

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN 1**  
**MODUL 13**  
**REPEAT-UNTIL**



**DISUSUN OLEH:**  
**NUFAIL ALAUDDIN TSAQIF**  
**103112400084**  
**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

### 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.

### 2. Karakteristik Repeat-Until

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

2.1 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.

2.2 Kondisi/berhenti, merupakan kondisi berhenti dari perulangan, harus bernilai false selama perulangan dilakukan.

## CONTOH SOAL

### 1. Contoh 1

Source Code:

```
coso1 > go cososs1.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 }
```

Output:

```
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
pagi 3
pagi
pagi
pagi
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
kursi 5
kursi
kursi
kursi
kursi
kursi
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> 
```

Deskripsi Program:

program 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.

Masukan berupa suatu kata dan jumlah pengulangan yang diinginkan oleh pengguna.

Keluaran berupa kata yang diinputkan pengguna dan dicetak sebanyak jumlah pengulangan yang diinginkan oleh pengguna.

## 2. Contoh 2

Source Code:

```
coso2 > go coso2.go > main
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:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13\coso2.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13\coso2.go"
17
17 adalah bilangan bulat positif
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> 
```

Deksripsi Program:

program 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.

### 3. Contoh 3

Source Code:

```
cos03 / go cos03.go > ...
1  package main
2
3  import "fmt"
4
5  func main() {
6      var x, y int
7      var selesai bool
8      fmt.Scan(&x, &y)
9      for selesai = false; !selesai; {
10         x = x - y
11         fmt.Println(x)
12         selesai = x <= 0
13     }
14     fmt.Println(x == 0)
15 }
16
```

Output:

```
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
5
2
3
1
-1
false
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
15
3
12
9
6
-1
false
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
15
3
12
9
6
9
6
6
3
0
true
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
25
5
20
15
10
5
0
true
```

Deksripsi Program:

program 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.

## SOAL LATIHAN

1.

Source Code:

```
latsol1 > go latsol1.go > main
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:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
5
1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
234
3
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
78787
5
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGO
1894256
7
```

Deskripsi 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.

2.

Source Code:

```
latsol2 > -o latsol2.go > main
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:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
2.7
2.8
2.9
3
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> 
```

Deksripsi Program:

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.

3.

Source Coding:

```
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:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
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:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
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:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> go run "d:\ALGORITMA PROGRAMING\PRAKTIKUM
200
300
Donatur 1: menyumbang 300. Total terkumpul: 300
Targer tercapai! Total donasi: 300 dari 1 donatur.
PS D:\ALGORITMA PROGRAMING\PRAKTIKUM\Praktikum_Modul_13> |
```

Deksripsi Program:

Program ini berfungsi untuk menghitung total donasi hingga mencapai target yang ditentukan. Pertama, pengguna memasukkan nilai target donasi.

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 donatur.



## **DAFTAR PUSTAKA**

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

**MODUL PRAKTIKUM 13-REPEAT-UNTIL ALGORITMA DAN PEMOGRAMAN 1  
S1 INFORMATIKA**