3 Struktur Kontrol di Go

Objektif :

- **Mengenal struktur kontrol pada pemrograman Go**
- **Mampu membuat program menggunakan statement control**

Buat program untuk menghitung sampai 10, mulai dari bilangan 1 dan setiap bilangan ditulis dalam baris berbeda.

```
package main

import "fmt"

func main() {
    fmt.Println(1)
    fmt.Println(2)
    fmt.Println(3)
    fmt.Println(4)
    fmt.Println(5)
    fmt.Println(6)
    fmt.Println(7)
    fmt.Println(8)
    fmt.Println(9)
    fmt.Println(10)
}
```

Atau program :

```
package main
import "fmt"

func main() {
    fmt.Println(`1
2
3
4
5
6
7
8
9
10`)
}
```

Kedua program tersebut dituliskan dengan cara yang membosankan. Yang diperlukan adalah cara membuat program yang melakukan sesuatu beberapa kali.

3.1. FOR

Statement for memungkinkan untuk mengulang sederetan statement (sebuah blok) beberapa kali. Penulisan kembali program sebelumnya menggunakan statement for menjadi :

```
package main

import "fmt"

func main() {
    i := 1
    for i <= 10 {
        fmt.Println(i)
        i = i + 1
    }
}
```

Pertama, buat variabel dalam hal ini i yang digunakan untuk menyimpan bilangan yang akan dicetak. Kemudian buat perulangan for menggunakan keyword for, yang menyediakan ekspresi kondisi yang mempunyai nilai true atau false sampai akhirnya eksekusi sebuah blok. Loop for bekerja seperti :

1. Evaluasi ekspresi apakah nilai i <= 10. Jika true makan jalankan statement dalam blok, selain itu lompat ke baris berikutnya setelah blok (dalam hal ini tidak ada, sehingga langsung keluar dari program).
2. Setelah statement dalam blok dijalankan, program kembali ke awal dari statement for dan diulang dengan menambahkan 1.

Baris i = i + 1 berisi statement penting karena tanpa statement tersebut i <= 10 akan selalu bernilai true dan program tidak akan pernah berhenti. Loop for berkerja seperti :

- Buat variabel yang diberi nama i dengan nilai 1
- Apakah i <= 10 ? ya
- Cetak i
- Set i = i + 1 (i sama dengan 2)
- Apakah i <= 10 ? ya
- Cetak i
- Set i = i + 1 (i sama dengan 3)
- .....
- Set i = i + 1 (i sama dengan 11)
- Apakah i <= 10 ? tidak
- Tidak ada yang dikerjakan, keluar dari program.

Bahasa pemrograman lain mempunyai banyak tipe perulangan yang berbeda (while, do, until, foreach ,...) tetapi Go hanya mempunyai satu yang dapat digunakan dalam program sebelumnya dapat ditulis seperti dibawah :

```
func main() {
    for i := 1; i <= 10; i++ {
        fmt.Println(i)
    }
}
```

3.2. IF

Modifikasi program yang mencetak bilangan 1 – 10 di setiap baris dan menetapkan apakah bilangan-bilangan tersebut adalah genap atau ganjil seperti di bawah ini :

```
1 odd
2 even
3 odd
4 even
5 odd
6 even
7 odd
8 even
9 odd
10 even
```

Pertama, diperlukan suatu cara untuk menentukan apakah sebuah bilangan tersebut adalah genap atau ganjil. Cara mudah untuk itu adalah bagi bilangan dengan 2. Jika dari hasil pembagian tersebut tidak ada sisa makan bilangan adalah bilangan genap, selain itu ganjil. Bagaimana menemukan sisa pembagian di Go ? Gunakan operator %. 1 % 2 sama dengan 1, 2 % 2 sama dengan 0, 3 % 2 sama dengan 1 dan seterusnya. Kemudian diperlukan cara untuk memilih sesuatu yang berbeda berdasarkan sebuah kondisi. Untuk itu gunakan statement if :

```
if i % 2 == 0 {
  // even
} else {
  // odd
}
```

Sebuah statement if mirip dengan statament for, mempunyai kondisi yang diikuti dengan sebuah blok. Statement if juga mempunyai bagian opsional else. Jika kondisi dievaluasi bernilai true makan blok setelah kondisi dijalankan, selain itu blok dilewatkan atau jika blok else ada, blok tersebut dijalankan. Statement if berikut mempunyai bagian else if :

```
if i % 2 == 0 {
    // divisible by 2
} else if i % 3 == 0 {
    // divisible by 3
} else if i % 4 == 0 {
    // divisible by 4
}
```

Kondisi dicek dari atas ke bawah dan baris pertama yang meghasilkan nilai true akan menjadi blok yang dieksekusi. Tidak ada blok lain yang dieksekusi, bahkan jika kondisinyadalah *pass* (Sebagai contoh, bilangan 8 dapat dibagi dengan 4 dan 2, tapi blok //divisible by 4 tidak akan pernah dijalankan karena blok //divisible by 2 dijalankan pertama kali. Program untuk menentukan ganjil atau genap :

```
func main() {
    for i := 1; i <= 10; i++ {
        if i % 2 == 0 {
            fmt.Println(i, "even")
        } else {
            fmt.Println(i, "odd")
        }
    }
}
```

Struktur kendali if berkerja seperti :

- Buat sebuah variabel i bertipe int dan beri nilai 1
- Apakah i kurang dari sama dengan 10 ? Ya, lompat ke blok
- Apakah sisa hasil bagi i ÷ 2 sama dengan 0 ? Tidak, lompat ke blok else
- Cetak i diikuti dengan dengan odd
- Tambahkan i (statement sesudah kondisi)
- Apakah i kurang dari sama dengan 10 ? Ya, lompat ke blok
- Apakah sisa hasil bagi i ÷ 2 sama dengan 0 ? Ya, lompat ke blok if
- Cetak i diikuti dengan dengan even
- .....

5.3. SWITCH

Buat program untuk mencetak bahasa Inggris dari suatu bilangan. Menggunakan apa yang telah dipelajari, maka dapat dibuat program :

```
if i == 0 {
    fmt.Println("Zero")
} else if i == 1 {
    fmt.Println("One")
} else if i == 2 {
    fmt.Println("Two")
} else if i == 3 {
    fmt.Println("Three")
} else if i == 4 {
    fmt.Println("Four")
} else if i == 5 {
    fmt.Println("Five")
}
```

Penulisan program dengan cara seperti itu akan membosanan. Go menyediakan statement lain untuk membuatnya menjadi lebih mudah yaitu statement switch. Program diatas dapat ditulis dengan cara berbeda seperti dibawah ini :

```
switch i {
case 0: fmt.Println("Zero")
case 1: fmt.Println("One")
case 2: fmt.Println("Two")
case 3: fmt.Println("Three")
case 4: fmt.Println("Four")
case 5: fmt.Println("Five")
default: fmt.Println("Unknown Number")
}
```

Statement switch dimulai dengan keyword switch diikuti dengan ekspresi (dalah hal ini i) dan kemudian deretan case. Nilai dari ekspresi dibandingkan dengan ekspresi yang mengikuti setiap case. Jika sama makan statement yang mengikuti **:** dijalankan.

Seperti statement if, setiap case diperiksa dari atas ke bawah dan baris pertama yang ditemui sesuai dipilih. Switch juga mendukung default case yang akan dijalankan jika tidak ada case yang cocok (seperti else pada statement if).