MODUL VII
FUNCTION MULTIPLE VALUE & VARIADIC FUNCTION

Pertemuan : 7
Waktu : 8 x 60 menit (Online)

1.1 Tujuan Modul III

Setelah mahasiswa mempelajari materi ini, diharapkan dapat :

1. Memahami penggunaan function multiple value di golang.
2. Memahami penggunaan variadic value di golang.

1.2 Landasan Teori

1.2.1 Function Multiple Value

Umumnya fungsi hanya memiliki satu buah nilai balik saja. Jika ada kebutuhan dimana data yang dikembalikan harus banyak, biasanya digunakanlah tipe seperti `map`, slice, atau `struct` sebagai nilai balik.

Go menyediakan kapabilitas bagi programmer untuk membuat fungsi memiliki banyak nilai balik. Di bab ini akan dibahas bagaimana penerapannya.

1.2.1.1 Penerapan Fungsi Multiple Return

Cara membuat fungsi yang memiliki banyak nilai balik tidaklah sulit. Tinggal tulis saja pada saat deklarasi fungsi semua tipe data nilai yang dikembalikan, dan pada keyword `return` tulis semua data yang ingin dikembalikan. Contoh bisa dilihat pada berikut.

```
package main

import "fmt"
import "math"

func calculate(d float64) (float64, float64) {
```

```
    // hitung luas
    var area = math.Pi * math.Pow(d / 2, 2)
    // hitung keliling
    var circumference = math.Pi * d

    // kembalikan 2 nilai
    return area, circumference
}
```

Fungsi `calculate()` di atas menerima satu buah parameter (`diameter`) yang digunakan dalam proses perhitungan. Di dalam fungsi tersebut ada 2 hal yang dihitung, yaitu nilai **keliling** dan **lingkaran**. Kedua nilai tersebut kemudian dijadikan sebagai return value fungsi.

Cara pendefinisian banyak nilai balik bisa dilihat pada kode di atas, langsung tulis tipe data semua nilai balik dipisah tanda koma, lalu ditambahkan kurung diantaranya.

```
func calculate(d float64) (float64, float64)
```

Tak lupa di bagian penulisan keyword `return` harus dituliskan juga semua data yang dijadikan nilai balik (dengan pemisah tanda koma).

```
return area, circumference
```

Implementasi dari fungsi `calculate()` di atas, bisa dilihat pada kode berikut.

```
func main() {
    var diameter float64 = 15
    var area, circumference = calculate(diameter)

    fmt.Printf("luas lingkaran\t\t: %.2f \n", area)
    fmt.Printf("keliling lingkaran\t: %.2f \n",
circumference)
}
```

Output program:

```
[novalagung:belajar-golang $ go run bab18.go                                    ]
 luas lingkaran          : 176.71
 keliling linkgkaran     : 47.12
 novalagung:belajar-golang $
```

Karena fungsi tersebut memiliki banyak nilai balik, maka pada pemanggilannya harus disiapkan juga banyak variabel untuk menampung nilai kembalian yang ada (sesuai jumlah nilai balik fungsi).

```
var area, circumference = calculate(diameter)
```

1.2.1.2 Fungsi Dengan Predefined Return Value

Keunikan lainnya yang jarang ditemui di bahasa lain adalah, di Go variabel yang digunakan sebagai nilai balik bisa didefinisikan di-awal.

```
func calculate(d float64) (area float64, circumference
float64) {
    area = math.Pi * math.Pow(d / 2, 2)
    circumference = math.Pi * d

    return
}
```

Fungsi `calculate` kita modif menjadi lebih sederhana. Bisa dilihat di kode di atas, ada cukup banyak perbedaan dibanding fungsi `calculate` sebelumnya. Perhatikan kode berikut.

```
func calculate(d float64) (area float64, circumference
float64) {
```

Fungsi dideklarasikan memiliki 2 buah tipe data, dan variabel yang nantinya dijadikan nilai balik juga dideklarasikan. Variabel `area` yang bertipe `float64`, dan `circumference` bertipe `float64`.

Karena variabel nilai balik sudah ditentukan di awal, untuk mengembalikan nilai cukup dengan memanggil `return` tanpa perlu diikuti variabel apapun. Nilai terakhir `area` dan `circumference` sebelum pemanggilan keyword `return` adalah hasil dari fungsi di atas.

