

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 10
“ELSE - IF”



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.

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.

CONTOH SOAL

1. Buatlah program dengan bahasa Go yang digunakan untuk menentukan apakah seorang penduduk bisa membuat KTP atau tidak. Syarat utama membuat KTP adalah berusia minimal 17 tahun dan memiliki kartu keluarga.

Masukan terdiri dari dua baris. Baris pertama adalah bilangan bulat yang menyatakan usia, sedangkan baris kedua adalah boolean yang menyatakan memiliki KTP atau tidak.

Keluaran berupa teks "bisa membuat KTP" atau "belum bisa membuat KTP" sesuai dengan masukan yang diberikan.

Source Code:

```
package main
import "fmt"

func main() {
    var umur int
    var kk bool
    fmt.Scan(&umur, &kk)
    if umur >= 17 && kk {
        fmt.Println("Bisa buat ktp")
    } else {
        fmt.Println("Belum bisa buat ktp")
    }
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso1\coso1.go"
17 true
Bisa buat ktp

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso1\coso1.go"
20 false
Belum bisa buat ktp

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso1\coso1.go"
15 true
Belum bisa buat ktp
```

2. Buatlah suatu program Go yang digunakan untuk menentukan suatu alfabet yang diberikan adalah vokal atau konsonan.

Masukan terdiri dari suatu huruf.

Keluaran berupa teks yang menyatakan huruf adalah "vokal" atau "konsonan" atau "bukan huruf".

Source Code:

```
package main

import "fmt"

func main() {
    var x rune
    var huruf, vKecil, vBesar bool
    fmt.Scanf("%c", &x)
    huruf = (x >= 'a' && x <= 'z') || (x >= 'A' && x <= 'Z')
    vKecil = (x == 'a' || x == 'i' || x == 'u' || x == 'e' || x == 'o')
    vBesar = (x == 'A' || x == 'I' || x == 'U' || x == 'E' || x == 'O')
    if huruf && (vKecil || vBesar) {
        fmt.Println("vokal")
    } else if huruf && !(vKecil || vBesar) {
        fmt.Println("konsonan")
    } else {
        fmt.Println("bukan huruf")
    }
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso2\coso2.go"
A
vokal

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso2\coso2.go"
f
konsonan

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso2\coso2.go"
1
bukan huruf

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso2\coso2.go"
$
bukan huruf
```

3. Buatlah program untuk menentukan apakah digit dalam suatu bilangan terurut membesar, mengecil, atau tidak terurut. Bilangan hanya terdiri dari empat digit saja, atau lebih besar atau sama dengan 1000 dan kecil sama atau sama dengan 9999.

Source Code:

```
package main

import "fmt"

func main() {
    var x, d1, d2, d3, d4 int
    var teks string
    fmt.Print("Bilangan : ")
```

```

    fmt.Scan(&x)
    d4 = x % 10
    d3 = (x % 100) / 10
    d2 = (x % 1000) / 100
    d1 = x / 1000

    if d1 < d2 && d2 < d3 && d3 < d4 {
        teks = "terurut membesar"
    } else if d1 > d2 && d2 > d3 && d3 > d4 {
        teks = "terurut mengecil"
    } else {
        teks = "tidak terurut"
    }

    fmt.Println("Digit pada bilangan", x, teks)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso3\tempCodeRunnerFile.go"
Bilangan : 2489
Digit pada bilangan 2489 terurut membesar

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso3\tempCodeRunnerFile.go"
Bilangan : 3861
Digit pada bilangan 3861 tidak terurut

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\coso3\tempCodeRunnerFile.go"
Bilangan : 9651
Digit pada bilangan 9651 terurut mengecil

```

LATIHAN SOAL

1. Latihan1

Source Code:

```

package main

import "fmt"

func main() {
    var beratParsel, kg, gr, biayaKg, biayaGr, totalBiaya int
    fmt.Scan(&beratParsel)
    kg = beratParsel / 1000
    gr = beratParsel % 1000
    if gr >= 500 {
        biayaGr = gr * 5
    } else {
        biayaGr = gr * 15
    }

    biayaKg = kg * 10000
    if kg > 10 {

```

```

        totalBiaya = biayaKg
    } else {
        totalBiaya = biayaKg + biayaGr
    }
    fmt.Printf("Berat parcel (gram): %v \nDetail berat: %v kg + %v gr \nDetail biaya:
Rp. %v + Rp. %v \nTotal biaya: Rp. %v", beratParsel, kg, gr, biayaKg, biayaGr,
totalBiaya)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\satu\satu.go"
8500
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\satu\satu.go"
9250
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\satu\satu.go"
11750
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 110000

```

Deskripsi Program:

Program untuk menghitung biaya jasa kirim parcel dengan satuan berat gram. Alurnya adalah pertama input berat parcel dalam gram. Masuk logic pertama kita perlu mendefinisikan variable kg (karena akan digunakan untuk biaya per-kg) dan variable gr (untuk mengambil sisa gram) kemudian melakukan cek jika kondisi gr lebih dari 500 maka biayaGr = gr * 5, jika kondisi gr kurang dari 500 maka biayaGr = gr * 15. Kedua, akan mendefinisi biayaKg untuk mencari biaya kirim per-kg. Kemudian cek jika paket lebih dari 10kg maka tidak ada tambahan biaya sisa gram parcel. Lalu output detail berat, detail biaya, dan total biaya.

