

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

**Aryo Tegar Sukarno / 2311102018
11 – IF – 6**

Dosen Pengampu :

Abednego Dwi Septiadi S. Kom. M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

1. Pengenalan Go Language

Golang adalah bahasa pemrograman sumber terbuka (opensource) yang berguna untuk memudahkan pembuatan perangkat lunak yang sederhana, andal, dan efisien. Golang (Go) disusun pada bulan September 2007 oleh Robert Griesemer, Rob Pike, dan Ken Thompson, semua kegiatannya dilakukan di Google, dan diumumkan pada bulan November 2009. Golang (Go) memiliki tujuan yaitu efisien dalam kompilasi dan eksekusi, dan efektif dalam menulis program. Golang (Go) memiliki kemiripan dengan bahasa pemrograman C dalam bentuk penulisan dan pengkodeannya. Golang (Go) adalah proyek sumber terbuka (opensource), jadi kode sumber untuk kompiler, pustaka (library), dan alatnya tersedia secara gratis untuk siapa saja. Golang (Go) dapat berjalan pada Unix-like, Sistem Linux, FreeBSD, OpenBSD, Mac OS X, Plan 9, dan Microsoft Windows. Mengapa harus mempelajari Golang (Go), Saat ini Golang (Go) merupakan bahasa pemrograman yang digemari oleh para programmer di dunia karena mudah untuk dipelajari serta memiliki kecepatan yang berada diatas bahasa pemrograman lain.

2. Karakteristik Go Language

Go memiliki beberapa karakteristik utama yang membuatnya unik dan efektif dalam berbagai skenario pemrograman, di antaranya:

- **Tipe Statis dan Dikompilasi:** Go merupakan bahasa yang diketik secara statis (statically typed), artinya tipe data variabel ditentukan pada saat kompilasi. Ini memungkinkan deteksi kesalahan lebih awal dan meningkatkan performa karena kode dioptimalkan pada waktu kompilasi.
- **Bahasa Pemrograman Sistem:** Go dirancang untuk menjadi bahasa pemrograman sistem, dengan kemampuan yang mirip dengan C dan C++, seperti kontrol atas memori dan kemampuan menjalankan program dengan kinerja tinggi.
- **Concurrency Model (Goroutines):** Salah satu fitur terkuat dari Go adalah dukungannya terhadap *concurrency* melalui Goroutines. Goroutines adalah unit-unit kecil dari *thread* yang lebih ringan dan mudah dijalankan secara bersamaan, memungkinkan pemrogram menulis program yang dapat memanfaatkan CPU multi-core dengan mudah.

- **Garbage Collection:** Go memiliki sistem *garbage collection* yang otomatis membersihkan memori yang tidak lagi digunakan, memudahkan manajemen memori bagi pengembang tanpa perlu khawatir mengalokasikan dan melepaskan memori secara manual.
- **Cross-Platform:** Kode Go dapat dikompilasi dan dijalankan di berbagai platform, seperti Windows, macOS, dan Linux, tanpa perlu melakukan modifikasi pada kode sumber.

II. GUIDED

Soal Studi Case

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

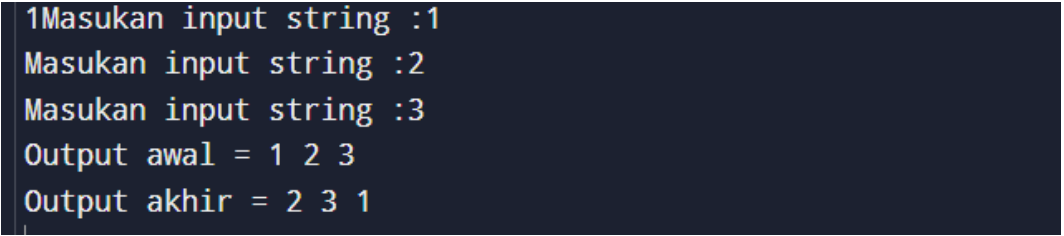
Sourcecode

```
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



