

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 12
“REPEAT-UNTIL”



DISUSUN OLEH:
Keishin Naufa Alfaridzhi
103112400061
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

A. Bahasa Yang Digunakan

Pada praktikum ini bahasa pemrograman yang digunakan adalah bahasa pemrograman Go, sesuai dengan modul yang menjadi acuan praktikum. Golang (atau Go) adalah bahasa pemrograman baru, yang mulai dilirik oleh para developer karena kelebihan-kelebihan yang dimilikinya. Sudah banyak Perusahaan besar yang menggunakan bahasa ini untuk produk-produk mereka hingga di level production.

B. Komentar

Komentar biasa dimanfaatkan untuk menyisipkan catatan pada kode program, menulis penjelasan atau deskripsi mengenai suatu blok kode, atau bisa juga digunakan untuk me-remark kode (men-non-aktifkan kode yang tidak digunakan). Komentar akan diabaikan Ketika kompilasi maupun eksekusi program.

Ada 2 jenis komentar di Golang, yaitu inline dan multiline.

1. Komentar Inline

Penulisan komentar jenis ini diawali dengan tanda *double slash* (*//*) lalu diikuti pesan komentarnya. Komentar inline hanya berlaku untuk satu baris pesan saja. Jika pesan komentar lebih dari satu baris, maka tanda *double slash* harus ditulis lagi di baris selanjutnya.

2. Komentar Multiline

Komentar yang cukup panjang akan lebih rapi jika ditulis menggunakan teknik komentar multiline. Ciri dari komentar jenis ini adalah penulisannya diawali dengan tanda *(/**) dan diakhiri *(*/)*.

C. Variabel

Golang mengadopsi 2 jenis penulisan variabel, yang dituliskan tipe data-nya dan yang tidak. Kedua cara tersebut intinya adalah sama, pembedanya hanyalah cara penulisannya saja. Untuk penulisan variabel dengan tipe data, keyword *var* digunakan untuk deklarasi variabel kemudian diakhiri dengan tipe data misalnya *string*. Kemudian untuk penulisan variabel tanpa tipe data, variabel dideklarasikan dengan menggunakan metode type inference. Penandanya tipe data tidak dituliskan pada saat deklarasi. Pada penggunaan metode ini, operand (*=*) harus diganti dengan (*:=*) dan keyword *var* dihilangkan.

Golang memiliki aturan unik yang tidak dimiliki bahasa lain, yaitu tidak boleh ada satupun variabel yang menganggur. Artinya, semua variabel yang dideklarasikan harus digunakan. Jika terdapat variabel yang tidak digunakan tapi dideklarasikan, program akan gagal dikompilasi. Untuk mengatasi itu, golang memiliki variabel yaitu underscore. Underscore (_) adalah predefined variabel yang bisa dimanfaatkan untuk menampung nilai yang tidak dipakai.

D. Tipe Data

Golang mengenal beberapa jenis tipe data, diantaranya adalah tipe data numerik (decimal dan non-desimal), string, dan boolean.

1. Tipe Data Numerik Non-Desimal (uint, int)
2. Tipe Data Numerik Desimal (float64, float32)
3. Tipe Data Bool (true, false)
4. Tipe Data String (string, “ “)

E. Operator Aritmatika

Operator aritmatika merupakan operator yang digunakan untuk operasi yang sifatnya perhitungan. Golang mendukung beberapa operator aritmatika standar, yaitu:

1. Penjumlahan (+)
2. Pengurangan (-)
3. Perkalian (*)
4. Pembagian (/)
5. Modulus atau sisa hasil pembagian (%)

F. Seleksi Kondisi

Seleksi kondisi pada program berguna untuk mengontrol sebuah blok kode yang akan dieksekusi. Yang dijadikan acuan oleh seleksi kondisi adalah nilai bertipe bool, bisa berasal dari variabel, ataupun hasil operasi perbandingan. Nilai tersebut menentukan blok kode mana yang akan dieksekusi. Go memiliki 2 macam keyword untuk seleksi kondisi, yaitu if else dan switch.

1. If Expression

If adalah salah satu kata kunci yang digunakan dalam percabangan. Percabangan artinya kita bisa mengeksekusi kode program tertentu ketika suatu kondisi terpenuhi. Hampir semua bahasa pemrograman mendukung if expression.

2. Else if expression

Terkadang kita butuh membuat beberapa kondisi. Kasus seperti ini dapat menggunakan else if expression. If mendukung short statement sebelum kondisi.

Hal ini sangat cocok untuk membuat statement yang sederhana sebelum melakukan pengecekan terhadap kondisi.

