

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1
MODUL 16
“SKEMA PEMROSESAN SEKUENSIAL”



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.

LATIHAN SOAL

1. Latihan no. 1

Source Code:

```
package main
import "fmt"

func main() {
    var bil, jumlah, rerata, i float64
    for {
        fmt.Scan(&bil)
        if bil == 9999 {
            break
        }
        i = i + 1
        jumlah = jumlah + bil
    }
    rerata = jumlah / i
    fmt.Println(rerata)
}
```

Output:

```
D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul16\latso\satu\satu.go"
2 5 8 19 27 59 80
9999
28.571428571428573
```

Deskripsi Program:

Program untuk mencari rata-rata dari n bilangan. Menggunakan while loop untuk melakukan perulangan terus menerus, dalam perulangan terdapat kondisi jika input = 9999 maka perulangan berhenti. Kumpulan bilangan yang diinputkan akan dicatat dalam variabel i yang akan bertambah terus setiap perulangan. Bilangan yang diinputkan disimpan dalam variabel jumlah. Lalu penghitungan rata-rata dilakukan diluar perulangan dengan rumus $\text{rerata} = \text{jumlah} / i$.

2. Latihan no. 3

Source Code:

```
package main
import (
    "fmt"
    "math/rand"
)

func main() {
    const ukuranTetesan = 0.0001
    var jumlahTetesan, daerahA, daerahB, daerahC, daerahD int
    var curahHujanA, curahHujanB, curahHujanC, curahHujanD float64
```

```

fmt.Print("Masukkan jumlah tetesan air hujan: ")
fmt.Scan(&jumlahTetesan)

for i := 0; i < jumlahTetesan; i++ {
    x := rand.Float64()
    y := rand.Float64()
    if x <= 0.5 && y <= 0.5 {
        daerahA++
    } else if x > 0.5 && y <= 0.5 {
        daerahB++
    } else if x <= 0.5 && y > 0.5 {
        daerahC++
    } else {
        daerahD++
    }
}

curahHujanA = float64(daerahA) * ukuranTetesan
curahHujanB = float64(daerahB) * ukuranTetesan
curahHujanC = float64(daerahC) * ukuranTetesan
curahHujanD = float64(daerahD) * ukuranTetesan

fmt.Printf("Curah hujan daerah A: %.4f milimeter\n", curahHujanA)
fmt.Printf("Curah hujan daerah B: %.4f milimeter\n", curahHujanB)
fmt.Printf("Curah hujan daerah C: %.4f milimeter\n", curahHujanC)
fmt.Printf("Curah hujan daerah D: %.4f milimeter\n", curahHujanD)
}

```

Output:

```

D:\Tugas\Alpro\Praktikum>go run "d:\Tugas\Alpro\Praktikum\modul16\latso\tiga\tiga.go"
Masukkan jumlah tetesan air hujan: 10000000
Hasil Curah Hujan per Daerah:
Curah hujan daerah A: 249.8532 milimeter
Curah hujan daerah B: 250.0991 milimeter
Curah hujan daerah C: 249.9676 milimeter
Curah hujan daerah D: 250.0801 milimeter

```

Deskripsi Program:

Program yang menerima input berupa banyaknya tetesan air hujan. Simpan value ukuranTetesan dalam konstanta; deklarasi variabel bertipe data integer yaitu jumlahTetesan, daerahA, daerahB, daerahC, daerahD; deklarasi variabel bertipe data float64 yaitu curahHujanA, curahHujanB, curahHujanC, curahHujanD. Pertama kita perlu inputkan terlebih dahulu nilai dari jumlahTetesan, setelah itu masuk ke dalam perulangan yang akan dilakukan sebanyak jumlahTetesan. Dalam perulangan terdapat logic untuk variabel x dan y yang menggunakan package math rand.Float64() yang akan membuat value dari x dan y akan selalu acak setiap perulangan. Setelahnya akan

dilakukan cek dengan if, dengan kondisi jika x dan y sekian maka masuk daerah A, B, C atau D dengan value dari daerah dengan kondisi true akan bertambah 1.

Setelah perulangan selesai akan dilakukan perkalian dari variabel curahHujan tiap daerah dengan konstanta ukuranTetesan, tidak lupa juga untuk casting nilai dari variabel daerahA – daerahD yang menggunakan tipe data integer ke dalam tipe data float64. Lalu dilakukan print untuk menunjukkan hasil dari curahHujan.

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>