

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2
MODUL 2
REVIEW STRUKTUR CONTROL**



Oleh:

Ben Waiz Pintus Widyosaputro

2311102169

IF-11-02

**S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

I. DASAR TEORI

Struktur kontrol adalah sekumpulan perintah yang mengontrol aliran eksekusi suatu program. Pemrogram menggunakan struktur kontrol untuk menentukan aliran suatu program berdasarkan kondisi tertentu, sehingga memungkinkan program merespons secara dinamis terhadap situasi yang berbeda. Struktur kendali berfungsi sebagai pedoman dalam suatu program, seperti halnya keputusan dalam buku bergambar mengubah jalannya cerita.

Jenis Struktur Pengendalian Struktur pengendalian umumnya terbagi dalam dua kategori utama :

-Struktur Pengendalian Keputusan

Pernyataan IF: Menentukan blok kode yang akan dieksekusi ketika kondisi tertentu terpenuhi.

Pernyataan IF-ELSE: Memungkinkan eksekusi blok kode berbeda berdasarkan apakah kondisi bernilai benar atau salah.

Pernyataan SWITCH: Memfasilitasi percabangan berdasarkan beberapa kemungkinan hasil, memungkinkan pemilihan dari beberapa pilihan.

- Struktur Kontrol Perulangan

Loop For Loop: Digunakan untuk mengulang blok kode dengan jumlah iterasi yang telah ditentukan.

While Loop: Mengulangi blok kode selama kondisi tertentu terpenuhi.

Perulangan Do-While: Mirip dengan perulangan while, namun menjamin bahwa blok kode dieksekusi setidaknya satu kali sebelum kondisi diperiksa.

Pentingnya Struktur Pengendalian :

-Mengatur alur program.

Memungkinkan program untuk menjalankan kode yang berbeda berdasarkan input atau kondisi yang berubah.

-Peningkatan efisiensi: Iterasi memungkinkan pemrogram menghindari penulisan kode berulang, membuat program lebih ringkas dan mudah dipelihara.

-Dukungan untuk logika kompleks: Keputusan dan loop bersyarat dapat digunakan untuk mengimplementasikan logika yang lebih kompleks,

sehingga meningkatkan kemampuan program untuk memecahkan masalah yang lebih kompleks.

II. GUIDED

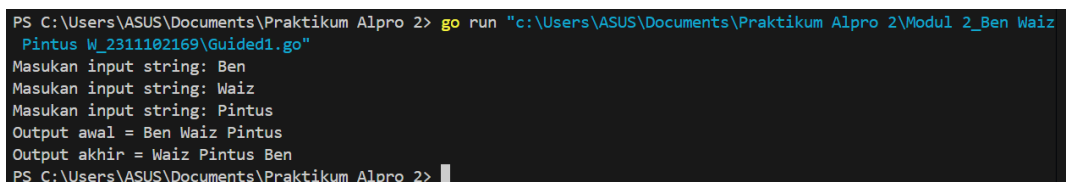
Guided 1

```
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)
}
```

Screenshot output



```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Guided1.go"
Masukan input string: Ben
Masukan input string: Waiz
Masukan input string: Pintus
Output awal = Ben Waiz Pintus
Output akhir = Waiz Pintus Ben
PS C:\Users\ASUS\Documents\Praktikum Alpro 2>
```

Penjelasan

Pengguna memasukkan tiga string secara bergantian, program mencetak urutan awal dari ketiga string tersebut. Selanjutnya, urutan string diubah menggunakan variabel sementara untuk melakukan penukaran.

Guided 2

```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Tahun : ")
    fmt.Scan(&tahun)
    fmt.Printf("Kabisat : %t \n", (tahun%4 == 0 && (tahun%100 != 0 || tahun%400
== 0)))
}
```

Screenshot output

```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Guided\Guided2.go"
Tahun : 2024
Kabisat : true
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> █
```

Penjelasan

Program meminta input berupa angka tahun dari pengguna, Setelah menerima input, program mengevaluasi apakah tahun tersebut memenuhi syarat sebagai tahun kabisat. Syarat tahun kabisat adalah jika tahun habis dibagi 4, tetapi tidak habis dibagi 100 kecuali juga habis dibagi 400.