1.2.2 Variadic Function

Go mengadopsi konsep **variadic function** atau pembuatan fungsi dengan parameter sejenis yang tak terbatas. Maksud **tak terbatas** disini adalah jumlah parameter yang disisipkan ketika pemanggilan fungsi bisa berapa saja.

Parameter variadic memiliki sifat yang mirip dengan slice. Nilai dari parameter-parameter yang disisipkan bertipe data sama, dan ditampung oleh sebuah variabel saja. Cara pengaksesan tiap datanya juga sama, dengan menggunakan index.

1.2.1.1 Penerapan Fungsi Variadic

Deklarasi parameter variadic sama dengan cara deklarasi variabel biasa, pembedanya adalah pada parameter jenis ini ditambahkan tanda titik tiga kali (`...`) tepat setelah penulisan variabel (sebelum tipe data). Nantinya semua nilai yang disisipkan sebagai parameter akan ditampung oleh variabel tersebut.

Berikut merupakan contoh penerapannya

```
package main

import "fmt"

func main() {
    var avg = calculate(2, 4, 3, 5, 4, 3, 3, 5, 5, 3)
    var msg = fmt.Sprintf("Rata-rata : %.2f", avg)
    fmt.Println(msg)
}

func calculate(numbers ...int) float64 {
    var total int = 0
    for _, number := range numbers {
        total += number
    }

    var avg = float64(total) / float64(len(numbers))
    return avg
}
```

Output program:

```
[novalagung:belajar-golang $ go run bab19.go
Rata-rata : 3.70
novalagung:belajar-golang $
```

Bisa dilihat pada fungsi `calculate()`, parameter `numbers` dideklarasikan dengan disisipkan tanda 3 titik (`...`), menandakan bahwa `numbers` adalah sebuah parameter variadic dengan tipe data `int`.

```
func calculate(numbers ...int) float64 {
```

Pemanggilan fungsi dilakukan seperti biasa, hanya saja jumlah parameter yang disisipkan bisa banyak.

```
var avg = calculate(2, 4, 3, 5, 4, 3, 3, 5, 5, 3)
```

Nilai tiap parameter bisa diakses seperti cara pengaksesan tiap elemen slice. Pada contoh di atas metode yang dipilih adalah `for` - `range`.

```
for _, number := range numbers {
```

Berikut merupakan penjelasan tambahan dari kode yang telah kita tulis.

1.2.1.1.1 Penggunaan Fungsi fmt.Sprintf()

Fungsi `fmt.Sprintf()` pada dasarnya sama dengan `fmt.Printf()`, hanya saja fungsi ini tidak menampilkan nilai, melainkan mengembalikan nilainya dalam bentuk string. Pada kasus di atas, nilai kembalian `fmt.Sprintf()` ditampung oleh variabel `msg`.

Selain `fmt.Sprintf()`, ada juga `fmt.Sprint()` dan `fmt.Sprintln()`.

1.2.1.1.2 Penggunaan Fungsi float64()

Sebelumnya sudah dibahas bahwa `float64` merupakan tipe data. Tipe data jika ditulis sebagai fungsi (penandanya ada tanda kurungnya) berguna untuk **casting**. Casting sendiri adalah teknik untuk konversi tipe sebuah data ke tipe lain. Hampir semua jenis tipe data dasar yang telah dipelajari di bab 9 bisa digunakan untuk casting. Dan cara penerapannya juga sama, cukup panggil sebagai fungsi, lalu masukan data yang ingin dikonversi sebagai parameter.

Pada contoh di atas, variabel `total` yang tipenya adalah `int`, dikonversi menjadi `float64`, begitu juga `len(numbers)` yang menghasilkan `int` dikonversi ke `float64`.

Variabel `avg` perlu dijadikan `float64` karena penghitungan rata-rata lebih sering menghasilkan nilai desimal.

Operasi bilangan (perkalian, pembagian, dan lainnya) di Go hanya bisa dilakukan jika tipe datanya sejenis. Maka dari itulah perlu adanya casting ke tipe `float64` pada tiap operand.

1.2.1.2 Pengisian Parameter Fungsi Variadic Menggunakan Data Slice

Slice bisa digunakan sebagai parameter variadic. Caranya dengan menambahkan tanda titik tiga kali, tepat setelah nama variabel yang dijadikan parameter. Contohnya bisa dilihat pada kode berikut.

```
var numbers = []int{2, 4, 3, 5, 4, 3, 3, 5, 5, 3}
var avg = calculate(numbers...)
var msg = fmt.Sprintf("Rata-rata : %.2f", avg)

fmt.Println(msg)
```

Pada kode di atas, variabel `numbers` yang merupakan slice int, disisipkan ke fungsi `calculate()` sebagai parameter variadic (bisa dilihat tanda 3 titik setelah penulisan variabel). Teknik ini sangat berguna ketika sebuah data slice ingin difungsikan sebagai parameter variadic.

