LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 13
“REPEAT-UNTIL”

DISUSUN OLEH:

Muhammad Shabrian Fadly

103112400087

S1 IF-12-01

DOSEN:

Yohani Setiya Rafika Nur, M. Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024/2025

DASAR TEORI

REPEAT-UNTIL

Perulangan merupakan salah satu struktur kontrol yang memungkinkan suatu instruksi yang sama dilakukan berulang kali dalam waktu atau jumlah yang lama. Tanpa instruksi perulangan, maka suatu instruksi akan ditulis dalam jumlah yang sangat banyak. Pada modul 12 sebelumnya telah dipelajari terkait penggunaan struktur kontrol perulangan dengan while-loop, selanjutnya perulangan juga dapat dilakukan menggunakan **repeat-until**.

Penggunaan repeat-until pada dasarnya sama dengan while-loop di mana perulangan berdasarkan kondisi. Perbedaan terletak pada kondisi yang digunakan, pada while-loop kondisi yang harus didefinisikan adalah kondisi perulangannya, atau kapan perulangan itu terjadi, sedangkan pada repeat-until kondisi yang harus didefinisikan merupakan kondisi berhenti, atau kapan perulangan tersebut harus dihentikan.

Kondisi perulangan dan kondisi berhenti memiliki keterhubungan sifat komplemen, sehingga apabila kita mengetahui kondisi perulangannya, maka cukup dengan menambahkan operator negasi atau not untuk mengubah menjadi kondisi berhenti. Hal ini berlaku juga sebaliknya, komplemen dari kondisi berhenti adalah kondisi perulangan.

Karakterisik **REPEAT-UNTIL**

Komponen dari repeat-until sama dengan while-loop, yaitu terdapat kondisi dan aksi, hanya struktur penulisannya saja yang berbeda.

1)**Aksi**, merupakan kumpulan instruksi yang akan dilakukan perulangan. Aksi minimaldijalankan sekali, baru dilakukan pengecekan kondisi berhenti setelahnya. Apabilakondisi bernilai true, maka perulangan dihentikan.

2)**Kondisi/berhenti**, merupakan kondisi berhenti dari perulangan, harus bernilai falseselama perulangan dilakukan.

CONTOH SOAL

1. Contoh soal 1

Source code:

```
package main

import "fmt"

func main() {
    var word string
    var repetitions int
    fmt.Scan(&word, &repetitions)
    counter := 0
    for done := false; !done; {
        fmt.Println(word)
        counter++
        done = counter >= repetitions
    }
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul13\conso1\conso1.go"
pagi
3
pagi
pagi
pagi
PS D:\Coding manja> go run "d:\Coding manja\modul13\conso1\conso1.go"
kursi
5
kursi
kursi
kursi
kursi
kursi
```

Deskripsi Program:
Program di atas membaca dua input dari pengguna: sebuah kata (word) dan jumlah pengulangannya (repetitions). Kemudian, program akan mencetak kata tersebut sebanyak jumlah yang diminta. Variabel counter digunakan untuk menghitung berapa kali kata sudah dicetak, dan pengulangan akan berhenti ketika counter mencapai jumlah yang ditentukan (repetitions).

2. Contoh soal 2

Source code:

```
package main

import "fmt"

func main() {
    var number int
    var continueLoop bool
    for continueLoop = true; continueLoop; {
        fmt.Scan(&number)
        continueLoop = number <= 0
    }
    fmt.Printf("%d adalah bilangan bulat positif\n", number)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul13\conso2\conso2.go"
-5
-2
-7
0
5
5 adalah bilangan bulat positif
PS D:\Coding manja> go run "d:\Coding manja\modul13\conso2\conso2.go"
17
17 adalah bilangan bulat positif
```

Deskripsi Program:
Program di atas berfungsi untuk meminta input dari pengguna berupa bilangan bulat (number) dan terus meminta input selama bilangan tersebut tidak lebih besar dari 0. Ketika pengguna memasukkan bilangan yang lebih besar dari 0, program akan berhenti dan mencetak bahwa bilangan tersebut adalah bilangan bulat positif.

Contoh soal 3

Source code:

```
package main

import "fmt"

func main() {
    var x int
    var y int
    var selesai bool
    fmt.Scan(&x, &y)
    for selesai = false; !selesai; {
        x = x - y
        fmt.Println(x)
        selesai = x <= 0
    }
    fmt.Println(x == 0)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul13\conso3\conso3.go"
5
2
3
1
-1
false
PS D:\Coding manja> go run "d:\Coding manja\modul13\conso3\conso3.go"
15
3
12
9
6
3
0
true
PS D:\Coding manja> go run "d:\Coding manja\modul13\conso3\conso3.go"
25
5
20
15
10
5
0
true
```

Deskripsi Program: Program di atas melakukan pengurangan nilai x dengan y berulang-ulang dan mencetak hasilnya setiap kali setelah pengurangan. Program berhenti ketika nilai x menjadi kurang dari atau sama dengan 0. Setelah perulangan selesai, program akan mencetak true jika nilai x tepat menjadi 0, atau false jika tidak.