```
1Masukan input string :1
Masukan input string :2
Masukan input string :3
Output awal = 1 2 3
Output akhir = 2 3 1
```

Deskripsi Program

Program diatas adalah bahasa pemrograman GO yang memprogram sebuah pertukaran huruf atau angka yang menukar secara horizontal

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

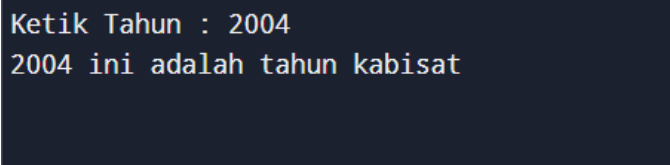
```
package main

import "fmt"

func kabisat(tahun int) bool {
    if tahun % 400 == 0 {
        return true
    } else if tahun % 100 == 0 {
        return false
    } else if tahun % 4 == 0 {
        return true
    } else {
        return false
    }
}

func main() {
    var tahun int
    fmt.Print("Ketik Tahun : ")
    fmt.Scanln(&tahun)
    if kabisat(tahun) {
        fmt.Println(tahun, "ini adalah tahun kabisat")
    } else {
        fmt.Println(tahun, "ini bukan tahun kabisat")
    }
}
```

Screenshoot Output



```
Ketik Tahun : 2004
2004 ini adalah tahun kabisat
```

Deskripsi Program

Dari deskripsi program tersebut adalah menandakan sebuah tahun kabisat atau bukan seperti contoh memasukan angka yang dapat dibagi 4 atau tidak apa bila tidak di bagi 4 maka bukan tahun kabisat. Ini menggunakan inkrement **if else** sebagai bentuk perjalanan dari sebuah pemrograman ini

Soal Studi Case

Buatlah sebuah program yang dapat menghitung volume dan luas permukaan bola. Program ini akan meminta pengguna untuk memasukkan nilai jari-jari bola sebagai input.

Rumus yang digunakan:

Volume bola: $V = (4/3) * \pi * r^3$

Luas permukaan bola: $L = 4 * \pi * r^2$

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jariJari float64
    fmt.Print("Masukan jari jari bola : ")
    fmt.Scanln(&jariJari)

    volume := (4.0/3.0) * math.Pi * math.Pow(jariJari,
3)
    luasPermukaan := 4 * math.Pi * math.Pow(jariJari, 2)

    fmt.Printf("Bola dengan jari-jari %.2f memiliki
volume %.5f dan luas permukaan %.5f\n", jariJari,
volume, luasPermukaan)
}
```

Screenshoot Output

```
Masukan jari jari bola : 5
Bola dengan jari-jari 5.00 memiliki volume 523.59878 dan luas permukaan 314
.15927
```

Deskripsi Program

Program ini menampilkan sebuah pemrograman penghitungan jari jari pada bola yang menggunakan **float** dan **math** sebagai penghitungan matematika

III. UNGUIDED

4A. Soal Studi Case

Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperature tersebut dalam Fahrenheit.

$$\text{Celsius} = (\text{Fahrenheit} - 32) \times 5/9$$

$$\text{Reamur} = \text{Celsius} \times 4/5$$

$$\text{Kelvin} = (\text{Fahrenheit} + 459.67) \times 5/9$$

Soal:

Buatlah sebuah program yang dapat mengkonversi suhu dari Celsius ke Fahrenheit. Gunakan rumus konversi yang telah diberikan di atas.

Sourcecode

```
package main

import "fmt"

func main(){
    var celcius float64

    fmt.Print("temperatur Celcius: ")
    fmt.Scan(&celcius)

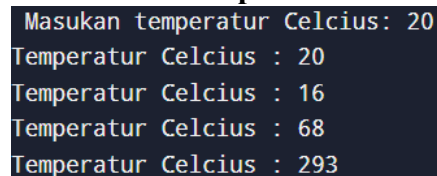
    fahrenheit :=(celcius * 9/5) + 32

    reamur := celcius * 4/5

    kelvin := celcius + 273.15

    fmt.Printf("Temperatur Celcius : %.0f\n",celcius)
    fmt.Printf("Temperatur Celcius : %.0f\n",reamur)
    fmt.Printf("Temperatur Celcius : %.0f\n",fahrenheit)
    fmt.Printf("Temperatur Celcius : %.0f\n",kelvin)
}
```

Screenshoot Output



```
Masukan temperatur Celcius: 20
Temperatur Celcius : 20
Temperatur Celcius : 16
Temperatur Celcius : 68
Temperatur Celcius : 293
```

Deskripsi Program

Program diatas menggunakan **float** sebagai pondasi utama untuk penghitungan dengan rumus yang telah tersedia di rumus celcius

5 A. Soal Studi Case

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).

