



Defer & Exit⁺

Defer #1

Defer merupakan sebuah keyword pada bahasa Go yang digunakan untuk memanggil sebuah *function* yang dimana proses eksekusi akan di tahan hingga block sebuah *function* selesai.

Agar lebih mudah dipahami, mari perhatikan pada gambar disebelah kanan.

Terdapat 3 buah function *fmt.Println* yang akan menampilkan text yang berbeda-beda. Namun function *fmt.Println* pertama dipanggil dengan menggunakan *defer*.

Jika dijalankan maka hasilnya akan seperti pada gambar kedua.

Perhatikan bahwa text yang berada pada posisi paling bawah gambar kedua merupakan text yang ditampilkan menggunakan *defer* pada line 6 di gambar pertama.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     defer fmt.Println("defer function starts to execute")
7     fmt.Println("Hai everyone")
8     fmt.Println("Welcome back to Go learning center")
9 }
```

```
Hai everyone
Welcome back to Go learning center
defer function starts to execute
```



Defer #2

Perhatikan pada gambar di bawah. Terdapat sebuah *function* bernama *callDeferFunc* yang digunakan untuk memanggil *function* *deferFunc* dengan keyword *defer*.

Kemudian *function* *callDeferFunc* dipanggil di baris paling atas *function* *main* yang kemudian diikuti sebuah pemanggilan *function* *fmt.Println*.

```
13 package main
14
15 import "fmt"
16
17 func main() {
18     callDeferFunc()
19     fmt.Println("Hai everyone!!")
20 }
21
22 func callDeferFunc() {
23     defer deferFunc()
24 }
25
26 func deferFunc() {
27     fmt.Println("Defer func starts to execute")
28 }
```



Defer #2

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

Perhatikan bahwa tulisan `Defer func starts to execute` yang berasal dari function `deferFunc` dipanggil menggunakan keyword `defer`, ditampilkan lebih dulu daripada tulisan `Hai everyone!!` yang berasal dari function `fmt.Println`.

Ini terjadi karena `deferFunc` dipanggil didalam block function `callDeferFunc`, bukan didalam block function `main`.

Maka dari itu pemanggilan function `deferFunc` akan di eksekusi setelah function `callDeferFunc` telah mengeksekusi seluruh statement atau syntax nya.

```
Defer func starts to execute  
Hai everyone!!
```



Exit

Function `exit` yang berasal dari package `os` berguna untuk menghentikan suatu program secara paksa. Perhatikan hasil eksekusi syntax dari gambar pertama pada gambar kedua.

Tulisan `Invoke with defer` tidak ditampilkan.

Ini karena setelah function `fmt.Println` pada line 44 dijalankan, program di hentikan secara paksa oleh function `os.Exit` ditengah jalan.

Function `os.Exit` menerima satu parameter berupa nilai dengan tipe data `int` yang digunakan sebagai status exit nya.

```
35 package main
36
37 import (
38     "fmt"
39     "os"
40 )
41
42 func main() {
43     defer fmt.Println("Invoke with defer")
44     fmt.Println("Before Exiting")
45     os.Exit(1)
46 }
47
```

```
Defer func starts to execute
Hai everyone!!
```

