pita != "" {
            pita += "-"
        }
        pita += namaBunga
        jumlahBunga++
    }

    fmt.Println("\nIsi pita:", pita)
```



```
    fmt.Println("Jumlah bunga:", jumlahBunga)
}
\\ Dwi Hesti Ariani_2311102094
```

### Output Program

```
go run /tmp/bC1kFJz/8U.go
Masukkan bunga yang anda inginkan
Nama bunga ke-1: lili
Nama bunga ke-2: tulip
Nama bunga ke-3: anggrek
Nama bunga ke-4: mawar
Nama bunga ke-5: kamboja
Nama bunga ke-6: selesai

Isi pita: lili-tulip-anggrek-mawar-kamboja
Jumlah bunga: 5
```

### Deskripsi Program

Program diatas ialah program yang meminta user untuk memasukkan daftar nama bunga lalu menjumlahkan hasil inputan nama bunga yang di inputkan oleh user dan menyimpannya dalam variable string pita yang dipisahkan oleh tanda penghubung.

Perulangan for digunakan untuk menginputkan nama bunga dan akan berhenti bila user menginputkan “selesai “ menggunakan if . if pita != “ “ digunakan untuk memeriksa apakah string pita sudah tidak kosong , jika sudah tidak kosong program akan menambahkan tanda penghubung sebelum menambahkan nama bunga baru ke string pita. Kemudian program akan mencetak output yaitu berupa isi pita dan jumlah bunga.

### Unguided 3

#### Source Code

```
package main

import "fmt"

func main() {
    var kantongKiri, kantongKanan float64

    for {
        fmt.Print("Masukkan berat kantong kiri: ")
```

```

    fmt.Scanln(&kantongKiri)

    fmt.Print("Masukkan berat kantong kanan: ")
    fmt.Scanln(&kantongKanan)

    if kantongKiri < 0 || kantongKanan < 0 {
        fmt.Println("Berat kantong tidak boleh negatif!")
        break
    }

    totalBerat := kantongKiri + kantongKanan
    if totalBerat > 150 {
        fmt.Println("Total berat melebihi 150 kg!")
        break
    }

    selisih := kantongKiri - kantongKanan
    if selisih < 0 {
        selisih = -selisih
    }

    fmt.Println("Sepeda motor pak Andi akan oleng :", selisih
    >= 9)
    }
}

```

## Output Program

```

go run /tmp/3vdLL4aju8.go
Masukkan berat kantong kiri: 5
Masukkan berat kantong kanan: 10
Sepeda motor pak Andi akan oleng : false
Masukkan berat kantong kiri: 55
Masukkan berat kantong kanan: 60
Sepeda motor pak Andi akan oleng : false
Masukkan berat kantong kiri: 55.5
Masukkan berat kantong kanan: 75.7
Sepeda motor pak Andi akan oleng : true
Masukkan berat kantong kiri: 100
Masukkan berat kantong kanan: 80
Total berat melebihi 150 kg!

```

## Deskripsi Program

Program diatas ialah sebuah program yang digunakan untuk menentukan apakah sepeda motor Pak Andi akan oleng berdasarkan berat kantong kiri dan kanan yang diinputkan oleh user. Kemudian, program melakukan beberapa pengecekan sebagai berikut:

- Jika berat kantong kiri atau kanan negatif, program akan menampilkan pesan error "Berat kantong tidak boleh negatif!" dan menghentikan proses.
- Jika total berat kantong kiri dan kanan melebihi 150 kg, program akan menampilkan pesan error "Total berat melebihi 150 kg!" dan menghentikan proses.
- Jika total berat kantong kiri dan kanan tidak melebihi 150 kg, program akan menghitung selisih berat antara kantong kiri dan kanan. Jika selisih berat kurang dari 0, program akan mengubah nilai selisih menjadi positif.
- Akhirnya, program akan menampilkan hasil apakah sepeda motor Pak Andi akan oleng atau tidak berdasarkan nilai selisih berat. Jika selisih berat lebih besar atau sama dengan 9, maka sepeda motor Pak Andi akan oleng.

## Unguided 4

### Source Code

