

**LAPORAN PRAKTIKUM  
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL II  
REVIEW STRUKTUR KONTROL**



Oleh:

**RAFI UBAID PUTRA**

2311102244

IF-11-07

**S1 TEKNIK INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

**2024**

## **I. DASAR TEORI**

### Dasar Teori

Struktur kontrol dalam pemrograman adalah alat yang memungkinkan pengendalian alur eksekusi program berdasarkan kondisi tertentu. Struktur ini dibagi menjadi tiga kategori utama: bersyarat, perulangan, dan lompatan. Struktur kontrol bersyarat, seperti pernyataan if, if-else, dan switch, memungkinkan program untuk mengambil keputusan berdasarkan hasil evaluasi suatu kondisi. Sementara itu, struktur kontrol perulangan, seperti for, while, dan do-while, memungkinkan blok kode untuk dieksekusi berulang kali hingga kondisi tertentu tidak lagi terpenuhi. Penggunaan struktur kontrol ini memberikan fleksibilitas dalam menangani berbagai situasi dalam logika pemrograman.

Selain itu, struktur kontrol lompatan, yang mencakup pernyataan break, continue, dan return, mengubah alur eksekusi program secara langsung. break digunakan untuk keluar dari loop saat kondisi tertentu terpenuhi, sementara continue digunakan untuk melewati iterasi saat ini dan melanjutkan ke iterasi berikutnya. Pernyataan return digunakan untuk keluar dari fungsi dan mengembalikan nilai jika diperlukan. Memahami dan menguasai struktur kontrol sangat penting bagi programmer untuk menciptakan kode yang efisien, fleksibel, dan mudah dibaca, sehingga memungkinkan pengembangan aplikasi yang lebih baik dan adaptif.

## II. GUIDED

### 1. Source Code

**package main**

```
package main

import "fmt"

func main() {
    var nama string = "Rafi"
    var umur int = 20
    var tinggi float64 = 180.5
    var isSunny bool = false
    var inisial rune = 'R'
    fmt.Println("Nama:", nama)
    fmt.Println("Umur:", umur)
    fmt.Println("Tinggi:", tinggi)
    fmt.Println("is sunny", isSunny)
    fmt.Println("inisial", inisial)
}
```

## Screenshot hasil program

```
PS E:\ALPRO\PRATIUM ST2> go run "e:\ALPRO\PRATIUM ST2\Unguided\guided1.go"
Nama: Rafi
Umur: 20
Tinggi: 190.5
is sunny false
inisial 82
PS E:\ALPRO\PRATIUM ST2> |
```

## Penjelasan program

- `var nama string = "Rafi":`
  - Variabel nama bertipe string untuk menyimpan teks atau kata-kata.
  - Nilainya adalah "Rafi".
  - `var umur int = 20:`
  - Variabel umur bertipe int untuk menyimpan bilangan bulat (integer).
  - Nilainya adalah 20.
  - `var tinggi float64 = 180.5:`
  - Variabel tinggi bertipe float64, digunakan untuk menyimpan angka desimal.
  - Nilainya adalah 180.5.
  - `var isSunny bool = false:`
  - Variabel isSunny bertipe bool, digunakan untuk menyimpan nilai logika true atau false.
  - Nilainya adalah false (menandakan cuaca tidak cerah).
  - `var inisial rune = 'R':`
  - Variabel inisial bertipe rune. Tipe rune adalah alias untuk int32 yang digunakan untuk menyimpan karakter Unicode.
  - Nilainya adalah 'R'.
- 
- **Definisi Target:** Program mendefinisikan urutan warna target sebagai slice string `targetSequence`.
  - **Input Warna:** Program meminta input warna dari pengguna untuk setiap percobaan (5 percobaan, masing-masing 4 warna).
  - **Pemeriksaan Urutan:** Fungsi `checkSequence` digunakan untuk memeriksa apakah urutan warna yang diinputkan sesuai dengan urutan warna target.
  - **Hasil:** Program mencetak hasil dari setiap percobaan, apakah berhasil atau tidak, serta menampilkan semua hasil di akhir.