Guided 3

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var r float64
    fmt.Print("Jejari = ")
    fmt.Scan(&r)
    volume := 4.0 / 3.0 * math.Pi * math.Pow(r, 3)
    luas := 4 * math.Pi * math.Pow(r, 2)
    fmt.Println("Bola dengan jejari ", r, "memiliki volume", volume, "dan luas
kulit", luas)
}
```

Screenshot output

```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Guided\Guided3.go"
Jejari = 9
Bola dengan jejari 9 memiliki volume 3053.6280592892795 dan luas kulit 1017.8760197630929
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> █
```

Penjelasan

Program meminta input nilai jejari, setelah itu rumus volume bola dan luas permukaan bola dihitung. Lalu hasil perhitungan volume dan luas permukaan bola ditampilkan.

III. UNGUIDED

Unguided 2B 1

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    correctOrder := []string{"merah", "kuning", "hijau", "ungu"}

    reader := bufio.NewReader(os.Stdin)
    var hadError bool

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)

        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        colors := strings.Split(input, " ")

        for j := 0; j < 4; j++ {
            if colors[j] != correctOrder[j] {
                hadError = true
                break
            }
        }
    }

    if !hadError {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    }
}
```

Screenshot output

```

PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\Unguided281.go"
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
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\Unguided281.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
PS C:\Users\ASUS\Documents\Praktikum Alpro 2>

```

Penjelasan

pengguna diminta memasukkan urutan warna sebanyak lima kali dan program memeriksa apakah urutan warna yang dimasukkan sesuai dengan urutan yang benar, yaitu "merah", "kuning", "hijau", "ungu". Program menggunakan `bufio.NewReader` untuk membaca input dari pengguna, memisahkannya berdasarkan spasi menggunakan `strings.Split()`, dan memeriksa apakah setiap warna sesuai dengan urutan yang ditentukan. Jika ada satu kesalahan dalam urutan selama lima percobaan, variabel `hadError` akan menjadi `true`. Setelah lima percobaan, program akan mencetak "BERHASIL: true" jika tidak ada kesalahan, atau "BERHASIL: false" jika ada kesalahan.

Unguided 2B 2

```

package main

import "fmt"

func main() {
    var pita string
    var banyakBunga int

    for {
        var bunga string
        fmt.Print("Bunga (ketik 'SELESAI' untuk berhenti): ")
        fmt.Scanln(&bunga)

        if bunga == "SELESAI" {
            break
        }

        if pita != "" {
            pita += " - "
        }
        pita += bunga
    }
}

```



```

        banyakBunga++
    }

    fmt.Println("Pita:", pita)
    fmt.Println("Banyaknya bunga:", banyakBunga)
}

```

Screenshot output

```

PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\Unguided2B2.go"
Bunga (ketik 'SELESAI' untuk berhenti): kertas
Bunga (ketik 'SELESAI' untuk berhenti): mawar
Bunga (ketik 'SELESAI' untuk berhenti): tulip
Bunga (ketik 'SELESAI' untuk berhenti): SELESAI
Pita: kertas - mawar - tulip
Banyaknya bunga: 3
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\Unguided2B2.go"
Bunga (ketik 'SELESAI' untuk berhenti): SELESAI
Pita:
Banyaknya bunga: 0
PS C:\Users\ASUS\Documents\Praktikum Alpro 2>

```

Penjelasan

Program menerima input nama bunga dari pengguna dan menambahkannya ke sebuah string sambil menghitung jumlah bunga yang dimasukkan. Program berjalan dalam sebuah loop yang terus meminta pengguna memasukkan nama bunga. Jika pengguna mengetik "SELESAI", loop akan berhenti.