```
package main

import (
    "fmt"
    "math"
)

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

    result := calculateSqrt2(k)
    fmt.Printf("Akar 2 untuk k = %d adalah %.10f\n", k, result)
}

func calculateSqrt2(k int) float64 {
```

```

    product := 1.0
    for i := 0; i <= k; i++ {
        numerator := math.Pow(float64(4*i+2), 2)
        denominator := float64((4*i + 1) * (4*i + 3))
        product *= numerator / denominator
    }
    return product
}
// Dwi Hesti Ariani_2311102094

```

### Output Program

```

Masukkan nilai k: 100
Akar 2 untuk k = 100 adalah 1.4133387072

```

### Deskripsi Program

Program diatas ialah program yang digunakan untuk menghitung akar 2 menggunakan rumus infinite series untuk k nilai yang di input oleh user. Kemudian, program memanggil fungsi **calculateSqrt2(k)** untuk menghitung akar 2 menggunakan rumus infinite series. Output program ini akan menampilkan hasil akar 2 untuk k nilai yang diinput oleh user dengan 10 digit di belakang koma.

- Latihan Modul 2 C

### Unguided 5

#### Source Code

```

package main

import "fmt"

func main() {
    var beratParsel int
    fmt.Print("Masukkan berat parsel (dalam gram): ")
    fmt.Scanln(&beratParsel)

    totalBeratKg := beratParsel / 1000
    sisaBeratGram := beratParsel % 1000
}

```

```

biayaPerKg := 10000
biayaTotalKg := totalBeratKg * biayaPerKg

if sisaBeratGram >= 500 {
    biayaSisaGram := sisaBeratGram * 5
    biayaTotal := biayaTotalKg + biayaSisaGram
    fmt.Println("Biaya total:", biayaTotal)
} else if sisaBeratGram < 500 {
    if totalBeratKg > 10 {
        biayaTotal := biayaTotalKg
        fmt.Println("Biaya total:", biayaTotal)
    } else {
        biayaSisaGram := sisaBeratGram * 15
        biayaTotal := biayaTotalKg + biayaSisaGram
        fmt.Println("Biaya total:", biayaTotal)
    }
}
}

```

## Output Program

```

Masukkan berat parcel (dalam gram): 11750
Biaya total: 113750

```

```

Masukkan berat parcel (dalam gram): 8500
Biaya total: 82500

```

```

Masukkan berat parcel (dalam gram): 9250
Biaya total: 93750

```

## Deskripsi Program

Program di atas adalah sebuah program Go yang digunakan untuk menghitung biaya pengiriman parcel berdasarkan berat parcel yang diberikan oleh pengguna. Kemudian, program menghitung total berat parcel dalam kilogram dan sisa berat dalam gram menggunakan operasi pembagian dan modulus.

Program juga menghitung biaya total pengiriman parcel berdasarkan total berat parcel dalam kilogram dan sisa berat dalam gram. Biaya per

kilogram adalah Rp 10.000,00. Jika sisa berat dalam gram lebih besar atau sama dengan 500 gram, maka biaya sisa berat dalam gram adalah Rp 5 per gram. Jika total berat parsel lebih besar dari 10 kilogram, maka biaya total adalah biaya total kilogram saja. Jika tidak, maka biaya total adalah biaya total kilogram ditambah biaya sisa berat dalam gram.

## Unguided 6

- a. Jika nilai `nam` ialah 80.1, maka program akan mencetak output “ A”. Karena hal ini sudah sesuai dengan spesifikasi soal yaitu nilai 80.1 diatas dari nilai 80 dan dibawah nilai 100 .
- b. Kesalahan dalam program :
  - Penggunaan beberapa kondisi “ if “ yang tidak benar dimana hal ini menyebabkan program tidak dapat menentukan hasil nilai akhir mata kuliah yang tepat.
  - Variabel “ `nmk` ” dideklarasikan tetapi tidak gunakan dalam program.
  - Kondisi ‘else if ‘ yang kurang : penggunaan ‘ if ‘ berulang kali tanpa penggunaan kondisi ‘ else if ‘ akan membuat program mengecek semua kondisi ‘ if ‘ secara berurutan , meskipun salah satu kondisi sebelumnya telah terpenuhi. Hal ini berbeda dengan penggunaan ‘ else if ‘ Dimana kondisi selanjutnya hanya akan dievaluasi jika kondisi sebelumnya tidak terpenuhi sehingga menggunakan ‘ else if ‘ lebih efisien.

### Alur Program

1. Mendklarasikan variable ‘ `nam` ‘ sebagai ‘float64’ untuk menyimpan nilai akhir mata kuliah.
2. Mendklarasikan variable ‘ `nmk` ‘ sebagai ‘string’ untuk menyimpan nilai akhir mata kuliah dalam bentuk huruf.
3. Gunakan kondisi ‘ if – else ‘ untuk menentukan nilai akhir mata kuliah berdasarkan nilai akhir yang dimasukkan.

### c . Source Code

