



Exported & Unexported⁺

Exported & Unexported- Sesi 3

Exported & Unexported

Pada bahasa Go, tiap folder yang berbeda akan dianggap sebagai suatu package.

Kita dapat menggunakan berbagai variable ataupun tipe data dari package lainnya asalkan variable atau tipe data lainnya tersebut telah ter-eksport dari package nya.

Kemudian cara kita meng-eksport suatu variable atau suatu tipe data adalah dengan mengawali penulisan variable maupun tipe data lainnya dengan huruf kapital atau upper case.

Untuk itu sekarang mari kita coba mempraktikkannya dengan membuat satu folder dengan nama belajar-go, lalu buatlah sebuah file dengan nama main.go dan sebuah folder bernama helpers di dalam folder belajar-go. Jika sudah maka *folder tree* kita akan terlihat seperti gambar dibawah ini.



Exported & Unexported- Sesi 3

Exported & Unexported

Kemudian buatlah 3 buah file di dalam folder helpers dengan nama `function.go` , `struct.go` dan `method.go`. Lalu pada file `function.go`, buatlah 2 buah *function* seperti pada gambar pertama dibawah. Lalu pada file `struct.go`, buatlah sebuah *struct* seperti pada gambar kedua dibawah. Dan yang terakhir pada file `method.go`, buatlah sebuah *method* seperti pada gambar ketiga dibawah.

```
1 package helpers
2
3 import "fmt"
4
5 func Greet() {
6     fmt.Println("Hallo dari function Greet")
7 }
8
9 func greet() {
10    fmt.Println("Hallo dari function greet")
11 }
```

```
1 package helpers
2
3 type Person struct {
4     Name    string
5     Age     int
6     Address string
7 }
8
```

```
1 package helpers
2
3 func (p Person) Invokegreet() {
4     greet()
5 }
```

Exported & Unexported

Jika kita perhatikan pada 3 gambar di halaman sebelumnya, pada line 1 dari ketiga gambar tersebut selalu ada tertulis `package helpers`.

Tulisan tersebut harus ada karena 3 file yang terbuat pada folder `helpers` akan dianggap sebagai suatu kesatuan package dengan nama `helpers`.

Nama dari package nya lebih baik disamakan dengan nama folder nya, maka dari itu karena file `struct.go`, `function.go` dan `method.go` merupakan file-file yang terdapat dalam folder `helpers`, maka semua file-file tersebut akan dianggap sebagai bagian dari package `helpers`.

Berbeda dengan penulisan `package main` yang merupakan sebuah keharusan atau kewajiban pada setiap project Go untuk memiliki `package main`.

Penamaan `main` bukan karena project kita berada dalam folder dengan nama `main`, tetapi penamaan `main` memang sudah merupakan peraturan khusus dari Go untuk menuliskan package dengan nama `main`.

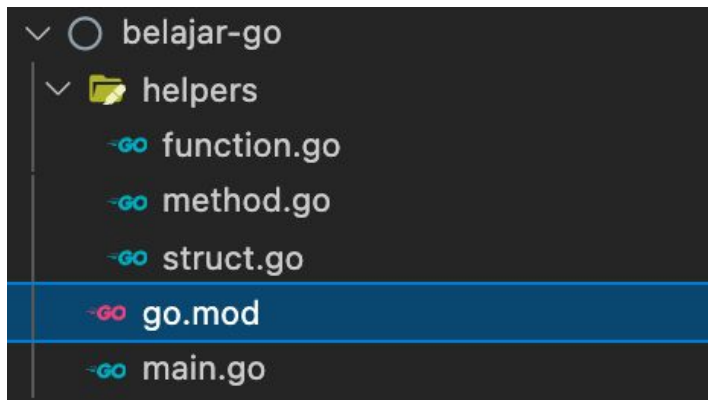


Exported & Unexported

Sekarang mari kita gunakan *struct*, *method* maupun *function* dari package helpers pada file main.go, tetapi sebelum itu mari kita buat file go.mod dengan menjalankan perintah go mod init belajar-go pada terminal kita. Jangan lupa untuk membuka terminal nya pada root folder dari folder belajar-go.

```
go mod init belajar-go
```

Setelah kita menjalankan perintah seperti pada gambar diatas, maka file go.mod akan otomatis terbuat dan folder tree kita akan terlihat seperti pada gambar dibawah ini. Perlu diingat bahwa penamaan *go modules* lebih baik disamakan dengan folder project kita.



Exported & Unexported

Mari kita coba untuk memanggil function *greet* dan *Greet* dari package *helpers* pada file *main.go*. Contohnya seperti pada gambar di sebelah kanan.