Unguided 2B 3

```

package main

import "fmt"

func main() {
    for {
        var berat1, berat2 float64

        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
        _, err := fmt.Scan(&berat1, &berat2)
        if err != nil {
            fmt.Println("Input tidak valid.")
            return
        }

        if berat1+berat2 > 150 || berat1 < 0 || berat2 < 0 {
            fmt.Println("Proses selesai.")
        }
    }
}

```

```

        break
    }

    selisih := berat1 - berat2
    if selisih < 0 {
        selisih = -selisih
    }
    akanOleng := selisih >= 9

    fmt.Printf("Sepeda motor pak Andi akan oleng: %t\n", akanOleng)
}
}

```

Screenshot output

```

PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\Unguided283.go"
Masukkan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukkan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukkan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> 

```

Penjelasan

Program terus meminta input dua nilai berat dari pengguna, jika input tidak valid, program akan berhenti dan menampilkan pesan kesalahan. Program juga akan berhenti jika jumlah berat melebihi 150 kg atau jika ada berat negatif, dengan menampilkan pesan "Proses selesai". Selisih antara kedua berat dihitung, dan jika selisihnya lebih dari atau sama dengan 9 kg akan bernilai true. Menandakan bahwa sepeda motor akan oleng. Hasil ini ditampilkan ke pengguna setiap kali input diberikan.

Unguided 2B 4

Before

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var k float64

```

```

fmt.Print("Masukkan nilai K: ")
fmt.Scan(&k)

fk := math.Pow(4*k+2, 2) / ((4*k + 1) * (4*k + 3))

fmt.Printf("Nilai f(k) = %.10f\n",fk)
}

```

Screenshot output

```

PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\Unguided2B4before.go"
Masukkan nilai K: 100
Nilai f(k) = 1.0000061880
PS C:\Users\ASUS\Documents\Praktikum Alpro 2>

```

After

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var k int

    for {
        fmt.Print("Masukkan nilai K: ")
        _, err := fmt.Scan(&k)
        if err != nil {
            fmt.Println("Input tidak valid.")
            return
        }

        if k < 0 {
            fmt.Println("Nilai K harus positif.")
            continue
        }

        akarDua := 0.0
        for i := 0; i <= k; i++ {
            pembilang := math.Pow(4*float64(i)+2, 2)
            penyebut := (4*float64(i) + 1) * (4*float64(i) + 3)
            akarDua += pembilang / penyebut
        }

        fmt.Printf("Nilai akar 2 = %.10f\n", akarDua)
    }
}

```

```
}  
}
```

Screenshot output

```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz  
Pintus W_2311102169\Unguided\Unguided2B4after.go"  
Masukkan nilai K: 10  
Nilai akar 2 = 11.3870201908  
Masukkan nilai K: 100  
Nilai akar 2 = 101.3920802736  
Masukkan nilai K: 1000  
Nilai akar 2 = 1001.3926366441  
Masukkan nilai K: 
```

Penjelasan

Program meminta input nilai k secara berulang, di mana k harus bernilai positif. Jika nilai yang dimasukkan tidak valid atau negatif, program akan memberikan pesan kesalahan dan meminta input ulang. Untuk setiap iterasi dari 0 hingga k, program menghitung sebuah ekspresi yang melibatkan perhitungan pembilang dan penyebut dari rumus tertentu, yang kemudian ditambahkan ke variabel akarDua sebagai pendekatan untuk nilai akar 2. Setelah selesai menghitung, program mencetak hasil pendekatan tersebut hingga 10 digit desimal.

