



Array

+

Array - Sesi 2

Array

Array pada bahasa Go merupakan sebuah tipe data untuk menyimpan sebuah kumpulan dari data-data. Data-data yang kita simpan pada sebuah array dalam bahasa Go harus memiliki tipe data yang sama, kecuali kita menyimpannya sebagai suatu interface kosong yang akan kita bahas nanti.

Array pada bahasa Go memiliki sifat *fixed-length* atau memiliki panjang yang tetap yang harus kita tentukan dari awal kita membuat arraynya.

```
package main

import "fmt"

func main() {
    var numbers [4]int
    numbers = [4]int{1, 2, 3, 4}

    var strings = [3]string{"Airell", "Nanda", "Mailo"}

    fmt.Printf("%#v\n", numbers)
    fmt.Printf("%#v\n", strings)
}
```

```
[4]int{1, 2, 3, 4}
[3]string{"Airell", "Nanda", "Mailo"}
```



Array - Sesi 2

Array

Kita dapat membuat array dengan 2 macam cara yaitu dengan mendeklarasikannya terlebih dahulu tanpa memberi nilai apapun, dan kita juga bisa mendeklarasikan dan langsung menginisialisasikannya dengan memberikannya sebuah nilai.

Contohnya seperti pada gambar pertama di sebelah kanan. Kita telah membuat 2 variable bernama *numbers* dan *strings*.

Variable *numbers* memiliki tipe data array of int atau array dengan tipe data int dan memiliki panjang 4. Lalu variable *strings* memiliki tipe data array of strings atau array dengan tipe data string.

Lalu kita memakai verb `%#v` untuk memformat tipe data array agar kita juga dapat melihat panjang dari arraynya. Jika kita jalankan maka hasilnya akan seperti pada gambar kedua.

```
package main

import "fmt"

func main() {
    var numbers [4]int
    numbers = [4]int{1, 2, 3, 4}

    var strings = [3]string{"Airell", "Nanda", "Mailo"}

    fmt.Printf("%#v\n", numbers)
    fmt.Printf("%#v\n", strings)
}
```

```
[4]int{1, 2, 3, 4}
[3]string{"Airell", "Nanda", "Mailo"}
```



Array - Sesi 2

Array(Modify Element Through Index)

Kita juga dapat memodifikasi element-element yang terdapat dalam sebuah array dengan cara mengakses indexnya. Contohnya seperti pada gambar pertama disebelah kanan kita.

Terdapat sebuah variable dengan nama *fruits* yang memiliki array yang isinya adalah nama buah-buahan menggunakan bahasa Indonesia.

Lalu buah-buahan yang terdapat pada array dalam variable *fruits* tersebut di modifikasi dengan cara mengakses indexnya dan mengganti buah-buahannya ke dalam bahasa Inggris. Dan jika dijalankan hasilnya seperti pada gambar kedua.

```
package main

import "fmt"

func main() {
    var fruits = [3]string{"apel", "pisang", "mangga"}
    fruits[0] = "apple"
    fruits[1] = "banana"
    fruits[2] = "mango"

    fmt.Printf("%#v\n", fruits)
}
```

```
go run array.go
[3]string{"apple", "banana", "mango"}
```



Array - Sesi 2

Array(Loop through elements)

Ada 2 cara penulisan agar kita dapat melakukan looping untuk mengakses element-element yang terdapat pada sebuah array. Perhatikan pada gambar pertama disebelah kanan. Cara pertamanya adalah dengan menggunakan *range loop*.

Dengan menggunakan *range loop* kita bisa dengan mudah mengakses seluruh element pada sebuah array dengan bisa juga sekaligus mendapatkan indexnya.

Lalu cara keduanya adalah kita bisa menggunakan looping biasa dan kita perlu menggunakan bantuan fungsi *len* untuk mendapatkan panjang dari array nya dengan cara penulisan `len(<nama array>)`. Jika kita jalankan syntax pada gambar pertama, maka hasilnya akan seperti pada gambar kedua di sebelah kanan.

```
package main

import (
    "fmt"
    "strings"
)

func main() {

    var fruits = [3]string{"apple", "banana", "mango"}
    //Cara Pertama
    for i, v := range fruits {
        fmt.Printf("Index: %d, Value: %s\n", i, v)
    }

    fmt.Println(strings.Repeat("#", 25))

    //Cara Kedua
    for i := 0; i < len(fruits); i++ {
        fmt.Printf("Index: %d, Value: %s\n", i, fruits[i])
    }
}
```

```
Index: 0, Value: apple
Index: 1, Value: banana
Index: 2, Value: mango
#####
Index: 0, Value: apple
Index: 1, Value: banana
Index: 2, Value: mango
```



Array - Sesi 2

Array(Multidimensional Array)

Kita juga dapat membuat suatu array multidimensional yang berarti terdapat sebuah array di dalam sebuah array. Contohnya seperti pada gambar pertama di sebelah kanan.

Array yang dikandung oleh variable *balances* merupakan sebuah multidimensional array.

Tanda `[2][3]int` memiliki arti bahwa array terluar nya memiliki panjang sama dengan 2, dan array yang berada di dalamnya memiliki panjang sama dengan 3 dengan tipe data *int*. `{5,6,7}` dan `{8,9,10}` merupakan array yang berada di dalam dan masing-masing memiliki panjang sama dengan 3 dengan tipe data *int*. Jika kita jalankan pada terminal kita, maka hasilnya akan seperti pada gambar kedua di sebelah kanan.

```
package main

import (
    "fmt"
)

func main() {

    balances := [2][3]int{{5, 6, 7}, {8, 9, 10}}

    for _, arr := range balances {
        for _, value := range arr {
            fmt.Printf("%d ", value)
        }
        fmt.Println()
    }
}
```

```
5 6 7
8 9 10
```

