



Scalable Web Service With Golang

sesi 2

The background is a solid dark blue. It features several decorative geometric elements: a light blue triangle in the upper left, a dark blue circle in the upper right, a light blue rectangle in the top right corner, a white circle in the lower left, and a white arc in the bottom right corner.

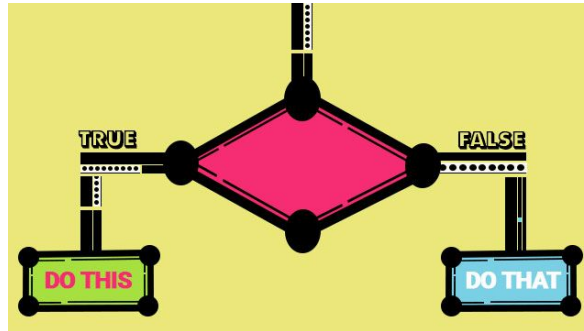
Conditions⁺

Conditions - Sesi 2

Conditions

Kondisional pada pemrograman dapat digunakan untuk mengatur alur dari suatu program. Analogi yang bisa kita pakai pada kehidupan sehari-hari adalah analogi lampu merah, yang dimana lampu merah digunakan untuk mengatur lalu lintas. Misalkan kita mendapat lampu merah maka kendaraan kita harus berhenti, misalkan kita mendapat lampu kuning maka kendaraan kita harus bersiap-siap dan jika kita mendapatkan lampu hijau maka kita harus menjalankan kendaraan kita. Sama halnya di dalam pemrograman, yang dimana dengan menggunakan conditions maka kita dapat mengatur program kita untuk mengetahui kapan harus mengeksekusi suatu kode.

Pada bahasa Go, terdapat 2 macam kondisional yang dapat kita pakai yaitu if else dan switch



<https://www.flocabulary.com/unit/coding-conditionals/>

Conditions - Sesi 2

Conditions(Temporary variable)

Pada bahasa Go, kita dapat membuat suatu variable yang dimana variable tersebut hanya bisa diakses dan digunakan pada scope block dari suatu kondisional.

Contohnya seperti pada gambar dibawah.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var currentYear = 2021
7
8     if age := currentYear - 1998; age < 17 {
9         fmt.Println("Kamu belum boleh membuat kartu sim")
10    } else {
11        fmt.Println("Kamu sudah boleh membuat kartu sim")
12    }
13 }
14
```

Conditions - Sesi 2

Conditions(Temporary variable)

Jika kita perhatikan pada baris 8, *age* merupakan sebuah variable yang terbuat pada scope block kondisional *if else* nya.

Kita membuat variable *age* yang dimana kita membuat perkurangan antara variable *currentYear* dengan angka 1998. Lalu setelah tanda *semicolon ;* kita langsung membuat kondisinya yang mengatakan jika variable *age* lebih kecil daripada angka 17. Kondisi dibawah akan menghasilkan kalimat yang ada pada blok *else* karena variable *age* menghasilkan nilai dengan angka 23.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var currentYear = 2021
7
8     if age := currentYear - 1998; age < 17 {
9         fmt.Println("Kamu belum boleh membuat kartu sim")
10    } else {
11        fmt.Println("Kamu sudah boleh membuat kartu sim")
12    }
13 }
14
```



Conditions - Sesi 2

Switch

Selain menggunakan keyword *if*, *else if* dan *else*, kita juga bisa menggunakan keyword *switch* untuk membuat suatu kondisional, seperti contohnya pada gambar disebelah kanan.

Jika kita perhatikan, kondisional pada gambar disebelah kanan akan menghasilkan kalimat "perfect" karena *case* pertama sudah terpenuhi. Jika tidak ada *case* yang terpenuhi maka kondisional tersebut akan menghasilkan kalimat "not bad" karena keyword *default* sama dengan keyword *else* yang dimana akan dieksekusi blocknya jika tidak ada kondisi yang terpenuhi.

Switch pada bahasa Go tidak memerlukan keyword *break*, jadi jika suatu *case* telah terpenuhi maka dia tidak akan berlanjut kepada *case* berikutnya.

```
package main

import "fmt"

func main() {

    var score = 8

    switch score {
    case 8:
        fmt.Println("perfect")
    case 7:
        fmt.Println("awesome")
    default:
        fmt.Println("not bad")
    }
}
```



Conditions - Sesi 2

Switch with relational operators

Kita juga dapat menggunakan switch dengan operator perbandingan seperti layaknya pada sebuah kondisional dengan keyword *if*, *else if* dan *else*.

Contohnya seperti pada gambar disebelah kanan. Kondisional pada gambar disebelah kanan akan menghasilkan kalimat "not bad" karena variable score memiliki angka 6 dan angka 6 memang lebih kecil dari angka 8 dan lebih besar daripada angka 3.

Lalu kita juga bisa melihat bahwa scope block *default* pada gambar disebelah kanan menggunakan kurung kurawal.

Hal ini sangat bagus diterapkan jika kita ingin membuat lebih dari satu *statement* pada scope block *default* agar syntax kita lebih rapi dan mudah di maintain.

```
package main

import "fmt"

func main() {
    var score = 6

    switch {
    case score == 8:
        fmt.Println("perfect")
    case (score < 8) && (score > 3):
        fmt.Println("not bad")
    default:
        {
            fmt.Println("study harder")
            fmt.Println("you need to learn more")
        }
    }
}
```



Switch (fallthrough keyword)

Ketika kita ingin switch pada bahasa Go melanjutkan pengecekan kepada case selanjutnya walaupun suatu case telah terpenuhi kondisinya, maka kita bisa menggunakan keyword *fallthrough*.

Jika kita perhatikan pada gambar disebelah kanan, kondisional *switch* tersebut akan menghasilkan 2 kalimat yaitu "not bad" dan "It is ok, but you need to study harder" karena ketika kondisi pada case kedua terpenuhi, maka dia pengecekan akan berlanjut kepada case ketiga karena case kedua menggunakan keyword *fallthrough*.

Ketika berlanjut ke case ketiga, kondisinya terpenuhi karena variable *score* memiliki nilai berupa angka 6 dan angka 6 memang lebih besar dari angka 5.

```
package main

import "fmt"

func main() {
    var score = 6

    switch {
    case score == 8:
        fmt.Println("perfect")
    case (score < 8) && (score > 3):
        fmt.Println("not bad")
        fallthrough
    case score < 5:
        fmt.Println("It is ok, but please study harder")
    default:
        {
            fmt.Println("study harder")
            fmt.Println("You don't have a good score yet")
        }
    }
}
```



Conditions - Sesi 2

Nested Conditions

Kita juga dapat melakukan kondisional bersarang seperti pada gambar disebelah kanan.

Kondisional bersarang merupakan sebuah proses kondisional yang didalamnya terdapat proses kondisional kembali. Kita dapat menggunakan *if*, *else if*, *else*, *switch* ataupun menggabungkan seluruhnya.

```
package main

import "fmt"

func main() {
    var score = 10

    if score > 7 {
        switch score {
            case 10:
                fmt.Println("perfect!")
            default:
                fmt.Println("nice!")
        }
    } else {
        if score == 5 {
            fmt.Println("not bad")
        } else if score == 3 {
            fmt.Println("keep trying")
        } else {
            fmt.Println("you can do it")
            if score == 0 {
                fmt.Println("try harder!")
            }
        }
    }
}
```

