



Aliase

+

# Aliase

Aliase merupakan sebuah fitur pada bahasa Go yang digunakan sebagai nama alternative dari tipe data yang sudah ada.

Tipe data dengan nama yang berbeda memiliki arti bahwa tipe data nya juga berbeda, tetapi terdapat pengecualian terhadap *aliase*.

Tipe data *byte* merupakan tipe data *aliase* dari tipe data *uint8*, yang berarti mereka merupakan tipe data yang sama dengan nama yang berbeda.

Tipe data *rune* merupakan tipe data *aliase* dari tipe data *uint32*, yang berarti mereka merupakan tipe data yang sama dengan nama yang berbeda.

# Aliase

```
package main

func main() {
    // deklarasi variable dengan tipe data uint8
    var a uint8 = 10
    var b byte // byte adalah alias dari tipe data uint8

    b = a // no error, karena byte memiliki tipe data yang sama dengan uint8
    _ = b
}
```

Perhatikan pada contoh diatas mengenai penggunaan *aliase*:

- Variable *a* memiliki tipe data *uint8*, sedangkan variable *b* memiliki tipe data *byte*.
- Ketika *b* di-reassign dengan *a*, maka tidak akan terjadi error karena *byte* merupakan tipe data alias dari tipe data *uint8*.

# Aliase

```
package main

import "fmt"

func main() {
    // Mendeklarasi tipe data alias bernama second
    // type nama_alias = nama_tipe_data
    type second = uint

    var hour second = 3600
    fmt.Printf("hour type: %T\n", hour) // => hour type: uint
}
```

Kita juga dapat mendeklarasikan tipe data alias dengan nama yang kita buat sendiri, contohnya seperti pada gambar diatas:

- Terdapat sebuah tipe data bernama *second* yang merupakan tipe data alias dari *uint*.
- Variable *hour* memiliki tipe data *second* yang merupakan sebuah tipe data alias.
- Ketika variable *hour* ingin di ketahui tipe data nya dengan menggunakan flag %T, maka hasil pada terminal akan tetapu menunjukan bahwa tipe data dari variable *hour* ada *uint*. Hal ini terjadi karena tipe data asli atau *underlying type* dari tipe data *second* adalah tipe data *uint*.