

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL 2
REVIEW STRUKTUR KONTROL**



Oleh:

NAMA : AJI TRI PRASETYO

NIM : 2311102064

KELAS : IF 11 02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

2024

I. DASAR TEORI

Struktur Kontrol dalam Golang

Struktur kontrol dalam pemrograman, termasuk Golang, adalah mekanisme yang digunakan untuk mengontrol alur eksekusi program. Dengan struktur kontrol, kita dapat membuat keputusan, mengulang blok kode, dan mengatur urutan eksekusi program berdasarkan kondisi tertentu.

Golang menyediakan beberapa jenis struktur kontrol yang umum digunakan, yaitu:

1. Percabangan (Selection)

- **if-else:** Digunakan untuk membuat keputusan berdasarkan kondisi tertentu. Jika kondisi bernilai benar (true), maka blok kode di dalam if akan dieksekusi. Jika salah (false), maka blok kode di dalam else (jika ada) akan dieksekusi.

```
if kondisi {  
    // Kode yang akan dijalankan jika kondisi benar  
}  
else {  
    // Kode yang akan dijalankan jika kondisi salah  
}
```

switch: Digunakan untuk mengevaluasi sebuah ekspresi dan menjalankan blok kode yang sesuai dengan nilai ekspresi tersebut.

```
switch ekspresi {  
case nilai1:  
    // Kode jika ekspresi sama dengan nilai1  
case nilai2:  
    // Kode jika ekspresi sama dengan nilai2  
default:  
    // Kode jika ekspresi tidak sama dengan nilai1 atau nilai2  
}
```

Perulangan (Iteration)

- **for:** Golang hanya memiliki satu jenis perulangan, yaitu for. Perulangan for sangat fleksibel dan dapat digunakan untuk berbagai macam kasus.

// Perulangan dengan kondisi

```
for kondisi {
```

```
    // Kode yang akan diulang selama kondisi benar
```

```
}
```

// Perulangan dengan inisialisasi, kondisi, dan post-statement

```
for i := 0; i < 10; i++ {
```

```
    // Kode yang akan diulang 10 kali
```

```
}
```

// Perulangan tanpa kondisi (infinite loop)

```
for {
```

```
    // Kode yang akan diulang terus-menerus
```

```
    // Biasanya digunakan dengan break untuk menghentikan perulangan
```

```
}
```

II. GUIDED

Source Code + Screenshot hasil program beserta penjelasan

NO 1.

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)
```

}

```
Masukan input string: Telkom
Masukan input string: University
Masukan input string: Purwokerto
Output awal = Telkom University Purwokerto
Output akhir = University Purwokerto Telkom
```

Penjelasan: berdasarkan output diatas program ini ada tiga input yang dibaca oleh pengguna, dan ditampilkan urutan awalnya, lalu dari tiga input tersebut ditukar urutannya secara acak, jadi variabel “temp” ditukar dengan variabel “satu” variabel “satu” ditukar dengan variabel “dua” variabel “tiga” ditukar dengan variabel “temp” jadi yang urutan awal a, b, c ditukar menjadi b, c, a. dan setelah itu ditampilkan Kembali

NO 2.

Source Code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var tahun int
```

```
    fmt.Print("Tahun : ")
```

```
    fmt.Scan(&tahun)
```

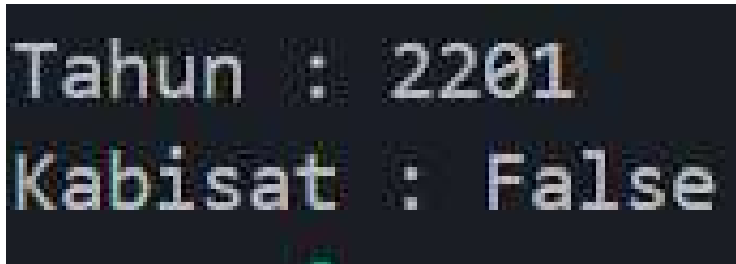
```
    if tahun%4 == 0 && tahun%100 != 0 {
```

```
        fmt.Print("Kabisat : True")
```

```

    } else {
        fmt.Print("Kabisat : False")
    }
}

```



```

Tahun : 2201
Kabisat : False

```

Penjelasan :

Kode Go ini merupakan contoh sederhana dari penggunaan struktur kontrol if-else untuk membuat keputusan dalam pemrograman. Kode ini juga menunjukkan bagaimana cara membaca input dari pengguna dan menampilkan output ke layar.