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var a, b, c, d, e int
    var char1, char2, char3 rune

    fmt.Println("Masukkan 5 angka integer :")
    fmt.Scanf("%d %d %d %d %d", &a, &b, &c, &d, &e)

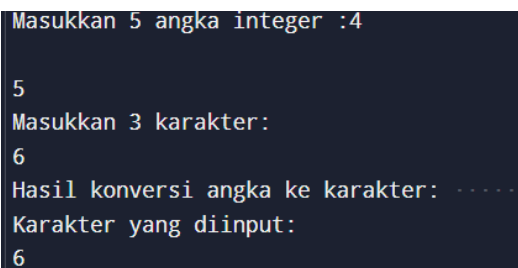
    fmt.Scanf("\n")

    fmt.Println("Masukkan 3 karakter:")
    fmt.Scanf("%c%c%c", &char1, &char2, &char3)

    fmt.Printf("Hasil konversi angka ke karakter:
%c%c%c%c%c\n", a, b, c, d, e)

    fmt.Printf("Karakter yang diinput: %c%c%c\n", char1, char2,
char3)
}
```

Screenshoot Output

A screenshot of a terminal window showing the output of the Go program. The text is as follows:

```
Masukkan 5 angka integer :4
5
Masukkan 3 karakter:
6
Hasil konversi angka ke karakter: .....
Karakter yang diinput:
6
```


Deskripsi Program

Program ini adalah program yang menginputkan nomor berdasarkan inputan melalui nomor yang di inputkan oleh user dan hasil dari output tersebut disusun menjadi angka kode ASCII.

1 B. Soal Studi Case

Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berurutan adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang.

Buatlah sebuah program yang menerima input berupa warna dari ke-4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan true apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan **warna lainnya**.

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var warna [5][4]string
    urutanBenar := [4]string{"merah", "kuning", "hijau", "ungu"}
    berhasil := true

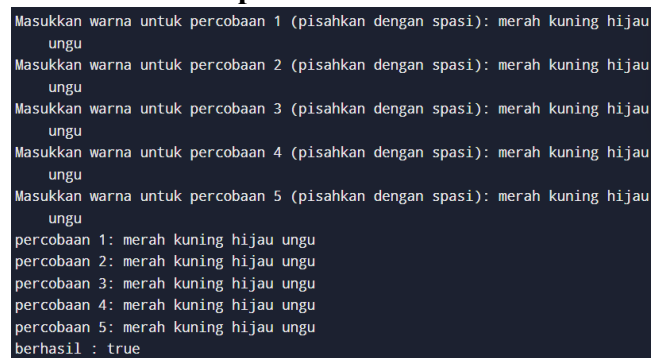
    for i := 0; i < 5; i++ {
        fmt.Printf("Masukkan warna untuk percobaan %d (pisahkan dengan spasi): ", i+1)
        fmt.Scan(&warna[i][0], &warna[i][1], &warna[i][2], &warna[i][3])
    }

    for i := 0; i < 5; i++ {
        fmt.Printf("percobaan %d: %s %s %s %s\n", i+1, warna[i][0], warna[i][1], warna[i][2], warna[i][3])
        if warna[i] != urutanBenar {
```

```
        berhasil = false
    }
}

fmt.Printf("berhasil : %t\n", berhasil)
}
```

Screenshoot Output



```
Masukkan warna untuk percobaan 1 (pisahkan dengan spasi): merah kuning hijau ungu
Masukkan warna untuk percobaan 2 (pisahkan dengan spasi): merah kuning hijau ungu
Masukkan warna untuk percobaan 3 (pisahkan dengan spasi): merah kuning hijau ungu
Masukkan warna untuk percobaan 4 (pisahkan dengan spasi): merah kuning hijau ungu
Masukkan warna untuk percobaan 5 (pisahkan dengan spasi): merah kuning hijau ungu
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
|
```

Deskripsi Program

Program ini adalah program perulangan user diminta menginputkan nama sesuai dengan warna urutan yang sama dan outputnya akan menghasilkan kodingan yang sama.

