

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



DISUSUN OLEH:
ICHYA ULUMIDDIIN
103112400076
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. Pendahuluan

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.

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.

CONTOH SOAL

1. Latihan 1

Buatlah program menggunakan bahasa 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.

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.

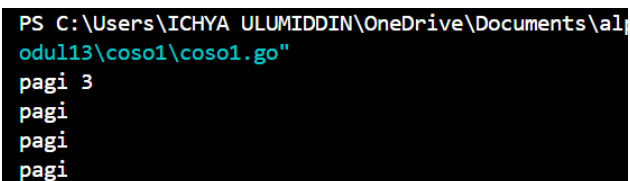
Source Code:

```
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 C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\al...
odul13\coso1\coso1.go
pagi 3
pagi
pagi
pagi
```

Deskripsi Program:

Program ini bertujuan untuk mencetak sebuah kata yang dimasukkan oleh pengguna sebanyak jumlah pengulangan yang diinginkan. Program menerima dua input: kata dan jumlah pengulangan. Menggunakan perulangan for, program mencetak kata hingga jumlah yang ditentukan tercapai, dengan bantuan variabel counter untuk menghitung dan variabel done untuk menghentikan perulangan ketika jumlah pengulangan tercapai. Tujuan utama program ini adalah untuk mengajarkan konsep perulangan (looping) dan kontrol alur dalam pemrograman.

2. Latihan 2

Buatlah program dalam bahasa 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.

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 C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro_
odul13\coso2\coso2.go"
17
17 adalah bilangan bulat positif
```

Deskripsi Program:

Program ini meminta pengguna untuk memasukkan angka dan memeriksa apakah angka tersebut bilangan bulat positif. Perulangan terus berlangsung hingga pengguna memasukkan angka lebih besar dari 0, setelah itu program mencetak pesan bahwa angka tersebut adalah bilangan bulat positif. Tujuan program ini adalah untuk mengajarkan penggunaan perulangan dengan kondisi dan validasi input.

3. Latihan 3

Buatlah 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.

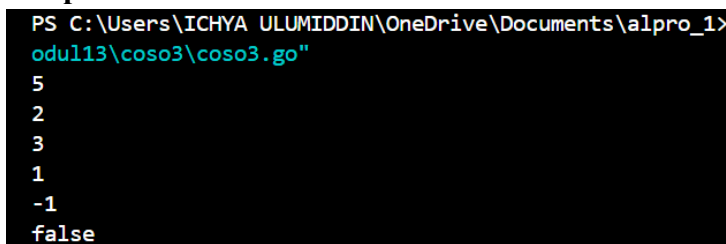
Source Code:

```
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 C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro_1>
odul13\coso3\coso3.go"
5
2
3
1
-1
false
```

Deskripsi Program:

Program ini bertujuan untuk melakukan pengurangan berulang antara dua angka yang dimasukkan pengguna, yaitu x dan y, hingga nilai x menjadi kurang dari atau sama dengan 0. Program menggunakan perulangan for yang terus mengurangi nilai x dengan nilai y dalam setiap iterasi dan mencetak hasil pengurangan tersebut. Perulangan berhenti ketika nilai x menjadi 0 atau negatif, dan program kemudian mencetak apakah nilai x sama dengan 0 setelah perulangan berhenti.

SOAL LATIHAN

- 1) 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 C:\Users\ICHYA ULUMIDDIN\OneDrive\Documents\alpro_
OMework_WEEK13(LAB)\latsol1\latsol1.go"
234
3
```

Deskripsi Program:

Program ini bertujuan untuk menghitung jumlah digit dalam sebuah bilangan bulat yang dimasukkan oleh pengguna. Program menerima input berupa angka, lalu menggunakan perulangan untuk membagi angka tersebut secara berulang dengan 10, yang pada setiap iterasi mengurangi angka tersebut satu digit. Setiap kali angka dibagi, variabel digitCount akan bertambah satu, yang menunjukkan jumlah digit yang telah diproses. Perulangan berhenti ketika angka menjadi 0, dan program mencetak jumlah digit dari angka yang dimasukkan.

- 2) 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:

```
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
```

Deskripsi Program :

Program ini bertujuan untuk mencetak angka desimal mulai dari angka yang dimasukkan pengguna, dengan penambahan 0.1 pada setiap iterasi, hingga nilai tersebut lebih besar dari angka bulat setelah nilai input. Program ini mengajarkan konsep perulangan dengan penambahan bertahap dan pencetakan angka desimal menggunakan tipe data float64.

- 3) Sebuah organisasi amal sedang mengumpulkan donasi untuk mendukung kegiatan sosial mereka. Setiap donatur dapat memberikan sumbangan dalam jumlah tertentu. Program 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 donatur

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:

```
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.
```

Deskripsi Program:

Program ini pertama-tama meminta input berupa angka target donasi yang harus dicapai. Setelah itu, program memasuki sebuah loop yang terus berulang dan meminta input donasi dari setiap donatur satu per satu. Setiap kali donasi diterima, program akan menambahkan jumlah donasi tersebut ke total donasi yang terkumpul dan menghitung jumlah donatur yang sudah memberikan sumbangan.

Setiap kali donasi baru diterima, program akan menampilkan informasi yang mencakup urutan donatur, jumlah donasi yang diberikan, dan total donasi yang terkumpul sampai saat itu. Proses ini terus berlanjut hingga total donasi mencapai atau melebihi target yang telah ditentukan sebelumnya. Setelah target tercapai, program akan keluar dari loop dan menampilkan pesan bahwa target telah tercapai, diikuti dengan total donasi yang terkumpul dan jumlah donatur yang telah berpartisipasi.

DAFTAR PUSTAKA

Amisha. (2023, May 4). *Repeat-Until in Golang*. Retrieved from Scaler:
<https://www.scaler.com/topics/golang/while-loop-in-golang/>