```
package main  
  
import "fmt"
```

```

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah : ")
    fmt.Scanln(&nam)
    if nam >= 80 {
        nmk = "A"
    } else if nam >= 72.5 {
        nmk = "AB"
    } else if nam >= 65 {
        nmk = "B"
    } else if nam >= 57.5 {
        nmk = "BC"
    } else if nam >= 50 {
        nmk = "C"
    } else if nam >= 40 {
        nmk = "D"
    } else {
        nmk = "E"
    }
    fmt.Println("Nilai mata kuliah : ", nmk)
}

// Dwi Hesti Ariani_2311102094

```

### Output program

```

Nilai akhir mata kuliah : 93.5
Nilai mata kuliah : A

Nilai akhir mata kuliah : 70.6
Nilai mata kuliah : B

Nilai akhir mata kuliah : 49.5
Nilai mata kuliah : D

```

## Unguided 7

### Source code

```
package main

import "fmt"

func main() {
    var bilangan int

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

    fmt.Println("Bilangan:", bilangan)
    fmt.Print("Faktor: ")
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            fmt.Print(i, " ")
        }
    }
    fmt.Println()

    prima := true
    if bilangan <= 1 {
        prima = false
    } else {
        for i := 2; i*i <= bilangan; i++ {
            if bilangan%i == 0 {
                prima = false
                break
            }
        }
    }

    if prima {
        fmt.Println("Prima: true")
    } else {
        fmt.Println("Prima: false")
    }
}

// Dwi Hesti Ariani_2311102094
```



## Output program

```
go run /tmp/Z9jnP86AH7.go
Masukkan bilangan: 7
Bilangan: 7
Faktor: 1 7
Prima: true
```

## Deskripsi Program

Program diatas ialah program yang digunakan untuk menentukan faktor-faktor dari sebuah bilangan dan menentukan apakah bilangan tersebut prima atau tidak. Kemudian, program menampilkan bilangan yang dimasukkan dan faktor-faktor dari bilangan tersebut menggunakan loop for yang memeriksa apakah bilangan dapat dibagi habis oleh angka-angka dari 1 sampai bilangan itu sendiri.

Program juga menentukan apakah bilangan tersebut prima atau tidak. Bilangan prima adalah bilangan yang hanya dapat dibagi habis oleh 1 dan bilangan itu sendiri. Program menggunakan loop for yang memeriksa apakah bilangan dapat dibagi habis oleh angka-angka dari 2 sampai akar kuadrat dari bilangan tersebut. Jika bilangan dapat dibagi habis oleh salah satu angka tersebut, maka bilangan tidak prima.