

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2
MODUL 2
REVIEW STRUKTUR KONTROL**



Oleh:

NAMA: MOHAMMAD REYHAN ARETHA FATIN

NIM: 103112400078

KELAS: IF-12-01

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

I. DASAR TEORI

Paradigma Perulangan merupakan salah satu kontrol yang memungkinkan suatu instruksi yang sama dilakukan berulang kali dalam waktu atau jumlah yang lama. Tanpa instruksi perulangan, maka suatu intruksi akan ditulis dalam jumlah yang sangat banyak.

Karakteristik For-Loop (perulangan berdasarkan iterasi) Salah satu intruksi perulangan yang paling mudah adalah for-loop, yang mana dengan instruksi ini dapat digunakan untuk mengulangi intruksi sebanyak n kali (iterasi). Batasan besar nilai n menyesuaikan dengan batasan besar nilai dari n menyesuaikan dengan Batasan dari tipe data integer yang di gunakan.

Instruksi for-loop memiliki beberapa komponen, yaitu:

1. inisialisasi merupakan assignment variabel iterasi yang bertipe integer. Pada contoh variabel iterasi = 0 atau 1, artinya iterasi dimulai dari 0 atau 1.
2. kondisi merupakan suatu operasi bernilai boolean yang menyatakan kapan perulangan harus dilakukan. Pada contoh di atas kondisi adalah variabel iterasi $\leq n$ (kurang dari atau sama dengan)
3. update merupakan ekspresi yang menyatakan perubahan nilai dari variabel iterasi. Pada contoh di atas update adalah variabel iterasi = variabel iterasi + 1.

Paradigma percabangan dalam pemrograman adalah suatu mekanisme yang memungkinkan program untuk mengeksekusi kode berdasarkan suatu kondisi atau syarat tertentu. Tanpa percabangan, eksekusi program berjalan secara sekuensial dari baris pertama hingga terakhir. Dengan percabangan, alur eksekusi program dapat bercabang sesuai dengan kondisi yang diberikan.

Karakteristik If-Then Penulisan struktur kontrol percabangan dengan menggunakan if-then pada dasarnya terdiri dari dua bagian, yaitu:

1. Kondisi, yaitu sesuatu syarat atau ketentuan dari suatu percabangan. Kondisi ini harus bernilai boolean, baik itu variabel ataupun operasi tipe data.
2. Aksi, yaitu kumpulan instruksi yang akan dilakukan apabila kondisi terpenuhi atau bernilai true. Artinya baris aksi tidak secara default dieksekusi, tetapi ada syarat yang harus terpenuhi terlebih dahulu sehingga aksi dapat dieksekusi.

II. GUIDED

SOURCE CODE GUIDED 1

```
coso1 > go coso1.go > main
3  import "fmt"
4
5  func main() {
6      var greetings = "selamat datang di dunia DAP"
7      var a, b int
8      fmt.Println(greetings)
9      fmt.Scanln(&a, &b)
10     fmt.Printf("%v + %v = %v\n", a, b, a+b)
11 }
12
```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\Pr
selamat datang di dunia DAP
5 5
5 + 5 = 10
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> 
```

DEKSRIPSI

Program ini mencetak pesan “selamat datang di dunia DAP” dan menerima dua input angka dari pengguna. program meminta input dua nilai integer dan menyimpannya dalam variabel a dan b. Terakhir, program mencetak hasil penjumlahan kedua angka tersebut dengan format yang ditentukan, yaitu mencetak angka pertama dan kedua serta hasil penjumlahannya. Output dari program ini menunjukkan input dan hasil penjumlahan yang dilakukan oleh pengguna.

SURCE CODE GUIDED 2

```
coso2 > go coso2.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      for i := 1; i <= 5; i++ {
7          fmt.Println("iterasi ke-", i)
8      }
9  }
```

OUTPUT

```
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\Week2\coso2.go"
iterasi ke- 1
iterasi ke- 2
iterasi ke- 3
iterasi ke- 4
iterasi ke- 5
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2>
```

DEKSRIPSI

Program ini menggunakan loop for untuk mencetak pesan "iterasi ke-" diikuti oleh nomor iterasi dari 1 hingga 5. Setiap kali loop dijalankan, nilai variabel i akan meningkat dan ditampilkan dalam output, yang menunjukkan urutan iterasi dari 1 sampai 5.

SOURCE CODE GUIDED 3

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     nilai := 80
7     pcthaadir := 0.75
8     adaTubes := true
9
10    var indeks string
11    if nilai > 75 && adaTubes {
12        indeks = "A"
13    } else if nilai > 65 {
14        indeks = "B"
15    } else if nilai > 55 && pcthaadir > 0.7 {
16        indeks = "C"
17    } else {
18        indeks = "F"
19    }
20    fmt.Printf("nilai %d dengan kehadiran %.75% dan buat tubes=%t, mendapat indeks %s\n", nilai, pcthaadir*100, adaTubes, indeks)
21 }
22
```

