

Hari 3 - Operator dan Conditional

Operator dalam Go

Operator adalah karakter khusus yang merepresentasikan sebuah tindakan. Operator terbagi ke dalam beberapa jenis:

1. Operator Aritmatika

Operator aritmatika adalah operator yang digunakan untuk operasi yang sifatnya perhitungan. Go mendukung beberapa operator aritmatika standar, list-nya bisa dilihat di tabel berikut.

Operator	Keterangan
+	penjumlahan
-	pengurangan
*	perkalian
/	pembagian
%	modulus / sisa hasil pembagian

contoh penggunaan:

```
// operator penjumlahan
jumlah := 8 + 3
fmt.Println(jumlah) // hasilnya 13

// operator pengurangan
kurang := 8 - 3
fmt.Println(kurang) // hasilnya 5

// operator perkalian
kali := 8 * 3
fmt.Println(kali) // hasilnya 24

// operator pembagian
bagi := 8 / 4
fmt.Println(bagi) // hasilnya 2

// operator modulus
modulus := 8 % 3
fmt.Println(modulus) // hasilnya 2
```

selain itu terdapat juga augmented assignment berikut listnya:

Operasi Matematika	Augmented Assignments
a = a + 10	a += 10
a = a - 10	a -= 10
a = a * 10	a *= 10
a = a / 10	a /= 10
a = a % 10	a %= 10

contoh penggunaan:

```
var angka = 8
fmt.Println(angka) // 8
angka+=10
fmt.Println(angka) // 18

var angka2 = 5
fmt.Println(angka2) // 5
angka2+=5
fmt.Println(angka2) // 10
```

selain itu terdapat juga unary operator berikut listnya:

Operator	Keterangan
++	a = a + 1
--	a = a - 1

+	positive
!	negasi/ kebalikan dari tipe data boolean

contoh penggunaan:

```
angka := 8
fmt.Println(angka) // 8
angka++
fmt.Println(angka) // 11

angka2 := 5
fmt.Println(angka2) // 5
angka2--
fmt.Println(angka2) // 4
```

2. Operator Perbandingan

Operator perbandingan digunakan untuk menentukan kebenaran suatu kondisi. Hasilnya berupa nilai boolean, **true** atau **false**.

Tabel di bawah ini berisikan operator perbandingan yang bisa digunakan di Go.

Operator	Keterangan
>	lebih dari
<	kurang dari
>=	lebih dari sama dengan
<=	kurang dari sama dengan
==	sama dengan
!=	tidak sama dengan

contoh penggunaan:

```
var angka = 8

fmt.Println(angka > 5) // true

fmt.Println(angka < 5) // false

fmt.Println(angka >= 5) // true

fmt.Println(angka <= 5) // false

fmt.Println(angka == 5) // false

fmt.Println(angka != 5) // true
```

3. Operator Logika

Operator ini digunakan untuk mencari benar tidaknya kombinasi data bertipe **bool** (bisa berupa variabel bertipe **bool**, atau hasil dari operator perbandingan).

Beberapa operator logika standar yang bisa digunakan:

Operator	Keterangan
&&	dan
	atau
!	negasi / nilai kebalikan

contoh penggunaan:

```
var d = false

fmt.Println(a && c) // true

fmt.Println(a && b) // false

fmt.Println(a || b) // true

fmt.Println(b || d) // false

fmt.Println(!b && !d) // true

fmt.Println(!a || b) // false
```

Conditional

Conditional merupakan cara yang digunakan dalam program komputer untuk mengambil keputusan dari kemungkinan benar(true) atau salah(false) terhadap beberapa kondisi.

1. Conditional Menggunakan Keyword `if`, `else if`, & `else`

Cara penerapan if-else di Go sama seperti pada bahasa pemrograman lain. Yang membedakan hanya tanda kurungnya (*parentheses*), di Go tidak perlu ditulis

- Contoh menjalankan kode jika premis bernilai `true`

```
if true {
    fmt.Println("jalankan code")
}
```

- Contoh kode tidak dijalankan jika premis bernilai `false`

```
if false {
    fmt.Println("code tidak dijalankan")
}
```

- Contoh Premis dengan perbandingan suatu nilai

```
var mood = "happy"
if mood == "happy"{
    fmt.Println("hari ini aku bahagia!")
}
```

Di dalam conditional dikenal juga dengan istilah branching atau percabangan. Dengan percabangan kita bisa mengecek nilai kebenaran dari berbagai premis yang kita sediakan. cara menambahkan kondisi atau premis lain dari premis pertamanya adalah dengan sintaks `else` . Contohnya ketika kita diminta untuk membeli telur dan buah ke minimarket oleh ibu kita maka akan banyak sekali kondisi atau kemungkinan yang terjadi dalam perjalanan kita membeli barang-barang tersebut di supermarket.

- Contoh Branching sederhana

```
var minimarketStatus = "open"
if minimarketStatus == "open" {
    fmt.Println("saya akan membeli telur dan buah")
} else {
    fmt.Println("minimarketnya tutup")
}
```

Kondisi else di atas adalah kondisi selain `minimarketStatus == "open"` .

- Contoh Branching dengan kondisi

```
var minimarketStatus = "close"
var minuteRemainingToOpen = 5
if minimarketStatus == "open"{
    fmt.Println("saya akan membeli telur dan buah")
} else if minuteRemainingToOpen <= 5 {
    fmt.Println("minimarket buka sebentar lagi, saya tungguin")
} else {
    fmt.Println("minimarket tutup, saya pulang lagi")
}
```

contoh di atas kita memberikan kondisi tambahan yaitu jika minimarket akan buka kurang atau sama dengan 5 menit lagi maka saya akan menunggu.