NO 3.

```

package main

```

```

import (
    "fmt"
    "math"
)

```

```

func main() {
    var jejari, volume, luas_kulit float64
    fmt.Print("Jejari = ")
}

```

```

    fmt.Scan(&jejari)

    volume = math.Pi * 4.0 / 3.0 * math.Pow(jejari, 3)

    luas_kulit = math.Pi * 4 * math.Pow(jejari, 2)

    fmt.Printf("Bola dengan jejari %v memiliki volume %.4f dan luas
    kulit %.4f", jejari, volume, luas_kulit)
}

```

```

Jejari = 43
Bola dengan jejari 43 memiliki volume 333038.1428 dan luas kulit 23235.2193

```

Penjelasan : Kode Go ini memberikan contoh tentang bagaimana menggunakan bahasa Go untuk melakukan perhitungan matematika dan berinteraksi dengan pengguna. Kode ini menunjukkan penggunaan fungsi-fungsi dari paket `fmt` dan `math` untuk input, output, dan perhitungan matematika. Dengan pemahaman yang baik tentang kode ini, Anda dapat mengembangkan aplikasi Go yang lebih kompleks untuk berbagai keperluan.

III. UNGUIDED

NO. 4

Source Code :

```

package main

import "fmt"

func main() {
    var celcius, reamur, fahrenheit, kelvin float64

    fmt.Print("Temperatur Celcius : ")

    fmt.Scan(&celcius)
}

```

$\text{fahren} = \text{celcius} * 9.0 / 5.0 + 32$

$\text{reamur} = \text{celcius} * 4.0 / 5.0$

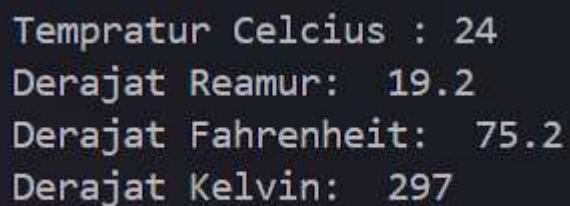
$\text{kelvin} = 5.0 / 9.0 * (\text{fahren} + 459.67)$

`fmt.Println("Derajat Reamur: ", reamur)`

`fmt.Println("Derajat Fahrenheit: ", fahren)`

`fmt.Println("Derajat Kelvin: ", int(kelvin))`

`}`



```
Tempratur Celcius : 24
Derajat Reamur: 19.2
Derajat Fahrenheit: 75.2
Derajat Kelvin: 297
```

Penjelasan : Mendeklarasikan variabel bertipe float untuk mewakili suhu.

- . Membaca input suhu dalam derajat Celcius dari pengguna.
- . Melakukan konversi suhu dari Celcius ke Reamur, Fahrenheit, dan Kelvin menggunakan rumus-rumus yang sesuai.
- . Mencetak hasil konversi ke layar.

Konversi suhu Kelvin ke bilangan bulat mungkin tidak selalu tepat, terutama untuk nilai suhu yang sangat kecil. Jika presisi desimal sangat penting, sebaiknya tidak melakukan konversi ke integer.

Ada banyak satuan suhu lainnya selain yang disebutkan di sini. Kode ini dapat diperluas untuk mendukung konversi ke satuan suhu lainnya.

NO. 5

Source Code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var angka1, angka2, angka3, angka4, angka5 int
```

```
    var char1, char2, char3 byte
```

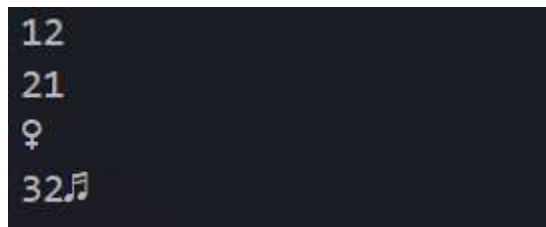
```
    fmt.Scanln(&angka1, &angka2, &angka3, &angka4, &angka5)
```

