

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN
MODUL 13
REPEAT-UNTIL



DISUSUN OLEH:
HISYAM NURDIATMOKO
103112400049
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 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.

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.

Notasi repeat-until memiliki banyak sekali keragaman kata kunci di dalam bahasa pemrograman. Penggunaan repeat-until sebenarnya berasal dari keluarga bahasa pemrograman Pascal. Pada keluarga bahasa pemrograman C/C++ digunakan do-while, sedangkan pada bahasa Go tidak ada instruksi eksplisit untuk repeat-until.

13.3 Implementasi menggunakan Go

Sebagai contoh, misalnya terdapat suatu program yang digunakan untuk mengecek username dan password yang digunakan pengguna ketika login adalah "admin" dan "admin12345".

CONTOH SOAL 1

SOURCE CODE:

```
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 }
```

OUTPUT:

```
● PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
.go"
pagi 3
pagi
pagi
pagi
● PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
.go"
kursi 5
kursi
kursi
kursi
kursi
kursi
○ PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> █
```

Deskripsi program contoh soal 1:

Tujuan program contoh soal 1 adalah program Go yang menerima input kata dan mencetaknya sebanyak jumlah pengulangan yang diinginkan oleh pengguna. Program akan dihentikan ketika jumlah kata yang dicetak mencapai jumlah yang diinginkan oleh pengguna.

Contoh:

No	Masukan	Keluaran
1	Pagi 3	Pagi pagi pagi
2	Kursi 5	Kursi kursi kursi kursi kursi

CONTOH SOAL 2

SOURCE CODE:

```
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 }
```

OUTPUT:

```
• PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
• .go"
17
17 adalah bilangan bulat positif
```

Deskripsi program contoh soal 2:

Tujuan program contoh soal 2 adalah program Go yang meminta pengguna untuk memasukkan bilangan bulat positif. Program akan terus meminta input hingga pengguna memasukkan bilangan bulat positif. Masukan berupa bilangan bulat positif, apabila bukan maka program akan terus meminta masukan hingga bilangan yang diberikan adalah bilangan bulat positif. Keluaran berupa satu baris keluaran yang menunjukkan n bilangan adalah bilangan bulat positif.

Contoh:

No	Masukan	Keluaran
1	-5 -2 -1 0 5	5 adalah bilangan bulat positif
2	17	17 adalah bilangan bulat positif

CONTOH SOAL 3

SOURCE CODE:

```
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 }
```

OUTPUT:

```
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
.go"
5 2
3
1
-1
false
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
.go"
15 3
12
9
6
3
0
true
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
.go"
25 5
20
15
10
5
0
true
```

Deskripsi program contoh soal 3:

Tujuan program contoh soal 3 adalah program Go yang digunakan untuk melakukan pengecekan apakah suatu bilangan merupakan kelipatan dari bilangan lainnya. Masukan terdiri dari dua buah bilangan bulat positif X dan Y. Keluaran terdiri dari perulangan

pengurangan kelipatan dengan hasil akhir boolean yang menyatakan apakah bilangan X merupakan kelipatan dari Y.

Contoh:

No	Masukan	Keluaran
1	5 2	3 1 -1 False
2	15 3	12 9 6 3 0 True
3	25 5	20 15 10 5 0 true

LATIHAN SOAL 1

SOURCE CODE:

```
1  package main
2
3  import "fmt"
4
5  func main() {
6      var angka, hasil int
7      fmt.Scan(&angka)
8      for angka > 0 {
9          hasil++
10         angka = angka / 10
11     }
12     fmt.Print(hasil)
13 }
```

OUTPUT:

```
● PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
tkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding\latsol1\latsol1.go
5
1
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
● tkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding\latsol1\latsol1.go
234
3
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
● tkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding\latsol1\latsol1.go
78787
5
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
● tkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding\latsol1\latsol1.go
1894256
7
```

Deskripsi program latihan soal 1:

Tujuan program latihan soal 1 adalah program 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.

Contoh:

No	Masukan	Keluaran
1	5	1
2	234	3
3	78787	5
4	1894256	7

LATIHAN SOAL 2

SOURCE CODE:

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

OUTPUT:

```
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
sol2.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run
● sol2.go"
2.7
2.8
2.9
3
```

Deskripsi program latihan soal 2:

Tujuan program latihan soal 2 adalah program Go 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.

Contoh:

No	Masukan	Keluaran
1	0.2	0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
2	2.7	2.8 2.9 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, masukan, temp int
7      fmt.Scan(&target)
8      urutan := 0
9      cek := false
10     for !cek {
11         fmt.Scan(&masukan)
12         temp = temp + masukan
13         urutan++
14         fmt.Printf("Donatur %d: menyumbang %d. Total terkumpul: %d\n", urutan, masukan, temp)
15         cek = temp >= target
16     }
17     fmt.Printf("Targer tercapai! Total donasi: %d dari %d donatur.", temp, urutan)
18 }
```

OUTPUT:

```
● PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run sol3.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
Targer tercapai! Total donasi: 350 dari 3 donatur.
PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run sol3.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
Targer tercapai! Total donasi: 600 dari 4 donatur.
● PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> go run sol3.go
200
300
Donatur 1: menyumbang 300. Total terkumpul: 300
Targer tercapai! Total donasi: 300 dari 1 donatur.
○ PS D:\0_Matkul\Algoritma Pemrograman\alpro_week_13\kamis_modul13\coding> 
```

Deskripsi program latihan soal 3:

Tujuan program latihan soal 3 adalah Program Go untuk menghitung jumlah donasi dan donator, program Go ini akan terus meminta input dari pengguna untuk jumlah donasi hingga total donasi mencapai atau melebihi target yang telah ditentukan. Masukan pada baris pertama berupa bilangan bulat yang merupakan target donasi yang harus dicapai. Masukan pada baris berikut dan seterusnya merupakan bilangan bulat yang menyatakan donasi oleh setiap donatur, masukan terus diterima hingga target tercapai. Keluaran berupa bilangan hasil total penjumlahan tiap perulangannya serta jumlah donator.

Contoh :

No	Masukan	Keluaran
1	300 100 50 200	Donatur 1: Menyumbang 100. Total terkumpul: 100 Donatur 2: Menyumbang 50. Total terkumpul: 150 Donatur 3: Menyumbang 200. Total terkumpul: 350 Target tercapai! Total donasi: 350 dari 3 donatur.
2	500 150 100 50 300	Donatur 1: Menyumbang 150. Total terkumpul: 150 Donatur 2: Menyumbang 100. Total terkumpul: 250 Donatur 3: Menyumbang 50. Total terkumpul: 300 Donatur 4: Menyumbang 300. Total terkumpul: 600 Target tercapai! Total donasi: 600 dari 4 donatur.
3	200 300	Donatur 1: Menyumbang 300. Total terkumpul: 300 Target tercapai! Total donasi: 300 dari 1 donatur.

DAFTAR PUSTAKA

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

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