



Empty Interface⁺

Empty interface

Empty interface merupakan suatu tipe data yang dapat menerima segala tipe data pada bahasa Go. Maka dari itu, sebuah variable dengan tipe data *empty interface* dapat diberikan nilai dengan tipe data yang berbeda-beda. Contohnya seperti pada gambar di sebelah kanan.

Pada line 4 terdapat sebuah variable bernama *randomValues* yang memiliki tipe data *empty interface*. Kemudian pada line 8, variable *randomValues* diberikan nilai dengan tipe data *string*. Pada line 10, variable *randomValues* diberikan nilai dengan tipe data *int*. Pada line 12, variable *randomValues* diberikan nilai dengan tipe data *bool*. Dan yang terakhir pada line 14, variable *randomValues* diberikan nilai dengan tipe data *slice of int*. Ini menunjukkan bahwa ketika suatu variable mempunyai tipe data *empty interface*, maka variable tersebut dapat diberikan nilai dengan tipe data apapun.

```
1 package main
2
3 func main() {
4     var randomValues interface{}
5
6     _ = randomValues
7
8     randomValues = "Jalan Sudirman"
9
10    randomValues = 20
11
12    randomValues = true
13
14    randomValues = []string{"Airell", "Nanda"}
15
16 }
```



Empty interface (Type assertion)

Perhatikan pada gambar pertama disebelah kanan. Terdapat sebuah variable bernama *v* yang memiliki tipe data *empty interface*. Kemudian variable *v* diberikan nilai dengan tipe data *int*.

Lalu terjadi error pada saat compile time pada saat variable *v*, di reassign dengan hasil perkalian 20 dengan 9. Error tersebut disebabkan karena kita hanya bisa melakukan perkalian terhadap tipe data *int* yang konkrit atau asli, sedangkan variable *v* memiliki tipe data *empty interface* yang diberikan nilai dengan tipe data *int*.

Untuk menanggulangi errornya, maka kita bisa melakukan *type assertion* dengan cara yang sama dengan *type assertion* pada tipe data *interface* yang telah kita bahas pada materi sebelumnya. Caranya seperti pada gambar kedua.

```
package main

func main() {
    var v interface{}

    v = 20

    v = v * 9
}
```

```
package main

func main() {
    var v interface{}

    v = 20

    if value, ok := v.(int); ok == true {
        v = value * 9
    }
}
```



Empty interface (Map & Slice with Empty Interface)

Ketika sebuah *map* kita deklarasikan dan *value-value* nya diberikan tipe data *empty interface*, maka *map* tersebut dapat memiliki *value* dengan tipe data yang berbeda-beda.

Sama halnya dengan *slice* ataupun *array*, ketika dideklarasikan dan isinya diberi tipe data *empty interface* maka *slice* atau *array* tersebut dapat memiliki nilai dengan tipe data yang berbeda-beda. Contohnya seperti pada gambar di sebelah kanan.

Terdapat 2 buah variable bernama *rs* dan *rm*. Variable *rs* memiliki tipe data *slice of empty interface* dan memiliki nilai dengan tipe data yang berbeda-beda. Kemudian variable *rm* memiliki tipe data *map* yang memiliki *key* dengan tipe data *string* dan *value* dengan tipe data *empty interface*. *Value* yang dimiliki oleh *rm* memiliki tipe data yang berbeda-beda.

```
package main

func main() {
    rs := []interface{}{1, "Airell", true, 2, "Ananda", true}

    rm := map[string]interface{}{
        "Name": "Airell",
        "Status": true,
        "Age": 23,
    }

    _, _ = rs, rm
}
```