```
    fmt.Scanf("%c%c%c", &char1, &char2, &char3)
```

```
    fmt.Printf("%c%c%c%c%c\n", angka1, angka2, angka3, angka4,  
angka5)
```

```
    fmt.Printf("%c%c%c\n", char1+1, char2+1, char3+1)
```

```
}
```



Penjelasan :

Kode ini seperti sebuah program kecil yang meminta kita memasukkan beberapa angka dan huruf. Setelah itu, program akan menampilkan angka-angka tersebut sebagai huruf dan juga akan mengubah huruf-huruf yang kita masukkan menjadi huruf berikutnya dalam abjad

B. MODUL 2B

NO 1.

Source Code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var warnaInput [4]string
```

```
    warnaDiharapkan := [4]string{"merah", "kuning", "hijau", "ungu"}
```

```
    var adaPerbedaan bool = false
```

```
    for percobaan := 1; percobaan <= 5; percobaan++ {
```

```
        fmt.Print("Percobaan ", percobaan, " : ")
```

```
        fmt.Scan(&warnaInput[0], &warnaInput[1], &warnaInput[2],  
&warnaInput[3])
```

```
        for j := 0; j < 4; j++ {
```

```
            if (warnaInput[j] != warnaDiharapkan[j]) && !adaPerbedaan {
```

```
                adaPerbedaan = true
```

```
            }
```

```
        }
```

```

    }

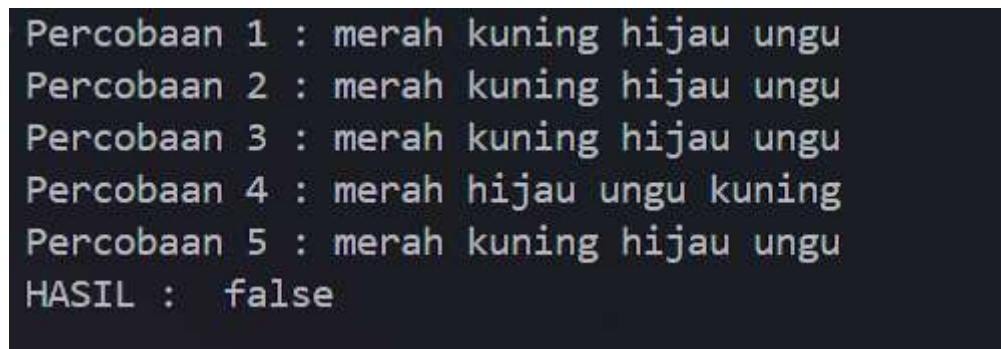
    fmt.Println("HASIL : ", !adaPerbedaan)
}

```

Penjelasn :

Kode ini dirancang untuk membandingkan dua daftar warna (dalam bentuk array string) dan menentukan apakah ada perbedaan di antara keduanya. Proses perbandingan ini dilakukan sebanyak 5 kali percobaan.

Screenshoot



```

Percobaan 1 : merah kuning hijau ungu
Percobaan 2 : merah kuning hijau ungu
Percobaan 3 : merah kuning hijau ungu
Percobaan 4 : merah hijau ungu kuning
Percobaan 5 : merah kuning hijau ungu
HASIL : false

```

NO 2.

Source Code :

```

package main

import "fmt"

func main() {
    var jumlahBunga int

    var namaBunga, rangkaianBunga string

    fmt.Print("N : ")
}

```

```

    fmt.Scan(&jumlahBunga)

    for i := 0; i < jumlahBunga; i++ {

        fmt.Print("Bunga ke-", i+1, ": ")

        fmt.Scan(&namaBunga)

        rangkaianBunga += namaBunga + " - "

    }

    fmt.Print("Rangkaian: ", rangkaianBunga)

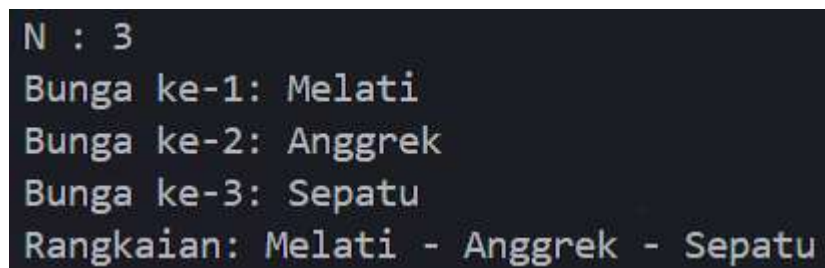
}

```

Penjelasan :

Kode Golang di atas dirancang untuk mengumpulkan nama-nama bunga dari pengguna dan kemudian menyusunnya menjadi sebuah rangkaian bunga.

Screenshoot



```

N : 3
Bunga ke-1: Melati
Bunga ke-2: Anggrek
Bunga ke-3: Sepatu
Rangkaian: Melati - Anggrek - Sepatu

```

NO 3.

Source Code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var beratKantongA, beratKantongB float64
```

```
    for {
```

```
        fmt.Print("Masukkan berat belanjaan kedua kantong: ")
```

```
        fmt.Scan(&beratKantongA, &beratKantongB)
```

```
        if beratKantongA >= 9 || beratKantongB >= 9 {
```

```
            fmt.Println("Proses selesai")
```

```
            break
```

```
        }
```

```
    }
```

```
}
```

Penjelasan :

Kode Golang di atas dirancang untuk terus-menerus meminta pengguna memasukkan berat dua kantong belanja hingga salah satu atau kedua kantong memiliki berat 9 kilogram atau lebih. Setelah kondisi tersebut terpenuhi, program akan berhenti.

Screenshoot

```
Masukkan berat belanjaan kedua kantong: 2.4 21
Proses selesai
```

NO 4.

Source Code :