Selain Branching atau percabangan, conditional juga dapat memiliki sarang atau tumpukan yaitu terdapat conditional di dalam sebuah kondisional.

- Contoh conditional bersarang

```
if minimarketStatus == "open" {
    fmt.Println("saya akan membeli telur dan buah")
    if telur == "soldout" || buah == "soldout" {
        fmt.Println("belanjaan saya tidak lengkap")
    } else if telur == "soldout" {
        fmt.Println("telur habis")
    } else if buah == "soldout" {
        fmt.Println("buah habis")
    }
} else {
    fmt.Println("minimarket tutup, saya pulang lagi")
}
```

pada kondisi di atas minimarketnya sudah buka lalu cek kondisi selanjutnya cek apakah telur atau buah habis. Karena kondisinya adalah jika telur ATAU buah salah satunya ada yang soldout maka belanjaan saya lengkap. Demikian seterusnya akan mengecek kondisi-kondisi yang disediakan dalam logika program yang dibuat.

2. Variabel Temporary Pada **if - else**

Variabel temporary adalah variabel yang hanya bisa digunakan pada blok conditional dimana ia ditempatkan saja. Penggunaan variabel ini membawa beberapa manfaat, antara lain:

- Scope atau cakupan variabel jelas, hanya bisa digunakan pada blok conditional itu saja
- Kode menjadi lebih rapi
- Ketika nilai variabel tersebut didapat dari sebuah komputasi, perhitungan tidak perlu dilakukan di dalam blok masing-masing kondisi.

Berikut contoh penggunaannya:

```
if minimarketStatus, minuteRemainingToOpen := "close", 5; minimarketStatus == "open"{
    fmt.Println("saya akan membeli telur dan buah")
} else if minuteRemainingToOpen <= 5 {
    fmt.Println("minimarket buka sebentar lagi, saya tungguin")
} else {
    fmt.Println("minimarket tutup, saya pulang lagi")
}
```

3. Conditional Menggunakan Keyword **switch - case**

Cara lain untuk melakukan pengondisian atau conditional adalah dengan **switch case**. Penggunaan switch case mirip seperti kita menyalakan tombol switch pada remote. Jika kondisi tombol yang dipijit adalah tombol dengan nomer tertentu maka akan menjalankan program tertentu. **switch case** pada Go agak unik dan berbeda selain terdapat switch case seperti pada umumnya di bahasa pemrograman lain, terdapat juga case dengan banyak kondisi maupun bisa diisi dengan logika seperti if else. berikut ini contoh penggunaanya:

- Contoh Conditional dengan **switch case**

```
var buttonPushed = 1
switch buttonPushed {
case 1:
    fmt.Println("matikan TV!")
case 2:
    fmt.Println("turunkan volume TV!")
case 3:
    fmt.Println("tingkatkan volume TV!")
case 4:
    fmt.Println("matikan suara TV!")
default:
    fmt.Println("Tidak terjadi apa-apa")
}
```

Pada kode di atas, switch akan mengevaluasi nilai **buttonPushed** jika **case** yang ditemui cocok dengan nilai yang diberikan maka kode pada **case** tersebut dijalankan sehingga akan dimunculkan “**matikan TV!**”. Terdapat **default** yaitu kondisi dimana tidak ditemukan **case** yang sesuai

- Contoh Conditional dengan **switch case** dengan banyak kondisi

```
var buttonPushed = 1
switch buttonPushed {
case 1:
    fmt.Println("matikan TV!")
case 2, 3, 4:
    fmt.Println("turunkan volume TV!")
default:
    fmt.Println("Tidak terjadi apa-apa")
}
```

```
var buttonPushed = 1
switch{
case buttonPushed == 1:
    fmt.Println("matikan TV!")
case buttonPushed == 2 && buttonPushed == 3:
    fmt.Println("turunkan volume TV!")
case buttonPushed == 4:
    fmt.Println("matikan suara TV!")
default:
    fmt.Println("Tidak terjadi apa-apa")
}
```

Penggunaan Keyword **fallthrough**Dalam switch

Seperti yang sudah dijelaskan sebelumnya, bahwa switch pada Go memiliki perbedaan dengan bahasa lain. Ketika sebuah **case** terpenuhi, pengecekan kondisi tidak akan diteruskan ke case-case setelahnya.

Keyword **fallthrough** digunakan untuk memaksa proses pengecekan diteruskan ke **case** selanjutnya dengan **tanpa menghiraukan nilai kondisinya**, jadi case di pengecekan selanjutnya tersebut selalu dianggap benar (meskipun aslinya adalah salah).

```
var point = 6

switch {
case point == 8:
    fmt.Println("perfect")
case (point < 8) && (point > 3):
    fmt.Println("awesome")
    fallthrough
case point < 5:
    fmt.Println("you need to learn more")
default:
    {
        fmt.Println("not bad")
        fmt.Println("you need to learn more")
    }
}
```

Setelah pengecekan **case (point < 8) && (point > 3)** selesai, akan dilanjut ke pengecekan **case point < 5**, karena ada **fallthrough** di situ.

Referensi Video:

- [Operasi Matematika](#) (Programmer Zaman Now)
- [Operasi Perbandingan](#) (Programmer Zaman Now)
- [Operasi Boolean](#) (Programmer Zaman Now)
- [IF Expression](#) (Programmer Zaman Now)
- [Switch Expression](#) (Programmer Zaman Now)

Referensi Tulisan:

- <https://dasarpemrogramangolang.novalagung.com/A-operator.html>
- <https://dasarpemrogramangolang.novalagung.com/A-seleksi-kondisi.html>

Rating - Feedback

Berikan Rating pada posting ini:



Berikan kritik dan saran..