OUTPUT

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\Praktikum
nilai 80 dengan kehadiran 75% dan buat tubes=true, mendapat indeks A
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2>
```

DEKSRISI

Program ini menentukan indeks nilai mahasiswa berdasarkan nilai, persentase kehadiran, dan apakah mahasiswa mengikuti tugas (tubes). Jika nilai lebih dari 75 dan ada tugas, indeksnya "A". Jika nilai lebih dari 65, indeksnya "B", lebih dari 55 dan persentase kehadiran lebih dari 70%, indeksnya "C", dan selain itu, indeksnya "F". Program kemudian mencetak nilai, persentase kehadiran, status tugas, dan indeks nilai yang diperoleh. Output yang ditampilkan menunjukkan bahwa nilai mahasiswa 80 dengan kehadiran 75% dan ada tugas, sehingga mendapatkan indeks "A".

III. UNGUIDED

SOURCE CODE UNGUIDED 1 (2A NO2)

```
1 package main
2 //103112400078 MOHAMMAD REYHAN ARETHA FATIN
3 import "fmt"
4 func main() {
5     var tahun int
6     fmt.Print("Tahun: ")
7     fmt.Scan(&tahun)
8     if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {
9         fmt.Println("kabisat: true")
10    } else {
11        fmt.Println("kabisat: false")
12    }
13 }
```

OUTPUT

```
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\Praktikum
Tahun: 2016
kabisat: true
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\Praktikum
Tahun: 2000
kabisat: true
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\Praktikum
Tahun: 2018
kabisat: false
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> █
```

DEKSRIPSI

Program menentukan apakah tahun tersebut merupakan tahun kabisat. Tahun kabisat adalah tahun yang dapat dibagi 400 atau tahun yang dapat dibagi 4 tetapi tidak dibagi 100. Jika salah satu kondisi tersebut terpenuhi, program mencetak "kabisat: true", jika tidak, mencetak "kabisat: false". Contohnya, tahun 2016 dan 2000 adalah tahun kabisat, sedangkan 2018 bukan tahun kabisat.

SOURCE CODE UNGUIDED 2 (2B NO 4 A)

```
latsol2bno4 > latsol2.go > main
1 package main
2
3 // 103112400078 MOHAMMAD REYHA ARETHA FATIN
4 import "fmt"
5
6 func main() {
7     var k float64
8     fmt.Print("Nilai K: ")
9     fmt.Scan(&k)
10    hasil := float64((4*k+2)*(4*k+2)) / float64((4*k+1)*(4*k+3))
11    fmt.Printf("Nilai f(K) = %.10f\n", hasil)
12 }
13
```

OUTPUT

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\Week2\latsol2.go"
Nilai K: 100
Nilai f(K) = 1.0000061880
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2>
```

DEKSRIPSI

Program ini menghitung dan mencetak hasil dari sebuah fungsi matematika yang melibatkan operasi perkalian dan pembagian. Fungsi yang dihitung adalah $f(k) = \frac{((4*k + 2)*(4*k + 2))}{((4*k + 1)*(4*k + 3))}$. Hasil perhitungan kemudian ditampilkan dengan format angka desimal yang memiliki 10 angka di belakang koma. Sebagai contoh, jika k diinputkan dengan nilai 100, hasilnya adalah 1.0000061880.