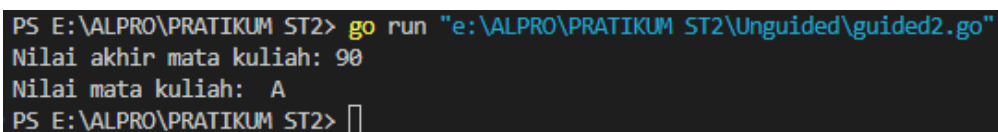
## 2. Source Code

```
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)
}
```

### Hasil Program



```
PS E:\ALPRO\PRATIUM ST2> go run "e:\ALPRO\PRATIUM ST2\Unguided\guided2.go"
Nilai akhir mata kuliah: 90
Nilai mata kuliah: A
PS E:\ALPRO\PRATIUM ST2> 
```

### Penjelasan Program

- Package dan Import: Program dimulai dengan mendeklarasikan package main dan mengimpor package fmt untuk input dan output.
- Fungsi main: Ini adalah fungsi utama yang dijalankan saat program dieksekusi.
- Deklarasi Variabel: Variabel nam dideklarasikan untuk menyimpan nilai akhir mata kuliah dalam tipe float64, dan nmk untuk menyimpan grade dalam tipe string.
- Input Pengguna: Program meminta pengguna untuk memasukkan nilai akhir mata kuliah.

- Logika Penilaian: Menggunakan serangkaian if dan else if, program menentukan grade (nmk) berdasarkan nilai nam yang dimasukkan.
- Output Hasil: Program mencetak grade yang diperoleh berdasarkan nilai akhir mata kuliah.

### 3. Source Code

```
package main

import "fmt"

func findFactors(b int) []int {
    var factors []int
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            factors = append(factors, i)
        }
    }
    return factors
}

func isPrime(b int) bool {
    factors := findFactors(b)
    return len(factors) == 2
}

func main() {
    var b int
    fmt.Print("Bilangan: ")
    fmt.Scan(&b)

    factors := findFactors(b)
    fmt.Print("Faktor: ")
    for _, factor := range factors {
        fmt.Printf("%d ", factor)
    }
    fmt.Println()

    primeStatus := isPrime(b)
    fmt.Printf("Prima: %t\n", primeStatus)
}
```

Hasil Program

```
PS E:\ALPRO\PRATIUM ST2> go run "e:\ALPRO\PRATIUM ST2\Unguided\guided3.go"
Bilangan: 10,2
Faktor: 1 2 5 10
Prima: false
PS E:\ALPRO\PRATIUM ST2> █
```

### Penjelasan Program

- **Fungsi findFactors:** Fungsi ini menerima sebuah bilangan bulat b sebagai argumen dan mengembalikan sebuah slice dari faktor-faktor bilangan tersebut. Di dalamnya, program menggunakan loop untuk memeriksa setiap angka dari 1 hingga b. Jika b habis dibagi oleh i, maka i dianggap sebagai faktor dan ditambahkan ke dalam slice factors.
- **Fungsi isPrime:** Fungsi ini juga menerima sebuah bilangan bulat b dan menggunakan fungsi findFactors untuk mendapatkan faktor-faktornya. Kemudian, fungsi ini mengecek apakah panjang slice factors sama dengan 2, yang berarti b adalah bilangan prima (hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri).
- **Fungsi main:** Di sini, program meminta input dari pengguna untuk bilangan bulat b. Setelah itu, program memanggil findFactors untuk mendapatkan faktor-faktor dari b dan mencetaknya. Kemudian, program memanggil isPrime untuk menentukan apakah b adalah bilangan prima, dan mencetak status tersebut sebagai true atau false.

### III. UNGUIDED

#### 1. Source Code

**package main**

```
import (  
    "fmt"  
)  
  
func main() {  
    targetSequence := []string{"merah", "kuning",  
"hijau", "ungu"}  
    var results []bool  
  
    for i := 1; i <= 5; i++ {  
        fmt.Printf("Percobaan %d:\n", i)  
        var colors [4]string  
  
        for j := 0; j < 4; j++ {  
            fmt.Printf("%d: ", j+1)  
            fmt.Scanln(&colors[j])  
        }  
  
        if checkSequence(colors[:],  
targetSequence) {  
            results = append(results, true)  
            fmt.Println("Percobaan BERHASIL :  
true")  
        }  
    }  
}
```



```

        } else {
            results = append(results, false)
            fmt.Println("Percobaan BERHASIL :
false")
        }
    }

    fmt.Println("\nHasil semua percobaan:")
    for i, result := range results {
        fmt.Printf("Percobaan %d: %t\n", i+1,
result)
    }
}

func checkSequence(colors []string, target
[]string) bool {
    for i := 0; i < len(target); i++ {
        if colors[i] != target[i] {
            return false
        }
    }

    return true
}

