

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



DISUSUN OLEH:
Muhammad Faris Rachmadi
103112400079
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

13.1 Paradigma Perulangan

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 komplementer, sehingga apabila kita mengetahui kondisi perulangannya, maka cukup dengan menambahkan operator negasi atau not untuk mengubah menjadi kondisi berhenti. Hal ini berlaku juga sebaliknya, komplementer dari kondisi berhenti adalah kondisi perulangan. Pahami beberapa contoh yang diberikan berikut ini:

- Statement while-loop: "Menulis teks tertentu selama tinta pena masih ada". Statement repeat-until: "Menulis teks tertentu sampai tinta pena habis". Komplementer dari kondisi "tinta pena masih ada" adalah "tinta pena habis".
- Statement while-loop: "Saya makan suap demi suap selama saya masih lapar". Statement repeat-until: "Saya makan suap demi suap sampai saya merasa kenyang". Komplementer dari kondisi "saya masih lapar" adalah "saya merasa kenyang".

13.2 Karakteristik 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 minimal dijalankan sekali, baru dilakukan pengecekan kondisi berhenti setelahnya. Apabila kondisi bernilai true, maka perulangan dihentikan.
- 2) **Kondisi/berhenti**, merupakan kondisi berhenti dari perulangan, harus bernilai false selama perulangan dilakukan.

CONTOH SOAL

13.1.1 Contoh Soal 1

Source Code:

```
coso1 > -go coso1.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var word string
7      var repetitions int
8      fmt.Scan(&word, &repetitions)
9      counter := 0
10     for done := false; !done; {
11         fmt.Println(word)
12         counter++
13         done = (counter >= repetitions)
14     }
15 }
16
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13\coso1\coso1.go"
pagi 3
pagi
pagi
pagi
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13\coso1\coso1.go"
kursi 5
kursi
kursi
kursi
kursi
kursi
```

Deskripsi Program:

Tujuan program GO ini adalah untuk mencetak sebanyak jumlah pengulangan yang diinginkan oleh pengguna. Program akan dihentikan ketika jumlah kata yang dicetak mencapai jumlah yang diinginkan oleh pengguna.

13.1.2 Contoh Soal 2

Source Code:

```
coso2 > go coso2.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var number int
7      var continueLoop bool
8      for continueLoop = true; continueLoop; {
9          fmt.Scan(&number)
10         continueLoop = number <= 0
11     }
12     fmt.Printf("%d adalah bilangan bulat positif\n", number)
13 }
14
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go
nts\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13\coso2\coso2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go
nts\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13\coso2\coso2.go"
17
17 adalah bilangan bulat positif
```

Deskripsi Program:

Tujuan program GO ini adalah untuk memasukkan bilangan bulat positif. Program akan terus meminta input hingga pengguna memasukkan bilangan bulat positif

13.1.3 Contoh Soal 3

Source Code:

```
coso3 > -go coso3.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var x int
7      var y int
8      var selesai bool
9      fmt.Scan(&x, &y)
10     for selesai = false; !selesai; {
11         x = x - y
12         fmt.Println(x)
13         selesai = x <= 0
14     }
15     fmt.Println(x == 0)
16 }
17
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS  Code
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run
nts\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13\coso3\coso3.go
5
2
3
1
-1
false
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run
nts\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13\coso3\coso3.go
15
3
12
9
6
3
0
true
```

Deskripsi Program:

Tujuan program GO ini adalah untuk melakukan pengecekan apakah suatu bilangan merupakan kelipatan dari bilangan lainnya

Latihan Soal

1. Latihan Soal 1

Source Code:

```
latsol1 > -go latsol1.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var bilangan int
7      fmt.Scan(&bilangan)
8      jumlahDigit := 0
9      for {
10         jumlahDigit++
11         bilangan /= 10
12         if bilangan == 0 {
13             break
14         }
15     }
16     fmt.Print(jumlahDigit)
17 }
18
```

Output:

```
PROBLEMS OUTPUT TERMINAL PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\
ul13\latsol1\latsol1.go"
5
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\
ul13\latsol1\latsol1.go"
234
3
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\
ul13\latsol1\latsol1.go"
78787
5
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\
ul13\latsol1\latsol1.go"
1894256
7
```

Deskripsi Program:

Tujuan program GO ini adalah untuk menghitung jumlah digit dari sebuah bilangan integer yang dimasukkan oleh pengguna.

2. Latihan Soal 2

Source Code:

```
latsol2 > -go latsol2.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var x float64
7      var itg int
8      fmt.Scan(&x)
9      itg = int(x) + 1
10     for done := false; !done; {
11         x = x + 0.1
12         fmt.Printf("%.1f\n", x)
13         done = x > float64(itg)-0.11
14     }
15     fmt.Println(itg)
16 }
17
```

Output:

```
PROBLEMS  OUTPUT  TERMINAL  PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PENROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PENROGRAMAN\Modul13\latsol2\latsol2.go"
9.2
9.3
9.4
9.5
9.6
9.7
9.8
9.9
1
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PENROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PENROGRAMAN\Modul13\latsol2\latsol2.go"
2.7
2.8
2.9
3
```

Deskripsi Program:

Tujuan program GO ini adalah aplikasi sederhana yang melakukan iterasi atas bilangan desimal, menambahkan 0.1 pada setiap langkah, hingga mencapai atau melewati angka pembulatan ke atas dari bilangan yang diinput oleh pengguna.

3. Latihan Soal 3

Source Code:

```
latsol3 > go latsol3.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var target, perdonasi, donatur, akumulasi int
7     fmt.Scan(&target)
8     for akumulasi = perdonasi; akumulasi < target; {
9         fmt.Scan(&perdonasi)
10        akumulasi += perdonasi
11        donatur++
12
13        fmt.Printf("Donatur %d: Menyumbang %d. Totalterkumpul: %d\n", donatur, perdonasi, akumulasi)
14    }
15
16    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.",
17        akumulasi, donatur)
18 }
19
```

Output:

```
PROBLEMS OUTPUT TERMINAL PORTS
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\A
ul13\latsol3\latsol3.go"
300
100
Donatur 1: Menyumbang 100. Totalterkumpul: 100
50
Donatur 2: Menyumbang 50. Totalterkumpul: 150
200
Donatur 3: Menyumbang 200. Totalterkumpul: 350
Target tercapai! Total donasi: 350 dari 3 donatur.
PS C:\Users\Faris\Documents\Mata Kuliah\RABU\ALGORITMA PEMROGRAMAN\Modul13> go run "c:\Users\Faris\Documents\Mata Kuliah\RABU\A
ul13\latsol3\latsol3.go"
200
300
Donatur 1: Menyumbang 300. Totalterkumpul: 300
```

Deskripsi Program:

Tujuan program GO ini adalah untuk mensimulasikan penggalangan dana, dengan tujuan menghitung total donasi yang terkumpul hingga mencapai target tertentu. Program ini interaktif dan mencetak detail setiap donasi yang diberikan serta jumlah donatur yang berpartisipasi.

DAFTAR PUSTAKA

Prasti Eko Yunanto, S.T., M.Kom

MODUL PRAKTIKUM 13 – REPEAT-UNTIL ALGORITMA DAN
PEMROGRAMAN 1 S1 INFORMATIKA