Unguided 2C 1

```
package main  
  
import "fmt"  
  
func main() {  
    var berat int  
    fmt.Print("Masukkan berat parsel (gram): ")  
    fmt.Scan(&berat)  
  
    kg := berat / 1000  
    gram := berat % 1000  
  
    biayaTotal := kg * 10000  
    biayaTambahan := 0  
  
    if gram >= 500 {  
        biayaTambahan = gram * 5  
    } else if gram > 0 {  
        biayaTambahan = gram * 15  
    }  
  
    if kg >= 10 {
```

```

        biayaTambahan = 0
    }

    biayaTotal += biayaTambahan

    fmt.Printf("Detail berat: %d kg + %d gram\n", kg, gram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", kg*10000,
biayaTambahan)
    fmt.Printf("Total biaya: Rp. %d\n", biayaTotal)
}

```

Screenshot output

```

PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\Unguided2C1.go"
Masukkan berat parcel (gram): 8500
Detail berat: 8 kg + 500 gram
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> 

```

Penjelasan

Program yang menghitung biaya pengiriman berdasarkan berat parcel dalam gram. Pertama, berat diinput oleh pengguna dan dikonversi menjadi kilogram dan sisa gram. Biaya dasar dihitung Rp 10.000 per kilogram, dan biaya tambahan dihitung berdasarkan sisa gram: jika sisa lebih dari atau sama dengan 500 gram, biaya tambahan Rp 5 per gram, sedangkan jika kurang dari 500 gram, biaya tambahan Rp 15 per gram. Namun, jika berat total lebih dari atau sama dengan 10 kg, biaya tambahan akan digratiskan. Program kemudian menampilkan detail berat dan biaya total.

Unguided 2C 2

```

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)
}

```

Screenshot output

```

PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben Waiz
Pintus W_2311102169\Unguided\tempCodeRunnerFile.go"
Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: A
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> 

```

Penjelasan

Program yang menentukan nilai huruf berdasarkan nilai akhir mata kuliah yang dimasukkan pengguna. Pengguna diminta untuk memasukkan nilai akhir dalam bentuk desimal. Program kemudian menggunakan serangkaian pernyataan if-else untuk mengevaluasi nilai tersebut dan menetapkan nilai huruf (nmk) sesuai dengan rentang yang telah ditentukan: A, AB, B, BC, C, D, atau E.

Unguided 2C 3

```

package main

import "fmt"

func main() {
    var bil int
    var prima bool = true
    fmt.Print("Bilangan: ")
    fmt.Scan(&bil)
    fmt.Print("Faktor: ")
    for i := 1; i <= bil; i++ {
        if bil%i == 0 {
            fmt.Print(i, " ")
            if !(i == 1 || i == bil) {
                prima = false
            }
        }
    }
}

```

```
    }  
    }  
    }  
    fmt.Println()  
    fmt.Print("Prima: ", prima)  
}
```

Screenshot output

```
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> go run "c:\Users\ASUS\Documents\Praktikum Alpro 2\Modul 2_Ben waiz  
Pintus W_2311102169\Unguided\Unguided2C3.go"  
Bilangan: 12  
Faktor: 1 2 3 4 6 12  
Prima: false  
PS C:\Users\ASUS\Documents\Praktikum Alpro 2> █
```

Penjelasan

Program yang menentukan faktor dari sebuah bilangan dan memeriksa apakah bilangan tersebut adalah bilangan prima. Pengguna diminta untuk memasukkan sebuah bilangan bulat (bil). Program kemudian menggunakan loop for untuk mencari dan mencetak semua faktor dari bilangan tersebut dengan memeriksa setiap angka dari 1 hingga bilangan itu sendiri. Jika angka tersebut merupakan faktor (yaitu, jika bil dibagi angka tersebut menghasilkan sisa 0), angka tersebut dicetak. Selain itu, program juga memeriksa apakah bilangan tersebut adalah bilangan prima; bilangan prima didefinisikan sebagai bilangan yang hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri. Jika ditemukan faktor lain selain 1 dan bilangan itu sendiri, variabel prima diubah menjadi false. Setelah mencetak semua faktor, program mencetak status prima dari bilangan tersebut.