```

## Screenshot hasil program

```
DEBUG CONSOLE  TERMINAL  PORTS
Percobaan BERHASIL : false
Percobaan 4:
1: merah
2: hijau
3: ungu
4: pink
Percobaan BERHASIL : false
Percobaan 5:
1: merah
2: kuning
3: hijau
4: ungu
Percobaan BERHASIL : true

Hasil semua percobaan:
Percobaan 1: false
Percobaan 2: false
Percobaan 3: false
Percobaan 4: false
Percobaan 5: true
PS E:\ALPRO\PRATIKUM ST2> 
```

## Penjelasan program

- **Definisi Target:** Program mendefinisikan urutan warna target sebagai slice string targetSequence.
- **Input Warna:** Program meminta input warna dari pengguna untuk setiap percobaan (5 percobaan, masing-masing 4 warna).
- **Pemeriksaan Urutan:** Fungsi checkSequence digunakan untuk memeriksa apakah urutan warna yang diinputkan sesuai dengan urutan warna target.
- **Hasil:** Program mencetak hasil dari setiap percobaan, apakah berhasil atau tidak, serta menampilkan semua hasil di akhir.

## 2. Source Code

```
package main

import (
    "fmt"
    "strings"
```

```

)

func main() {
    var n int
    var pita []string

    fmt.Print("Masukkan jumlah bunga (atau ketik SELESAI
untuk berhenti lebih awal): ")
    fmt.Scanln(&n)

    for i := 1; i <= n; i++ {
        var bunga string
        fmt.Printf("Bunga %d: ", i)
        fmt.Scanln(&bunga)

        if strings.ToLower(bunga) == "selesai" {
            break
        }

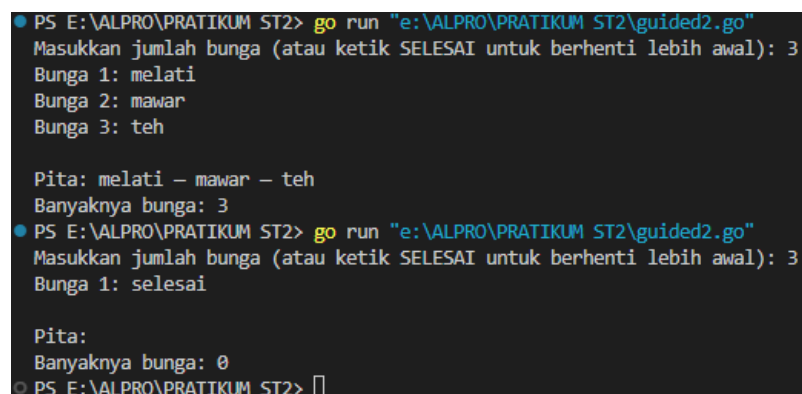
        pita = append(pita, bunga)
    }

    hasilPita := strings.Join(pita, " - ")

    fmt.Printf("\nPita: %s\n", hasilPita)
    fmt.Printf("Banyaknya bunga: %d\n", len(pita))
}

```

## Hasil Program



```

PS E:\ALPRO\PRATIUM ST2> go run "e:\ALPRO\PRATIUM ST2\guided2.go"
Masukkan jumlah bunga (atau ketik SELESAI untuk berhenti lebih awal): 3
Bunga 1: melati
Bunga 2: mawar
Bunga 3: teh

Pita: melati - mawar - teh
Banyaknya bunga: 3
PS E:\ALPRO\PRATIUM ST2> go run "e:\ALPRO\PRATIUM ST2\guided2.go"
Masukkan jumlah bunga (atau ketik SELESAI untuk berhenti lebih awal): 3
Bunga 1: selesai

Pita:
Banyaknya bunga: 0
PS E:\ALPRO\PRATIUM ST2>

```

## Penjelasan Program

- Input N: Program pertama-tama meminta pengguna untuk memasukkan jumlah bunga yang akan dimasukkan.

- Proses Input: Dalam setiap iterasi, program meminta nama bunga dan menyimpannya dalam slice pita. Jika pengguna mengetikkan "SELESAI", program akan langsung berhenti.
- Penggabungan Nama Bunga: Setelah selesai, program akan menggabungkan nama-nama bunga dengan separator " — " menggunakan fungsi strings.Join.
- Output: Program menampilkan isi pita (nama-nama bunga yang dimasukkan) dan jumlah bunga yang berhasil dimasukkan.

