



Pointer

+

Pointer

Pointer merupakan sebuah variable spesial yang digunakan untuk menyimpan alamat memori variable lainnya. Sebagai contoh sebuah variabel bertipe integer memiliki nilai 4, maka yang dimaksud pointer adalah alamat memori dimana nilai 4 disimpan, bukan nilai 4 itu sendiri.

Variabel-variabel yang memiliki *reference* atau alamat memori yang sama, saling berhubungan satu sama lain dan nilainya pasti sama. Ketika ada perubahan nilai, maka akan memberikan efek kepada variabel lain (yang alamat memorinya sama) yaitu nilainya juga ikut berubah.

Cara membuat suatu variable menjadi sebuah pointer cukup mudah. Caranya adalah dengan menggunakan tanda asterisk * sebelum penulisan tipe datanya. Contohnya seperti 2 variable pada gambar dibawah ini.

```
package main

func main() {
    var firstNumber *int
    var secondNumber *int
    _, _ = firstNumber, secondNumber
}
```



Pointer (Memory address)

Perhatikan pada gambar pertama disebelah kanan. Terdapat 2 variable bernama *firstNumber* dan *secondNumber*.

Variable *firstNumber* merupakan sebuah variable dengan tipe data *int*.

Kemudian *secondNumber* merupakan sebuah variable *pointer int* yang mengandung alamat memori dari *firstNumber*.

Perlu diperhatikan disini bahwa ketika kita ingin memberikan sebuah nilai kepada variable *pointer*, maka kita perlu menggunakan tanda *ampersand* & seperti yang kita lihat pada line 7.

Kemudian pada line 10, kita mencoba untuk menampilkan alamat memori dari variable *firstNumber* dengan cara menggunakan tanda *ampersand* &.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var firstNumber int = 4
7     var secondNumber *int = &firstNumber
8
9     fmt.Println("firstNumber (value) :", firstNumber)
10    fmt.Println("firstNumber (memori address) :", &firstNumber)
11
12    fmt.Println("secondNumber (value) :", *secondNumber)
13    fmt.Println("secondNumber (memori address) :", secondNumber)
14
15 }
```

```
go run pointer.go
firstNumber (value) : 4
firstNumber (memori address) : 0xc0000ae008
secondNumber (value) : 4
secondNumber (memori address) : 0xc0000ae008
```



Pointer (Memory address)

Kemudian pada line 12, kita mencoba untuk menampilkan nilai asli (bukan alamat memori) yang dikandung oleh *secondNumber* dengan cara memberikan tanda *asterisk* *.

Perlu diingat disini bahwa karena variable *secondNumber* merupakan sebuah variable *pointer* yang mengandung alamat memori dari *firstNumber*, maka ketika kita ingin mendapatkan nilai asli yang dikandung oleh sebuah variable *pointer*, kita perlu menggunakan tanda asterisk *.

Jika kita jalankan pada terminal kita, maka hasilnya akan seperti pada gambar kedua.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var firstNumber int = 4
7     var secondNumber *int = &firstNumber
8
9     fmt.Println("firstNumber (value) :", firstNumber)
10    fmt.Println("firstNumber (memori address) :", &firstNumber)
11
12    fmt.Println("secondNumber (value) :", *secondNumber)
13    fmt.Println("secondNumber (memori address) :", secondNumber)
14
15 }
```

```
go run pointer.go
firstNumber (value) : 4
firstNumber (memori address) : 0xc0000ae008
secondNumber (value) : 4
secondNumber (memori address) : 0xc0000ae008
```



Pointer (Memory address)

Jika kita melihat hasilnya pada gambar kedua di sebelah kanan.

Kita dapat menarik kesimpulan bahwa:

- Kita dapat mendapatkan alamat memori dari variable biasa dengan menggunakan tanda *ampersand* &.
- Kita juga dapat mendapatkan nilai asli dari variable pointer dengan cara menggunakan tanda *asterisk* *.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var firstNumber int = 4
7     var secondNumber *int = &firstNumber
8
9     fmt.Println("firstNumber (value) :", firstNumber)
10    fmt.Println("firstNumber (memori address) :", &firstNumber)
11
12    fmt.Println("secondNumber (value) :", *secondNumber)
13    fmt.Println("secondNumber (memori address) :", secondNumber)
14
15 }
```

```
go run pointer.go
firstNumber (value) : 4
firstNumber (memori address) : 0xc0000ae008
secondNumber (value) : 4
secondNumber (memori address) : 0xc0000ae008
```



Pointer (Changing value through pointer)

Pointer digunakan untuk menyimpan alamat memori, maka ketika kita mengganti nilai dari sebuah pointer, maka variable lainnya yang mempunyai alamat memori yang sama, akan ikut terganti nilainya. Contohnya seperti pada gambar pertama di sebelah kanan.

Variable *firstPerson* awalnya memiliki nilai "John", lalu terdapat variable bernama *secondPerson* yang mengandung alamat memori dari variable *firstPerson*. Lalu ketika variable *secondPerson* merubah nilai aslinya, maka variable *firstPerson* akan ikut terganti juga nilainya. Hal tersebut dapat dilihat dari hasil pada terminal ketika dijalankan pada gambar kedua.

Ketika nilainya berubah, tetapi perlu diingat disini bahwa alamat memorinya tidak berubah, hanya nilai asli yang dikandungnya saja. Dan juga ketika kita sebuah variable *pointer* ingin merubah nilai aslinya, maka kita juga perlu menggunakan tanda *asterisk* * seperti pada line 37 pada gambar pertama.

```
19 package main
20
21 import (
22     "fmt"
23     "strings"
24 )
25
26 func main() {
27     var firstPerson string = "John"
28     var secondPerson *string = &firstPerson
29
30     fmt.Println("firstPerson (value)   :", firstPerson)
31     fmt.Println("firstPerson (memori address) :", &firstPerson)
32     fmt.Println("secondNumber (value)   :", *secondPerson)
33     fmt.Println("secondNumber (memori address) :", secondPerson)
34
35     fmt.Println("\n", strings.Repeat("#", 30), "\n")
36
37     *secondPerson = "Doe"
38
39     fmt.Println("firstPerson (value)   :", firstPerson)
40     fmt.Println("firstPerson (memori address) :", &firstPerson)
41     fmt.Println("secondNumber (value)   :", *secondPerson)
42     fmt.Println("secondNumber (memori address) :", secondPerson)
43
44 }
```

```
firstPerson (value)   : John
firstPerson (memori address) : 0xc0000a0050
secondNumber (value)   : John
secondNumber (memori address) : 0xc0000a0050

#####

firstPerson (value)   : Doe
firstPerson (memori address) : 0xc0000a0050
secondNumber (value)   : Doe
secondNumber (memori address) : 0xc0000a0050
```



Pointer (Pointer as a parameter)

Pointer juga bisa dijadikan sebagai sebuah parameter pada sebuah *function*. Contohnya seperti pada gambar pertama di sebelah kanan.

Awalnya variable *a* memiliki nilai 10. Setelah itu kita memberikan variable *a* kepada function *changeValue* yang menerima satu parameter dengan tipe data *pointer int*. Kemudian nilai dari variable *a* diganti di dalam function *changeValue* melalui parameter yang diterimanya.

Ketika dijalankan maka hasilnya akan seperti pada gambar kedua.

```
package main

import "fmt"

func main() {
    var a int = 10

    fmt.Println("Before:", a)

    changeValue(&a)

    fmt.Println("After:", a)
}

func changeValue(number *int) {
    *number = 20
}
```

```
Before: 10
After: 20
```

