

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



DISUSUN OLEH:
SHEILA STEPHANIE ANINDYA
103112400086
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

I. Paradigma Perulangan

Perulangan adalah struktur kontrol yang memungkinkan eksekusi instruksi yang sama dilakukan secara berulang, sehingga menghindari penulisan instruksi berulang secara manual.

Penggunaan repeat-until mirip dengan while-loop, yaitu perulangan yang bergantung pada kondisi tertentu. Perbedaan utama terletak pada cara mendefinisikan kondisi: pada while-loop, kondisi yang didefinisikan adalah kapan perulangan terjadi, sedangkan pada repeat-until, kondisi yang didefinisikan adalah kapan perulangan berhenti.

Kondisi perulangan dan kondisi berhenti saling terkait sebagai komplemen. Artinya, jika kita mengetahui kondisi perulangan, kita dapat mengubahnya menjadi kondisi berhenti dengan menambahkan operator negasi (NOT), dan sebaliknya, kondisi berhenti dapat diubah menjadi kondisi perulangan dengan cara yang sama.

II. Repeat – Until

Perulangan Repeat Until adalah perulangan yang digunakan untuk mengulang statement-statement atau blok statement sampai (Until) kondisi yang diseleksi di Until tidak terpenuhi. Bila kondisi pada pernyataan UNTIL masih terpenuhi, proses pengulangan akan dilakukan lagi.

Contoh :

- a) Statement repeat-until: *"Menulis teks tertentu **sampai** tinta pena habis"*.
Komplemen dari kondisi "tinta pena masih ada" adalah "tinta pena habis".
- b) Statement repeat-until: "Saya makan suap demi suap sampai saya merasa kenyang".
Komplemen dari kondisi "saya masih lapar" adalah. "saya merasa kenyang".

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

1. Latihan1

Source Code:

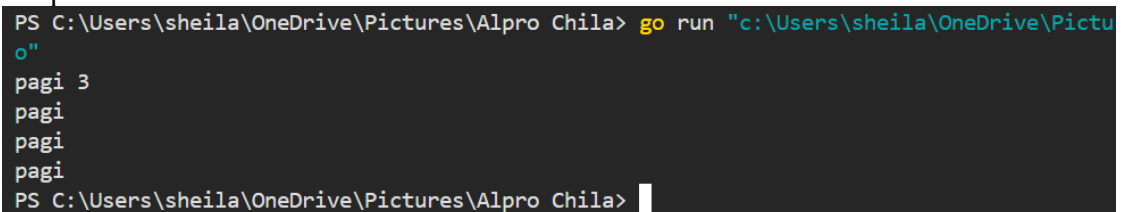
```
package main

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\sheila\OneDrive\Pictures\Alpro Chila> go run "c:\Users\sheila\OneDrive\Pictu
o"
pagi 3
pagi
pagi
pagi
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> |
```

Deskripsi Program:

Program ini menerima dua input, kemudian mencetak sebuah kata sebanyak jumlah pengulangan yang ditentukan.

2. Latihan2

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\sheila\OneDrive\Pictures\Alpro Chila> go run "c:\Users\sheila\OneDrive\Pictures\Alpro Chila\main.go"  
-5  
17  
17 adalah bilangan bulat positif  
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> |
```

Deskripsi Program:

Program ini meminta pengguna untuk memasukkan angka sampai pengguna memasukkan bilangan bulat positif. Setelah itu, program akan mencetak bilangan tersebut dengan menyatakan bahwa bilangan tersebut bulat positif.

3. Latihan3

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\sheila\OneDrive\Pictures\Alpro Chila> go run "c:\Users\sheila\OneDrive\Pictures\Alpro Chila\
o"
5 2
3
1
-1
false
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> go run "c:\Users\sheila\OneDrive\Pictures\Alpro Chila\
o"
15 3
12
9
6
3
0
true
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> 
```

Deskripsi Program:

Program ini membaca dua bilangan bulat x dan y dari input, lalu mengurangi nilai x dengan y secara berulang hingga x menjadi nol atau negatif.