SOURCE CODE UNGUIDED 2 (2B NO 4 B)

```
latsol2bno4 > go latsol2bno4b.go > main
1  package main
2
3  //10311240078 MOHAMMAD REYHAN ARETHA FATIN
4  import "fmt"
5
6  func main() {
7      var k int
8      var hasil float64
9      fmt.Print("Nilai K = ")
10     fmt.Scan(&k)
11     hasil = 1.0
12     for i := 0; i <= k; i++ {
13         pembilang := (4*i + 2) * (4*i + 2)
14         penyebut := (4*i + 1) * (4*i + 3)
15         hasil = hasil * float64(pembilang) / float64(penyebut)
16     }
17     fmt.Printf("nilai akar 2 = %.10f\n", hasil)
18 }
```

OUTPUT

```
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\PraktikumSem2\Week2\latsol2bno4b.go"
Nilai K = 100
nilai akar 2 = 1.4133387072
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> 
```

DEKSRIPSI

Program ini menghitung hasil dari sebuah deret matematika dengan menggunakan loop for. Di dalam loop, nilai pembilang dan penyebut dihitung berdasarkan rumus $(4*i + 2) * (4*i + 2)$ untuk pembilang, dan $(4*i + 1) * (4*i + 3)$ untuk penyebut. Hasil perhitungan diakumulasikan dalam variabel hasil dengan setiap iterasi. Setelah loop selesai, hasil akhirnya dicetak dengan format desimal yang memiliki 10 angka di belakang koma. Sebagai contoh, jika k diinputkan dengan nilai 100, hasilnya adalah 1.4133387072.

SOURCE CODE UNGUIDED 3 (2C NO1)

```
latsol2Cno1 > go latsol3.go > main
1  package main
2  // 103112400978 MOHAMMAD REYHAN ARETHA FATIN
3  import "fmt"
4  func main() {
5      var BeratGram int
6      fmt.Print("Berat parcel (gram): ")
7      fmt.Scan(&BeratGram)
8      BeratKg := BeratGram / 1000
9      SisaGram := BeratGram % 1000
10     BiayaKg := BeratKg * 10000
11     BiayaSisa := 0
12
13     if BeratKg > 10 && SisaGram < 1000 {
14         BiayaSisa = 0
15     } else if SisaGram >= 500 {
16         BiayaSisa = SisaGram * 5
17     } else {
18         BiayaSisa = SisaGram * 15
19     }
20     totalBiaya := BiayaKg + BiayaSisa
21     fmt.Printf("Detail berat: %d kg + %d gr\n", BeratKg, SisaGram)
22     fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", BiayaKg, BiayaSisa)
23     fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
24 }
```

OUTPUT

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
Berat parcel (gram): 9500
Detail berat: 9 kg + 500 gr
Detail biaya: Rp. 90000 + Rp. 2500
Total biaya: Rp. 92500
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> go run "d:\ALGORITMA PROGRAMING\Pr
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS D:\ALGORITMA PROGRAMING\PraktikumSem2\Week2> |
```

DEKSRIPSI

Program ini meminta input berat dalam gram dari pengguna dan menghitung biaya pengiriman berdasarkan berat tersebut. Berat diubah menjadi kilogram dan sisanya dihitung untuk menentukan biaya tambahan berdasarkan jumlah gram. Jika berat lebih dari 10 kg dan sisa gram kurang dari 1000 gram, biaya tambahan adalah 0. Jika sisa gram lebih dari atau sama dengan 500, biaya tambahan adalah 5, dan jika tidak, biaya tambahan adalah 15. Program kemudian mencetak rincian berat dalam kilogram dan gram, biaya per kilogram, biaya tambahan, serta total biaya pengiriman.

IV. KESIMPULAN

mengulas penggunaan struktur kontrol dalam pemrograman, khususnya mengenai perulangan dan percabangan, yang sangat penting dalam pengembangan algoritma. Program-program yang dijelaskan menunjukkan penerapan konsep-konsep tersebut untuk menyelesaikan berbagai masalah, seperti menghitung jumlah, menentukan tahun kabisat, menghitung hasil fungsi matematika, dan menentukan biaya pengiriman. Penggunaan perulangan for dan percabangan if-else memungkinkan program untuk berfungsi secara dinamis dan menyesuaikan output dengan kondisi yang diberikan. Secara keseluruhan, pemahaman dan penerapan struktur kontrol ini esensial dalam menciptakan program yang efisien dan tepat guna dalam menyelesaikan masalah yang kompleks.

V. REFERENSI

Prasti Eko Yunanto, S.T., M.Kom. MODUL PRAKTIKUM 5 & 6 FOR & LOOP
ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA

Prasti Eko Yunanto, S.T., M.Kom. MODUL PRAKTIKUM 9 - IF-THEN
ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA

Prasti Eko Yunanto, S.T., M.Kom. MODUL PRAKTIKUM 10 - ELSE-IF
ALGORITMA DAN PEMROGRAMAN 1 S1 INFORMATIKA