2 B . Soal Studi Case

Sourcecode

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.

(Petunjuk: gunakan operasi penggabungan string dengan operator "+")

Tampilkan isi pita setelah proses input selesai.

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var N int
    fmt.Print("Masukkan jumlah bunga (N): ")
    fmt.Scan(&N)

    var pita strings.Builder
    var bunga string
    count := 0

    for i := 1; i <= N; i++ {
        fmt.Printf("Bunga %d: ", i)
        fmt.Scan(&bunga)

        if strings.ToUpper(bunga) == "SELESAI" {
            break
        }

        if pita.Len() > 0 {
            pita.WriteString(" - ")
        }
        pita.WriteString(bunga)
        count++
    }

    fmt.Printf("Pita: %s\n", pita.String())
    fmt.Printf("Bunga: %d\n", count)
}
```

--

Screenshoot Output

```
Masukkan jumlah bunga (N): 1
Bunga 1: 3
Pita: 3
Jumlah Bunga: 1
```

Deskripsi Program

Program tersebut adalah program berbasis konsol yang meminta pengguna untuk memasukkan sejumlah nama bunga dan menyimpannya dalam sebuah pita yang dipisahkan oleh tanda " - ".

3 B . 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-kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong sisi 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 9 kg atau lebih.

Sourcecode

```
package main

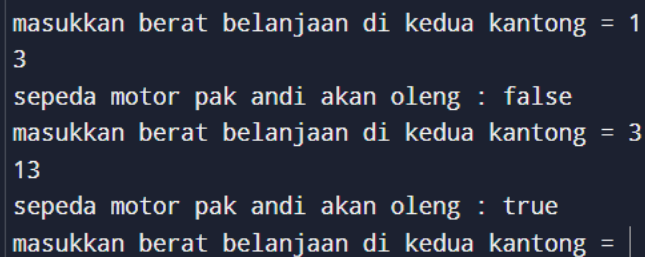
import (
    "fmt"
)

func main() {
    var berat1, berat2 float64

    for {
        fmt.Print("masukkan berat belanjaan di kedua kantong = ")
        fmt.Scan(&berat1, &berat2)

        if berat1 >= 9 || berat2 >= 9 {
            fmt.Println("sepeda motor pak andi akan oleng : true")
        } else {
            fmt.Println("sepeda motor pak andi akan oleng : false")
        }
    }
}
```

Screenshoot Output



```
masukkan berat belanjaan di kedua kantong = 1
3
sepeda motor pak andi akan oleng : false
masukkan berat belanjaan di kedua kantong = 3
13
sepeda motor pak andi akan oleng : true
masukkan berat belanjaan di kedua kantong = |
```

Deskripsi Program

Program tersebut adalah program yang menggunakan float sebagai indicator **if else** untuk mengetahui bahwa jumlah tersebut **berat** atau **tidak** menggunakan **true** or **false**

4 B . Soal Studi Case

Buatlah sebuah program yang menerima input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai $f(K)$ sesuai persamaan di atas.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func hitungF(k int) float64 {
    pembilang := math.Pow(float64(4*k+2), 2)
    penyebut := float64((4*k+1)*(4*k+3))
    return pembilang / penyebut
}

func hitungAkarDua(k int) float64 {
    hasil := 1.0
    for i := 0; i <= k; i++ {
        hasil *= hitungF(i)
    }
    return hasil
}

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

    nilaiF := hitungF(k)
    fmt.Printf("Nilai f(%d) = %.10f\n", k, nilaiF)

    akarDua := hitungAkarDua(k)
    fmt.Printf("Nilai akar 2 untuk K = %d: %.10f\n", k, akarDua)
}
```

Screenshoot Output

```
go run /tmp/71xYac1n6G.go
Masukkan nilai K: 8
Nilai  $f(8) = 1.0008658009$ 
Nilai akar 2 untuk  $K = 8: 1.4044354005$ 
```

Deskripsi Program

Program ini mungkin melakukan perhitungan numerik atau komputasi matematika untuk suatu nilai input KKK dan menampilkan hasil dari dua operasi berbeda. Jika dilihat dari hasil kedua, besar kemungkinan fungsi yang digunakan berk