Latihan Soal

1. Latihan soal 1

   Statement Perulangan:

   Buatlah program yang digunakan untuk menghitung banyaknya digit dari suatu bilangan.

   Masukan berupa bilangan bulat positif.

   Keluaran berupa bilangan bulat yang menyatakan banyaknya digit dari bilangan yang diberikan pada masukan.

   Source code:

```
package main

import "fmt"

func main() {
    var number, digitCount int
    fmt.Scan(&number)
    for number > 0 {
        digitCount++
        number = number / 10
    }
    fmt.Print(digitCount)
}
```

   Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol1\latsol1.go"
5
1
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol1\latsol1.go"
234
3
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol1\latsol1.go"
78787
5
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol1\latsol1.go"
1894256
7
```

   Deskripsi Program: Program ini berfungsi untuk menghitung jumlah digit dari sebuah bilangan bulat positif dengan membagi angka tersebut secara berulang dengan 10 sampai nilai number menjadi 0. Setelah itu, jumlah digit yang dihitung akan ditampilkan. Program bekerja dengan baik untuk angka positif, namun tidak menangani kasus ketika input adalah 0.

2. Latihan soal 2

Statement Perulangan:

Buatlah program yang digunakan untuk mendapatkan bilangan bulat optimal dari bilangan yang telah diinputkan. Melakukan penjumlahan tiap perulangan mencapai pembulatan keatas dari bilangan yang diinputkan.

Masukan berupa bilangan desimal.

Keluaran terdiri dari bilangan hasil penjumlahan tiap perulangannya sampai pembulatan keatas dari bilangan yang diinputkan.

Source code:

```
package main

import "fmt"

func main() {
    var number, jumlah float64
    fmt.Scan(&number)
    jumlah = number + 0.1
    for jumlah <= float64(int(number)+1) {
        fmt.Printf("%.1f\n", jumlah)
        jumlah += 0.1
    }
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol2\latsol2.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol2\latsol2.go"
2.7
2.8
2.9
```

Deskripsi Program: Program ini melakukan pencetakan angka dengan kenaikan 0.1 mulai dari number + 0.1 dan terus meningkat sampai angka tersebut lebih besar dari number + 1. Program ini bekerja dengan baik untuk menghitung dan menampilkan hasil dalam format yang tepat, dengan penanganan angka desimal yang akurat.

3. Latihan soal 3

Statement perulangan:

Buatlah program yang digunakan untuk mendapatkan bilangan bulat optimal dari bilangan yang telah diinputkan. Melakukan penjumlahan tiap perulangan mencapai pembulatan keatas dari bilangan yang diinputkan.

Masukan berupa bilangan desimal.

Keluaran terdiri dari bilangan hasil penjumlahan tiap perulangannya sampai pembulatan keatas dari bilangan yang diinputkan.

Source code:

```
package main

import (
    "fmt"
)

func main() {
    var target, donasi, totalDonasi, jumlahDonatur int
    fmt.Scan(&target)
    for totalDonasi < target {
        fmt.Scan(&donasi)
        jumlahDonatur++
        totalDonasi += donasi
        fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", jumlahDonatur, donasi,
totalDonasi)
    }
    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", totalDonasi, jumlahDonatur)
}
```

Output:

```
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol3\latsol3.go"
300
100
Donatur 1: Menyumbang 100. Total terkumpul: 100
50
Donatur 2: Menyumbang 50. Total terkumpul: 150
200
Donatur 3: Menyumbang 200. Total terkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol3\latsol3.go"
500
150
Donatur 1: Menyumbang 150. Total terkumpul: 150
100
Donatur 2: Menyumbang 100. Total terkumpul: 250
50
Donatur 3: Menyumbang 50. Total terkumpul: 300
300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS D:\Coding manja> go run "d:\Coding manja\modul13\latsol3\latsol3.go"
200
300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
```

Deskripsi Program:
Program ini menghitung total donasi yang terkumpul dari beberapa donatur hingga mencapai target yang ditentukan. Setiap kali donatur memberikan sumbangan, jumlah donasi mereka dicatat, dan total donasi yang terkumpul diperbarui. Program akan terus meminta input donasi dari donatur hingga total donasi lebih besar atau sama dengan target. Setelah itu, program mencetak informasi bahwa target tercapai, beserta total donasi dan jumlah donatur yang telah menyumbang. Program ini bekerja dengan baik untuk mencatat dan mengelola sumbangan hingga mencapai tujuan yang diinginkan.