### 3. Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKiri, beratKanan float64

    for {
        fmt.Print("Masukkan berat belanjaan di
kedua kantong (pisahkan dengan spasi): ")
        fmt.Scanln(&beratKiri, &beratKanan)

        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Proses selesai. Berat kantong
tidak boleh negatif.")
            break
        }

        totalBerat := beratKiri + beratKanan
        if totalBerat > 150 {
            fmt.Println("Proses selesai. Total berat
melebihi 150 kg.")
            break
        }

        selisih := math.Abs(beratKiri - beratKanan)
        if selisih >= 9 {
            fmt.Println("Sepeda motor Pak Andi akan
oleng: true")
        } else {
```

```

        fmt.Println("Sepeda motor Pak Andi akan
oleng: false")
    }

    mencapai 9 kg atau lebih
    if beratKiri >= 9 || beratKanan >= 9 {
        fmt.Println("Proses selesai.")
        break
    }
}
}

```

#### Hasil Program

```

PS E:\ALPRO\PRATIKUM ST2> go run "e:\ALPRO\PRATIKUM ST2\guided3.go"
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 5
Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 5
Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 5
Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 5
Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 5
Sepeda motor Pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 10
Sepeda motor Pak Andi akan oleng: true
Proses selesai.
PS E:\ALPRO\PRATIKUM ST2> 

```

#### Penjelasan Program

1. Input Berat: Program meminta input dua buah bilangan beratKiri dan beratKanan yang mewakili berat kantong kiri dan kanan pada sepeda motor Pak Andi.
2. Pengecekan Kondisi:
  - o Negatif: Jika salah satu kantong memiliki berat negatif, program langsung berhenti.
  - o Total Berat: Jika total berat kedua kantong melebihi 150 kg, program juga berhenti.
  - o Selisih Berat: Program menghitung selisih berat antara kedua kantong, jika selisih lebih dari atau sama dengan 9 kg, akan menampilkan true (motor akan oleng), jika tidak, false.
  - o Berat  $\geq 9$  kg: Jika berat salah satu kantong mencapai 9 kg atau lebih, program akan berhenti.

#### 4. Source Code

```

package main

import (

```

```

    "fmt"
)

func main() {
    var beratParsel int
    var biayaPerKg int = 10000
    var biayaTambahanPerGramLebihDari500 int = 5
    var biayaTambahanPerGramKurangDari500 int = 15

    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&beratParsel)

    beratKg := beratParsel / 1000
    sisaGram := beratParsel % 1000

    biayaKg := beratKg * biayaPerKg

    biayaSisaGram := 0
    if beratKg > 10 {
        biayaSisaGram = 0
    } else if sisaGram >= 500 {
        biayaSisaGram = sisaGram *
biayaTambahanPerGramLebihDari500
    } else {
        biayaSisaGram = sisaGram *
biayaTambahanPerGramKurangDari500
    }
}

```

```

        totalBiaya := biayaKg + biayaSisaGram

        fmt.Printf("Detail berat: %d kg + %d gr\n",
beratKg, sisaGram)

        fmt.Printf("Detail biaya: RP. %d + RP. %d\n",
biayaKg, biayaSisaGram)

        fmt.Printf("Total biaya: RP. %d\n", totalBiaya)
    }

```

### Hasil Program

```

PS E:\ALPRO\PRATIUM ST2> go run "e:\ALPRO\PRATIUM ST2\Modul 2\ungieded4.go"
Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: RP. 80000 + RP. 2500
Total biaya: RP. 82500
PS E:\ALPRO\PRATIUM ST2> 

```

### Penjelasan Program

- Input: Program meminta input berat parsel dalam gram.
- Perhitungan:
- Berat parsel dipecah menjadi kilogram (kg) dan sisa gram.
- Biaya dihitung berdasarkan kilogram (Rp. 10.000 per kg).
- Jika sisa gram  $\geq 500$ , tambahan biaya Rp. 5 per gram; jika kurang dari 500 gram, tambahan biaya Rp. 15 per gram.
- Jika total berat  $> 10$  kg, maka sisa gram diabaikan.
- Output: Program menampilkan detail berat, detail biaya, dan total biaya pengiriman.

### 5. Source Code

```
package main
```

```
import "fmt"

func main() {
    var nam float64
    var nmk string

    // Input nilai akhir mata kuliah
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)

    // Menentukan NMK berdasarkan NAM
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 && nam <= 80 {
        nmk = "AB"
    } else if nam > 65 && nam <= 72.5 {
        nmk = "B"
    } else if nam > 57.5 && nam <= 65 {
        nmk = "BC"
    } else if nam > 50 && nam <= 57.5 {
        nmk = "C"
    } else if nam > 40 && nam <= 50 {
        nmk = "D"
    } else if nam <= 40 {
        nmk = "E"
    }
}
```