2. Latihan2

Source Code (Sebelum Diperbaiki):

```

package main
import "fmt"

func main() {
    var nam float64
    var nmk string

    fmt.Print("Nilai akhir mata kuliah: ")
}

```



```

    fmt.Scan(&nam)

    if nam >= 80 {
        nam = "A"
    }
    if nam > 72.5 {
        nam = "AB"
    }
    if nam > 65 {
        nam = "B"
    }
    if nam > 57.5 {
        nam = "BC"
    }
    if nam > 50 {
        nam = "C"
    }
    if nam > 40 {
        nam = "D"
    } else if nam <= 40 {
        nam = "E"
    }
    }

    fmt.Println("Nilai mata kuliah:", nmk)
}

```

Source Code (Setelah Diperbaiki):

```

package main

import "fmt"

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scan(&nam)
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "AB"
    } else if nam > 65 {
        nmk = "B"
    } else if nam > 57.5 {
        nmk = "BC"
    } else if nam > 50 {
        nmk = "C"
    } else if nam > 40 {
        nmk = "D"
    }
}

```

```
} else {  
    nmk = "E"  
}  
fmt.Println("Nilai mata kuliah: ", nmk)  
}
```

- a. Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?

Jawab:

Program error dan tidak dapat dieksekusi / dijalankan

- b. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!

Jawab:

Pertama, pada percabangan hanya digunakan “if” saja yang akan membuat suatu kondisi untuk suatu kasus akan terpisah-pisah sehingga yang dieksekusi adalah blok kode “if” paling terakhir/bawah. Seharusnya “else if” digunakan untuk kasus ini karena terdapat beberapa kondisi. Kedua, variabel yang digunakan pada blok kode percabangan adalah variable nam yang mana memiliki tipe data float64 sedangkan yang dibutuhkan untuk cetak adalah tipe data string. Lalu satu variable (dalam kasus ini nam) tidak dapat memiliki tipe data selain yang sebelumnya sudah dideklarasikan (dalam kasus ini tipe data float64) kecuali dengan melakukan casting. Seharusnya variable yang ada dalam blok kode percabangan if pada soal ini adalah variable nmk yang memiliki tipe data string.

- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5.

Seharusnya keluaran yang diperoleh adalah ‘A’, ‘B’, dan ‘D’.

Output program setelah diperbaiki:

```
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\dua\dua.go"  
Nilai akhir mata kuliah: 93.5  
Nilai mata kuliah: A  
  
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\dua\dua.go"  
Nilai akhir mata kuliah: 70.6  
Nilai mata kuliah: B  
  
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\dua\dua.go"  
Nilai akhir mata kuliah: 49.5  
Nilai mata kuliah: D
```

3. Latihan3

Source Code:

```
package main
import "fmt"

func main() {
    var bil, factor int
    var prime bool
    fmt.Print("Bilangan: ")
    fmt.Scan(&bil)
    fmt.Print("Faktor: ")
    for i := 1; i <= bil; i++ {
        if bil%i == 0 {
            fmt.Printf("%v ", i)
            factor++
        }
    }

    if factor == 2 {
        prime = true
    }
    fmt.Printf("\nPrima: %v", prime)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\tiga\tiga.go"
Bilangan: 7
Faktor: 1 7
Prima: true
D:\Tugas\Alpro\Praktikum\modul10>go run "d:\Tugas\Alpro\Praktikum\modul10\latso\tiga\tiga.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
```

Deskripsi Program:

Program mencari faktor dari suatu bilangan dan menentukan bilangan prima. Variabel bil dan factor dengan tipe data integer dan variabel prime dengan tipe data bool. Menggunakan perulangan “for” dengan inisiasi i = 1 dan kondisi i <= bil. Didalamnya melakukan pengecekan jika bil%i == 0, maka print i saat ini (setiap perulangan nilai i akan bertambah 1 karena i++ atau increment) yang mana itu akan menjadi kumpulan faktor dari sebuah bilangan. Kemudian terdapat increment factor, value nya akan bertambah setiap kondisi terpenuhi. Lalu pada cek prima kita akan cek value dari factor tadi, jika factor == 2 maka bilangan tersebut adalah bilangan prima.

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>