SOAL LATIHAN

Statement perulangan

1. 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.

Source Code:

```
package main

import "fmt"

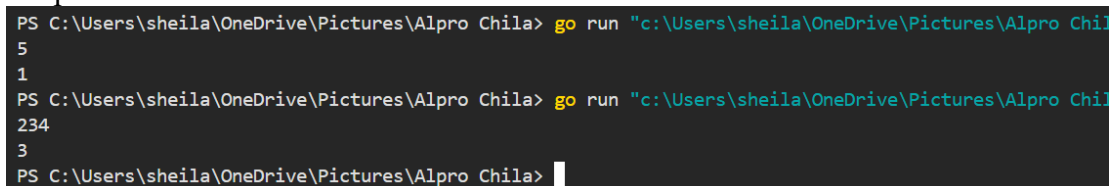
func main() {
    var n int
    fmt.Scan(&n)

    if n <= 0 {
        fmt.Println("harus bilangan bulat positif.")
        return
    }

    jumlahDigit := 0
    for n > 0 {
        jumlahDigit++
        n = n / 10
    }

    fmt.Println(jumlahDigit)
}
```

Output:



```
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> go run "c:\Users\sheila\OneDrive\Pictures\Alpro Chila\
5
1
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> go run "c:\Users\sheila\OneDrive\Pictures\Alpro Chila\
234
3
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> 
```

Deskripsi Program:

Program ini menghitung jumlah digit dari sebuah bilangan bulat positif.

Statement perulangan

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.

Source Code:

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var desimal float64

    fmt.Scan(&desimal)

    bulat := math.Ceil(desimal)

    for {
        fmt.Printf("%.1f\n", desimal)
        desimal += 0.1

        if desimal > bulat {
            break
        }
    }
}
```

Output:



```
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> go run "c:\Users\sheila\OneDrive\Pictures\Alpro Chila\Laparak 1"
0.2
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila>
```

Deskripsi Program:

Program ini mencetak angka inputan mulai dari nilai desimal hingga mencapai angka bulat terdekat yang lebih besar atau sama dengan angka desimal yang telah dibulatkan.

Statement perulangan

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.

Source Code:

```
package main

import "fmt"

func main() {
    var target, donasi, totalDonasi, donatur int

    fmt.Print("target donasi: ")
    fmt.Scan(&target)

    for totalDonasi < target {
        donatur++

        fmt.Printf("donasi dari donatur %d: ", donatur)
        fmt.Scan(&donasi)

        totalDonasi += donasi
        fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", donatur,
donasi, totalDonasi)
    }

    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", totalDonasi,
donatur)
}
```

Output:

```
Donatur 1: Menyumbang 150. Total terkumpul: 150
donasi dari donatur 2: 100
Donatur 2: Menyumbang 100. Total terkumpul: 250
donasi dari donatur 3: 50
Donatur 3: Menyumbang 50. Total terkumpul: 300
donasi dari donatur 4: 300
Donatur 4: Menyumbang 300. Total terkumpul: 600
Target tercapai! Total donasi: 600 dari 4 donatur.
PS C:\Users\sheila\OneDrive\Pictures\Alpro Chila> |
```

Deskripsi Program:

Program ini menghitung total donasi yang terkumpul hingga mencapai target donasi tertentu.

DAFTAR PUSTAKA

1. Prezi. (n.d.). *Repeat-Until*.
https://prezi.com/jxifye4-p6k_/repeat-until/#:~:text=Perulangan%20Repeat%20Until%20adalah%20perulangan,proses%20pengulangan%20akan%20dilakukan%20lagi.
2. Telkom University. *Modul 13 Praktikum Algoritma dan Pemrograman*. Fakultas Informatika, Telkom University, 2024.