```
}

// Output hasil penilaian
fmt.Println("Nilai mata kuliah: ", nmk)
}
```

### Hasil Program

- Program akan meminta input berupa nilai akhir mata kuliah (NAM).
- Program kemudian mengevaluasi nilai tersebut menggunakan kondisi if-else untuk menentukan nilai mata kuliah (NMK) berdasarkan kriteria yang telah diberikan.
- Setelah menentukan NMK, program akan menampilkan hasil nilai mata kuliah.

### Penjelasan Program

#### 6. Source Code

```
package main

import (
    "fmt"
    "math"
)

func f(k int) float64 {
    numerator := math.Pow(float64(4*k+2), 2)
```

```

        denominator := float64((4*k + 1) * (4*k + 3))

        return numerator / denominator
    }

func sqrt2Approximation(K int) float64 {
    product := 1.0
    for k := 0; k <= K; k++ {
        product *= f(k)
    }
    return product
}

func main() {
    var K int

    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&K)
    fmt.Printf("Nilai f(K) = %.10f\n", f(K))

    fmt.Print("Masukkan nilai K untuk akar 2: ")
    fmt.Scan(&K)
    fmt.Printf("Nilai akar 2 = %.10f\n",
sqrt2Approximation(K))
}

```

## Hasil Program

```
PS E:\ALPRO\PRATIUM ST2> go run "e:\ALPRO\PRATIUM ST2\Modul 2\unguided6.go"
Masukkan nilai K: 80
Nilai f(K) = 1.0000096448
Masukkan nilai K untuk akar 2: 70
Nilai akar 2 = 1.4129692218
PS E:\ALPRO\PRATIUM ST2> █
```

## Penjelasan Program

Fungsi f(k):

- Fungsi ini menghitung faktor berdasarkan rumus:  
$$f(k) = \frac{(4k+2)^2(4k+1)(4k+3)}{(4k+1)(4k+3)(4k+2)^2}$$
- Ini adalah bagian penting dari perkalian yang akan digunakan untuk mendekati nilai akar 2.
- Fungsi sqrt2Approximation(K):
  - Fungsi ini melakukan perkalian dari hasil f(k) dari k = 0 hingga k = K. Semakin besar nilai K, semakin banyak faktor yang digunakan dalam perkalian, dan hasilnya semakin mendekati nilai akar 2.
- Fungsi main:
  - Program akan meminta pengguna memasukkan nilai K untuk menghitung nilai fungsi f(k) dan kemudian menggunakan nilai K lain (atau sama) untuk menghitung pendekatan nilai akar 2.

## 7. Source Code

```
package main

import (
    "fmt"
)
```

```
func main() {  
    var b int  
  
    fmt.Print("Masukkan bilangan: ")  
    fmt.Scan(&b)  
  
    if b <= 1 {  
        fmt.Println("Bilangan harus lebih dari 1.")  
        return  
    }  
  
    fmt.Print("Faktor: ")  
    var faktorCount int  
    for i := 1; i <= b; i++ {  
        if b%i == 0 {  
            fmt.Print(i, " ")  
            faktorCount++  
        }  
    }  
    fmt.Println()  
  
    if faktorCount == 2 {  
        fmt.Println("Prima: true")  
    } else {
```

```
        fmt.Println("Prima: false")
    }
}
```

### Hasil Program

```
PS E:\ALPRO\PRATIKUM ST2> go run "e:\ALPRO\PRATIKUM ST2\Modul 2\unguided7.go"
Masukkan bilangan: 9.2
Faktor: 1 3 9
Prima: false
```

### Penjelasan Program

- Input Bilangan: Program meminta pengguna untuk memasukkan bilangan bulat b.
- Validasi Input: Jika bilangan yang dimasukkan kurang dari atau sama dengan 1, program akan menampilkan pesan bahwa bilangan harus lebih dari 1, karena bilangan 1 atau kurang tidak relevan dalam konteks ini.
- Mencari Faktor: Program melakukan perulangan dari 1 hingga b. Jika bilangan i habis membagi b, maka i adalah faktor dari b dan ditampilkan.
- Mengecek Bilangan Prima: Jika jumlah faktor yang ditemukan persis 2 (yaitu 1 dan dirinya sendiri), maka bilangan tersebut adalah bilangan prima dan akan menampilkan "Prima: true". Jika tidak, maka program menampilkan "Prima: false".