Perhatikan juga kode berikut ini. Intinya adalah sama, hanya caranya yang berbeda.

```
var numbers = []int{2, 4, 3, 5, 4, 3, 3, 5, 5, 3}
var avg = calculate(numbers...)

// atau

var avg = calculate(2, 4, 3, 5, 4, 3, 3, 5, 5, 3)
```

Pada deklarasi parameter fungsi variadic, tanda 3 titik (`...`) dituliskan sebelum tipe data parameter. Sedangkan pada pemanggilan fungsi dengan menyisipkan parameter array, tanda tersebut dituliskan di belakang variabelnya.

1.2.1.3 Fungsi Dengan Parameter Biasa & Variadic

Parameter variadic bisa dikombinasikan dengan parameter biasa, dengan syarat parameter variadic-nya harus diposisikan di akhir. Contohnya bisa dilihat pada kode berikut.

```
import "fmt"
import "strings"

func yourHobbies(name string, hobbies ...string) {
    var hobbiesAsString = strings.Join(hobbies, ", ")

    fmt.Printf("Hello, my name is: %s\n", name)
    fmt.Printf("My hobbies are: %s\n", hobbiesAsString)
}
```

Nilai parameter pertama fungsi `yourHobbies()` akan ditampung oleh `name`, sedangkan nilai parameter kedua dan seterusnya akan ditampung oleh `hobbies` sebagai slice.

Cara pemanggilannya masih sama seperti pada fungsi biasa.

```
func main() {
    yourHobbies("wick", "sleeping", "eating")
}
```

Jika parameter kedua dan seterusnya ingin diisi dengan data dari slice, menggunakan tanda titik tiga kali.

```
func main() {
    var hobbies = []string{"sleeping", "eating"}
    yourHobbies("wick", hobbies...)
}
```

1.3 Praktikum

1.3.1 Latihan Praktikum

1. Sebuah acara penghargaan musik bergengsi membutuhkan suatu sistem untuk menyimpan data jumlah vote yang dilakukan oleh audience. Data

tersebut terdiri dari ID, Judul, Penyanyi, dan Jumlah Vote. Buatkan sebuah program untuk mengolah data dengan ketentuan fitur berikut ini:

a. Terdapat tampilan menu untuk melakukan pengolahan data.
b. Memasukan data judul, penyanyi dan jumlah vote kedalam list terurut menurut jumlah vote. Untuk id dibuat langsung secara otomatis oleh sistem dengan syarat id tidak boleh kosong ataupun sama dengan data lain.
c. Menghapus data menurut ID.
d. Menampilkan keseluruhan data ditambah dengan menampilkan jumlah data yang tersimpan.
e. Menjumlahkan keseluruhan jumlah vote dan menampilkannya
f. Menampilkan top 3 musik terfavorit.
g. Menampilkan seluruh data dengan nama penyanyi berawalan dengan huruf ‘A’
h. Sertakan message validasi required dalam bentuk function multiple value

2. Seorang pebisnis ingin membuat suatu bisnis warnet. Beliau menginginkan program untuk menyimpan data dan mengolah data costumernya. Data tersebut terdiri dari ID, Nama, jumlah jam penggunaan, dan biaya yang harus dikeluarkan. Buatkan sebuah program untuk mengolah data dengan ketentuan fitur berikut ini:
    a. Terdapat tampilan menu untuk melakukan pengolahan data.
    b. Memasukan data ID, Nama dan Jam Penggunaan kedalam list terurut menurut ID. Untuk data biaya langsung dibuat secara otomatis oleh sistem dengan syarat biaya yang harus dikeluarkan setiap menitnya yaitu Rp 1.000,-.
    c. Menghapus data menurut ID.
    d. Menampilkan keseluruhan data ditambah dengan menampilkan jumlah data yang tersimpan.
    e. Menampilkan rata – rata jumlah jam penggunaan dari customer
    f. Menampilkan 3 buah data yang memiliki jam penggunaan paling sedikit.
    g. Menampilkan seluruh data customer yang menyewa komputer kurang dari rata – rata.
    h. Sertakan message validasi required dalam bentuk function multiple value