

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



DISUSUN OLEH:
DWI OKTA SURYANINGRUM
103112400066
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

PERULANGAN

Perulangan adalah proses mengulang-ulang eksekusi blok kode tanpa henti, selama kondisi yang dijadikan acuan terpenuhi. Biasanya disiapkan variabel untuk iterasi atau variabel penanda kapan perulangan akan diberhentikan. Di Go keyword perulangan hanya **for** saja, tetapi meski demikian, kemampuannya merupakan gabungan for, foreach, dan while ibarat bahasa pemrograman lain.

Perbedaan while-loop dengan repeat until

1. While-loop adalah kondisi dimana setiap kali memasukkan loop harus ada (benar/true) dan juga jika loop tidak terpenuhi maka akan keluar(salah/false)
2. Sedangkan untuk repeat until adalah kondisi dimana setiap kali looping kondisi harus dilakukan terus menerus sampai berhenti jika saat(benar/true) kalau kondisinya(salah/false) maka perulangan tidak akan terjadi
3. Untuk for loop tidak dapat dimasukkan apabila variabel perulangan tidak dapat di iterasi sehingga tidak dapat mencapai kondisi “i dan kondisi x”

Contoh syntax for-loop, while-loop, dan repeat-until

1. For-loop

```
<int>  
For <x> to <y> do  
    <kondisi>  
Endfor  
<terminasi>
```

2. While-loop

```
<int>  
While <x> do  
    <kondisi>  
Endwhile  
<terminasi>
```

3. Repeat-until

<int>

Repeat

 <ekspresi>

Until <kondisi>

CONTOH SOAL

1. Latihan1

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.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
1.	pagi 3	pagi pagi pagi
2.	kursi 5	kursi kursi kursi kursi kursi

Source Code:

```
package main

import "fmt"

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

```
}  
}
```

Output:

```
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Coso/tempCodeRunnerFile.go"  
pagi 3  
pagi  
pagi  
pagi  
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Coso/tempCodeRunnerFile.go"  
kursi 5  
kursi  
kursi  
kursi  
kursi  
kursi  
mymac@192 praktikum1 %
```

Deskripsi Program:

- Var word mendeklarasikan variabel dengan tipe data string
- Var repetitions mendeklarasikan var dengan tipe data integer
- Fmt.Scan(&word, &repetitions) untuk menginputkan isi dari variabel
- Membuat variabel baru dengan nama counter dan bernilai 0
- Program melakukan perulangan jika done bernilai false
- Dalam perulangan akan menampilkan word
- Counter akan terus bertambah setiap kali loop dijalankan
- Done akan melakukan cek apakah counter sudah mencapai atau melebihi repetitions, jika sudah maka program akan berhenti looping

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.

Contoh masukan dan keluaran:

No	Masukan	Keluaran
----	---------	----------

1.	-5 -2 -1 0 5	5 adalah bilangan bulat positif
2.	17	17 adalah bilangan bulat positif

Source Code :

```
package main

import "fmt"

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

Output:

```
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Coso/coso3.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Coso/coso3.go"
17
17 adalah bilangan bulat positif
○ mymac@192 praktikum1 %
```

Deskripsi Program:

- Var n mendeklarasikan variabel dengan tipe data integer
- Var result mendeklarasikan variabel dengan tipe data Boolean
- for result = true; result; { ... } melakukan perulangan selama result bernilai true
- fmt.Scan(&n) digunakan untuk menginputkan nilai n

- `result = n <= 0` memeriksa apakah nilai `n` kurang dari 0, jika benar maka perulangan akan terus berjalan. Jika salah maka perulangan akan berhenti.

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`.

Contoh masukan dan keluaran:

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

Source Code

```
package main

import "fmt"
```

```

func main() {
    var x, 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:

```

● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Coso/tempCodeRunnerFile.go"
5 2
3
1
-1
false
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Coso/tempCodeRunnerFile.go"
15 3
12
9
6
3
0
true
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Coso/tempCodeRunnerFile.go"
25 5
20
15
10
5
0
true
○ mymac@192 praktikum1 %

```

Deskripsi Program:

- Var x dan y mendeklarasikan 2 buah variabel dengan tipe data integer
- Var selesai mendeklarasikan variabel dengan tipe data Boolean
- Fmt.Scan(&x, &y) digunakan untuk menginputkan nilai dari var x dan y
- For selesai = false; !selesai; {...} melakukan perulangan selama selesai bernilai false
- X = x-y melakukan operasi pengurangan pada setiap iterasi, nilai x dikurangi y dan hasilnya disimpan Kembali di var x.
- Fmt.Println(x) menampilkan nilai var x setelah pengurangan

- **selesai = x <= 0** Program mengecek apakah nilai x lebih kecil atau sama dengan 0:
- Jika benar ($x \leq 0$), maka selesai akan menjadi true, dan perulangan berhenti.
- Jika salah, perulangan akan dilanjutkan.

SOAL LATIHAN

Statement Percabangan

1. Latihan Soal 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.

Contoh masukan dan keluaran

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

Source Code:

```
package main

import "fmt"

func main(){
    var bil, digit int

    digit = 0
    fmt.Scan(&bil)

    for bil > 0{
        bil = bil / 10
        digit++
    }
    fmt.Println(digit)
}
```

Output:

```

mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/tempCodeRunnerFile.go"
1
1
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/tempCodeRunnerFile.go"
234
3
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/tempCodeRunnerFile.go"
78787
5
mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/tempCodeRunnerFile.go"
1894256
7
mymac@192 praktikum1 %

```

Deskripsi Program:

- var bil dan digit mendeklarasikan variabel dengan tipe data integer
- menginisiasikan digit dengan nilai 0
- fmt.Scan(&bil) menginputkan nilai var bil
- for bil>0 akan melakukan perulangan apabila nilai bil lebih dari 0
- bil = bil/10 melakukan operasi pembagian pada setiap iterasi, nilai bil dibagi 10 dan hasilnya disimpan Kembali di var bil.
- **digit++** Setiap kali satu digit dihapus, nilai digit (jumlah digit) ditambah 1.

2. Latihan Soal 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.

Contoh masukan dan keluaran

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

Source Code:

```
package main

import "fmt"

func main() {

    var angka, done float32

    fmt.Scan(&angka)

    done = float32(int(angka)) + 1

    for selesai := false; !selesai; {
        angka += 0.1
        angka = float32(int(angka*10+0.5)) / 10

        if angka >= done {
            fmt.Printf("%.0f\n", angka)
            selesai = true
        } else {
            fmt.Printf("%.1f\n", angka)
        }
    }
}
```

Output:

```

● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/latsol2.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/latsol2.go"
2.7
2.8
2.9
3

```

Deskripsi Program :

- var angka dan done mendeklarasikan 2 variabel dengan tipe data float32
- fmt.Scan(&angka) menginputkan nilai angka
- done = float32(int(angka)) + 1 mengubah var done yang tadinya menggunakan tipe data float32 menjadi int(dibulatkan) dan ditambah 1
- For selesai = false; !selesai; {...} melakukan perulangan selama selesai bernilai false
- Angka += 0.1 digunakan untuk menambahkan nilai **0.1** ke variabel angka di setiap iterasi.
- angka = float32(int(angka*10+0.5)) / 10 Operasi ini digunakan untuk membulatkan angka ke satu desimal.
- if angka >= done = Mengecek apakah nilai angka telah mencapai atau melebihi done (bilangan bulat berikutnya).
- Jika iya, program mencetak nilai angka sebagai bilangan bulat (tanpa desimal) dengan %.0f, kemudian mengubah selesai menjadi true untuk menghentikan perulangan.
- Jika belum, program mencetak nilai angka dengan satu desimal menggunakan %.1f.

3. Latihan Soal 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.

Contoh masukan dan keluaran

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.

Source Code

```
package main

import "fmt"

func main() {
    var target, donasi, total, donatur int
    fmt.Scan(&target)
    total = 0
    donatur = 0

    for {
        fmt.Scan(&donasi)
        total += donasi
```

```

        donatur++

        fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n",
donatur, donasi, total)

        if total >= target {
            break
        }
    }

    // Tampilkan hasil akhir setelah target tercapai
    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n",
total, donatur)
}

```

Output:

```

● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/lats
ol3.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.
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/lats
ol3.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.
● mymac@192 praktikum1 % go run "/Users/mymac/praktikum1/Modul13/Latsol/lats
ol3.go"
200
300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.
○ mymac@192 praktikum1 %

```

Deskripsi Program :

- var target, donasi, total, donator mendeklarasikan variabel dengan tipe data integer
- fmt.Scan(&target) menginputkan nilai dari variabel target

- menginisiasikan total dan donator dengan nilai 0
- for perulangan akan terus berjalan hingga kondisi berhenti dengan (break)
- `fmt.Scan(&donasi)` menginputkan nilai dari var donasi
- `total += donasi`: Total donasi terkini diperbarui dengan jumlah donasi yang baru diterima.
- Jumlah donatur yang sudah memberikan donasi ditambah satu. Di setiap perulangan
- `fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", donatur, donasi, total)`: Program mencetak informasi setiap kali seorang donatur memberikan donasi, mencantumkan nomor donatur, jumlah donasi, dan total donasi yang terkumpul.
- `if total >= target { break }`: Jika total donasi sudah mencapai atau melebihi target, maka perulangan dihentikan dengan break.
- Program melompat keluar dari loop karena tidak perlu menerima lebih banyak donasi setelah mencapai target.
- `fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", total, donatur)`: Setelah perulangan selesai, program mencetak informasi bahwa target tercapai, bersama dengan jumlah total donasi dan jumlah donatur.

DAFTAR PUSTAKA

Perbedaan while-loop dengan repeat until - While-loop adalah kondisi dimana setiap kali memasukkan. (n.d.). Retrieved from <https://www.studocu.id/id/document/universitas-telkom/algorithm-and-programming/perbedaan-while-loop-dengan-repeat-until/31410481>