3. Switch-Case

Switch merupakan seleksi kondisi yang sifatnya fokus pada satu variabel, lalu kemudian di-cek nilainya. Contoh sederhananya seperti penentuan apakah nilai variabel x adalah: 1, 2, 3, atau lainnya. Perlu diketahui, switch pada pemrograman Go memiliki perbedaan dibanding bahasa lain. Di Go, ketika sebuah case terpenuhi, tidak akan dilanjutkan ke pengecekan case selanjutnya, meskipun tidak ada keyword “break” di situ. Konsep ini berkebalikan dengan switch pada umumnya pemrograman lain (yang ketika sebuah case terpenuhi, maka akan tetap dilanjutkan mengecek case selanjutnya kecuali ada keyword “break”).

G. Perulangan

Perulangan merupakan proses mengulang dan mengeksekusi blok kode tanpa henti sesuai dengan kondisi yang dijadikan acuan. Biasanya disiapkan variabel untuk iterasi atau penanda kapan perulangan akan dihentikan.

a. For Loop

For loop merupakan statement perulangan dasar dan cukup sering ditemui. Format for loop yaitu sebagai berikut.

- *Init Statement*: bagian ini akan dieksekusi sebelum perulangan dimulai. Biasanya diisi dengan mendeklarasi variabel iterasi.
- *Condition Expression*: bagian ini akan dicek dan dieksekusi setiap perulangan yang dilakukan, jika true maka perulangan akan terus berjalan hingga kondisi bernilai false.
- *Post Statement*: statement ini akan dieksekusi pada akhir iterasi. Jika terdapat range, maka perulangan akan dieksekusi untuk setiap item pada range.

b. While Loop

While loop merupakan perulangan yang akan terus berjalan hingga suatu kondisi terpenuhi. Penulisan while loop adalah dengan menuliskan kondisi setelah keyword for (hanya kondisi). Deklarasi dan iterasi variabel counter tidak dituliskan setelah keyword, hanya kondisi perulangan saja. Konsepnya mirip seperti while milik bahasa pemrograman lain.

c. Repeat Until

Untuk Repeat Until ini mirip seperti for loop biasa namun hanya menggunakan inisiasi dan kondisi saja.

CONTOH SOAL

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 (
        kata string
        x    int
    )

    fmt.Scan(&kata, &x)
    counter := 0
    for done := false; !done; {
        fmt.Println(kata)
        counter++
        done = (counter >= x)
    }
}
```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\coso\satu\satu.go"
pagi 3
pagi
pagi
pagi

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\coso\satu\satu.go"
kursi 5
kursi
kursi
kursi
kursi
kursi
```

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 x int
    for cout := true; cout; {
        fmt.Scan(&x)
        cout = x <= 0
    }
    fmt.Printf("%d adalah bilangan bulat positif\n", x)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\coso\dua\dua.go"
-5
-2
-1
0
5
5 adalah bilangan bulat positif

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\coso\dua\dua.go"
17
17 adalah bilangan bulat positif
```

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, y int
    var done bool
    fmt.Scan(&x, &y)
    for done = false; !done; {
        x = x - y
    }
}
```

```
    fmt.Println(x)
    done = x <= 0
}
fmt.Println(x == 0)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\coso\tiga\tempCodeRunnerFile.go"
5
2
3
1
-1
false

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\coso\tiga\tempCodeRunnerFile.go"
15
3
12
9
6
3
0
true

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\coso\tiga\tempCodeRunnerFile.go"
25 5
20
15
10
5
0
true
```

LATIHAN SOAL

1. Latihan1

Source Code:

```
package main
import "fmt"

func main() {
    var x, next, i int64
    next = 1
    fmt.Scan(&x)
    for process := true; process; {
        next = next * 10
        process = next < x
        i = i + 1
    }
    fmt.Println(i)
}
```


Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\satu\satu.go"
5
1

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\satu\satu.go"
234
3

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\satu\satu.go"
78787
5

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\satu\satu.go"
1894256
7
```

Deskripsi Program:

Program untuk cek panjang dari suatu bilangan. Menggunakan 3 variabel integer 64-bit yaitu x, next dan i. Pertama-tama deklarasi value dari variabel next yaitu 1, lalu terdapat input untuk bilangan yaitu input variabel x. Masuk ke dalam perulangan repeat-until dengan kondisi selama boolean dari process bernilai true maka perulangan akan terus dilakukan, Di dalam perulangan terdapat rumus untuk variabel next yaitu $\text{next} = \text{next} * 10$, lalu boolean dari process bernilai true selama value $\text{next} < x$, dan setelahnya terdapat variabel I untuk increment yang akan bertambah satu tiap perulangan.

Cara kerja program ini yaitu mengandalkan nilai dari next yang mana perulangan akan terus berjalan selama nilai next kurang dari nilai x. Pada setiap perulangan nilai next akan dikali 10 untuk menjadikan nilai next yang berawal dari satuan, ratusan, ribuan, dan seterusnya hingga nilainya lebih dari x yang akan membuat perulangan berhenti. Kemudian perulangan akan dicatat telah dijalankan berapa kali oleh variabel i, nilai nya akan sama dengan panjang bilangan.

2. Latihan2

Source Code:

```
package main
import "fmt"
func main() {
    var x float64
    var itg int
    fmt.Scan(&x)
    itg = int(x) + 1
    for done := false; !done; {
        x = x + 0.1
        fmt.Printf("%.1f\n", x)
        done = x > float64(itg)-0.11
    }
    fmt.Println(itg)
```

```
}
```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\dua\dua.go"
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\dua\dua.go"
2.7
2.8
2.9
3
```

Deskripsi Program:

Program untuk menambahkan mendapatkan bilangan bulat optimal dari bilangan yang diinputkan. Menggunakan 1 variabel bertipe data float64 yaitu **x**, dan 1 variabel bertipe data integer yaitu **itg**. Pertama kita akan input nilai dari x, setelah itu pada variabel itg dilakukan casting untuk mendapatkan nilai integer dari x kemudian ditambah 1. Pada perulangan dengan kondisi variabel done adalah false, dilakukan penambahan nilai x sebanyak 0.1 tiap perulangan hingga memenuhi kondisi done adalah true yaitu ketika " $x > \text{float64}(\text{itg}) - 0.11$ " (kita lakukan casting untuk mengubah nilai integer pada itg menjadi float64). Di luar perulangan kita cetak nilai dari itg.

Pada kondisi variabel done mengapa dikurangi 0.11 dan tidak 0.1? Karena tipe data float nilainya tidak tetap, untuk berjaga-jaga kita tetap lebihkan angka di belakang desimal nya karena terdapat kondisi tertentu jika hanya dikurangi dengan 0.1 saja maka perulangan berjalan lebih dari yang kita tentukan pada kondisi variabel done. Kita hanya ingin mencetak nilai x hingga 0.9 saja dalam perulangan, lalu akan dilakukan cetak di luar perulangan untuk nilai bilangan bulatnya yakni print variabel itg.

3. Latihan3

Source Code:

```
package main
import "fmt"

func main() {
    var target, perdonasi, donatur, akumulasi int
    fmt.Scan(&target)
    for done := false; !done; {
        fmt.Scan(&perdonasi)
        akumulasi += perdonasi
        done = akumulasi >= target
    }
}
```

```

        donatur++
        fmt.Printf("Donatur %d: Menyumbang %d. Total terkumpul: %d\n", donatur,
perdonasi, akumulasi)
    }
    fmt.Printf("Target tercapai! Total donasi: %d dari %d donatur.\n", akumulasi,
donatur)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\tiga\tiga.go"
300
100
Donatur 1: Menyumbang 100. Total terkumpul: 100
50
Donatur 2: Menyumbang 50. Total terkumpul: 150
200

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\tiga\tiga.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.

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul13\latso\tiga\tiga.go"
200
300
Donatur 1: Menyumbang 300. Total terkumpul: 300
Target tercapai! Total donasi: 300 dari 1 donatur.

```

Deskripsi Program:

Program untuk mengumpulkan donasi berdasarkan target yang ditentukan. Menggunakan 4 variabel integer yaitu target, perdonasi, donatur dan akumulasi. Pertama kita akan tentukan target dengan menginputkan nilai untuk variabel target. Lalu masuk ke dalam perulangan dengan kondisi “done” bernilai true jika akumulasi sudah melebihi target. Di dalam perulangan kita input nilai donasi, kemudian memasukkan nilai donasi ke dalam variabel akumulasi, lalu menambahkan index variabel donatur untuk mendapatkan donatur ke-(donatur), setelah itu dilakukan print untuk donatur pertama, jika nilai "done masih false, perulangan akan terus berlanjut hingga nilai done adalah true. Setelah perulangan berhenti akan dicetak target tercapai dengan jumlah donasi dan jumlah donatur.

DAFTAR PUSTAKA

Noval Agung Prayogo. *Dasar Pemrograman Golang*. Diakses pada 01 Oktober 2024.
<https://dasarpemrogramangolang.novalagung.com>

Annisa Nur Isnaeni. *Golang — Seleksi Kondisi*. Diakses pada 01 Oktober 2024.
<https://medium.com/@annisaisna/golang-seleksi-kondisi-f988ead004b4>

Parvez Alam, *Golang for loop example | Golang Loops Tutorial – Phpflow.com*
<https://medium.com/@parvez1487/golang-for-loop-example-golang-loops-tutorial-phpflow-com-f4b2b0e57944>