```
package main
```

```
import (
```

```
    "fmt"
```

```
    "math"
```

```
)
```

```
func main() {
```

```
    var k int
```

```
    var fungsi float64
```

```
    fmt.Print("Nilai K = ")
```

```
    fmt.Scan(&k)
```

```
    pembilang := math.Pow(4*float64(k)+2, 2)
```

```
    penyebut := ((4.0*float64(k) + 1) * (4.0*float64(k) + 3))
```

```
    fungsi = pembilang / penyebut
```

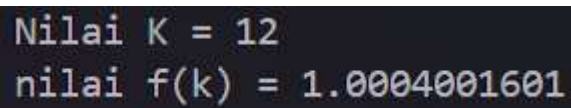
```
    fmt.Printf("nilai f(k) = %.10f", fungsi)
```

```
}
```

Penjelasan :

Kode Golang ini dirancang untuk menghitung nilai suatu fungsi matematika tertentu berdasarkan nilai k yang diinputkan oleh pengguna. Fungsi matematika yang dihitung adalah hasil pembagian antara kuadrat dari $4k+2$ dengan hasil kali $(4k+1)$ dan $(4k+3)$.

Screenshoot



```
Nilai K = 12
nilai f(k) = 1.0004001601
```

C. MODUL 2C.

NO 1.

Source Code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var gram, beratkg, beratGram, biayakg, biayagram int
```

```
    fmt.Print("Berat parsel (gram) : ")
```

```
    fmt.Scan(&gram)
```

```
    beratkg = gram / 1000
```

```
    beratGram = gram % 1000
```

```
    biayakg = beratkg * 10000
```

```

if beratGram >= 500 {
    biayagram = beratGram * 5
} else {
    biayagram = beratGram * 15
}

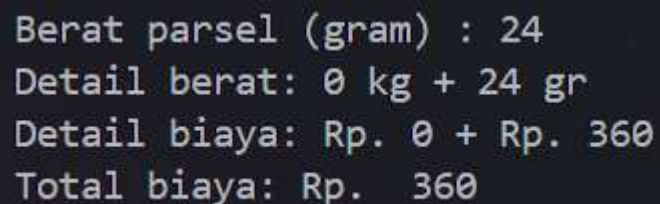
fmt.Printf("Detail berat: %v kg + %v gr\n", beratkg, beratGram)
fmt.Printf("Detail biaya: Rp. %v + Rp. %v \n", biyakg, biayagram)
fmt.Println("Total biaya: Rp. ", biayagram+biyakg)
}

```

Penjelasan :

Kode Go ini dirancang untuk menghitung biaya pengiriman suatu parcel berdasarkan beratnya. Program akan meminta pengguna untuk memasukkan berat parcel dalam gram, kemudian menghitung biaya pengiriman berdasarkan tarif yang ditentukan.

Screenshoot



```

Berat parcel (gram) : 24
Detail berat: 0 kg + 24 gr
Detail biaya: Rp. 0 + Rp. 360
Total biaya: Rp. 360

```


NO 2.

Source Code :

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var nam float64
```

```
    var nmk string
```

```
    fmt.Print("Nilai akhir mata kuliah: ")
```

```
    fmt.Scanln(&nam)
```

```
    switch {
```

```
    case nam > 80:
```

```
        nmk = "A"
```

```
    case nam > 72.5:
```

```
        nmk = "AB"
```

```
    case nam > 65:
```

```
        nmk = "B"
```

```
    case nam > 57.5:
```

```
        nmk = "BC"
```

```
    case nam > 50:
```

```
        nmk = "C"
```

```
    case nam > 40:
```

```

        nmk = "D"
    default:
        nmk = "E"
    }

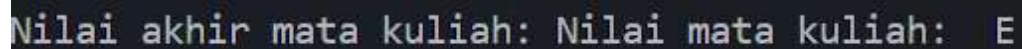
    fmt.Println("Nilai mata kuliah: ", nmk)
}

```

Penjelasan :

Kode ini dirancang untuk mengkonversi nilai akhir mata kuliah (dalam bentuk angka) menjadi nilai huruf (A, B, C, dan seterusnya). Ini sering digunakan dalam sistem penilaian akademik.

Screenshoot :



```

Nilai akhir mata kuliah: Nilai mata kuliah: E

```

NO 3.

Source Code :

```

package main

```

```

import "fmt"

```

```

func main() {

```

```

    var bilfaktor int

```

```

    fmt.Print("Bilangan: ")

```

```

    fmt.Scan(&bilfaktor)

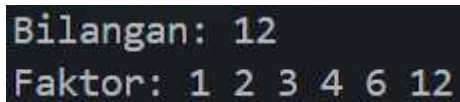
```

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

Penjelasan :

Kode ini dirancang untuk mencari dan mencetak faktor-faktor dari sebuah bilangan yang diinputkan oleh pengguna.

Screenshoot



```
Bilangan: 12  
Faktor: 1 2 3 4 6 12
```