Karena function *greet* dan *Greet* merupakan function-function dari package *helpers*, maka dari itu kita perlu mengimport nya terlebih dahulu dari package *helpers*. Maksud dari penulisan "*belajar-go/helpers*" pada line 3 adalah kita mencoba untuk meng-import package *helpers* yang berada dalam package *belajar-go* yang merupakan nama dari *go modules* pada project kita kali ini.

Lalu cara kita mengakses function-function dari package *helpers* adalah seperti pada line 7 dan 9. Kita perlu menuliskan nama package terlebih dahulu kemudian diikuti dengan tanda titik lalu nama *function* nya (*helpers.Greet()*).

Lalu perhatikan pada line 9 dimana terjadi error pada saat compile time. Error ini terjadi karena function *greet* bukan merupakan suatu *function* yang ter-eksport dari package *helpers* karena penulisan *function* nya dimulai dari huruf kecil, maka otomatis *function* *greet* tidak ter-eksport.

```
1 package main
2
3 import "belajar-go/helpers"
4
5 func main() {
6
7     helpers.Greet()
8
9     helpers.greet()
10 }
```



Exported & Unexported

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

```
1 package main
2
3 import "belajar-go/helpers"
4
5 func main() {
6
7     helpers.Greet()
8
9     helpers.greet()
10 }
```

```
# command-line-arguments
./main.go:9:2: cannot refer to unexported name helpers.greet
```



Exported & Unexported

Jika kita lihat kembali pada file `method.go` yang berada dalam folder `helpers`, terdapat sebuah method bernama `Invokegreet` yang digunakan untuk memanggil function `greet` yang tidak ter-eksport dari package `helpers`.

Maka dari itu, jika kita ingin memanggil function `greet` dari file `main.go`, maka kita perlu memanggil method `Invokegreet`. Untuk itu mari kita buat sebuah variable dengan nama `person` pada file `main.go`. Lalu variable tersebut akan memiliki tipe data dari `struct Person` yang berasal dari package `helpers`.

Contohnya seperti pada gambar kedua di sebelah kanan.

Perhatikan pada line 11 dari gambar kedua, kita memanggil method `Invokegreet` dan tidak terjadi error apapun. Ketika kita jalankan pada terminal, maka hasilnya akan seperti pada gambar ketiga.

Tulisan "Hallo dari function greet" berasal dari function `greet` yang telah terpanggil dari method `Invokegreet`.

```
1 package helpers
2
3 func (p Person) Invokegreet() {
4     greet()
5 }
6
```

```
1 package main
2
3 import "belajar-go/helpers"
4
5 func main() {
6
7     helpers.Greet()
8
9     var person = helpers.Person{}
10
11     person.Invokegreet()
12 }
```

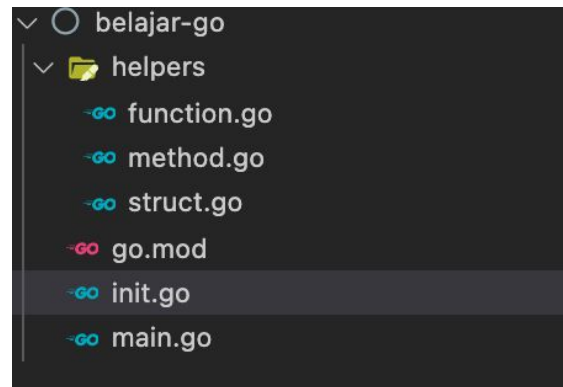
```
Hallo dari function Greet
Hallo dari function greet
```



Init function

Satu hal lagi yang perlu kita ketahui yaitu terdapat suatu *function* yang akan dieksekusi terlebih dahulu sebelum *function main*. *Function* tersebut bernama *init*. Mari kita buat suatu file bernama *Init.go* pada root direktori kita. Jika sudah maka folder tree kita hingga saat ini akan terlihat seperti pada gambar pertama di sebelah kanan.

Lalu pada file *init.go*, buatlah sebuah *function* dengan nama *init* seperti pada gambar kedua di sebelah kanan. Perhatikan line 1 pada gambar kedua. Karena file *init.go* merupakan file yang berada dalam root direktori project kita, maka file *init.go* akan dianggap sebagai bagian dari package *main*. Maka dari itu, sekarang bukan hanya file *main.go* saja yang perlu menuliskan *package main*, tetapi juga berlaku untuk file *init.go*.



```
1 package main
2
3 import "fmt"
4
5 func init() {
6     fmt.Println("Hallo!!!! Ini berasal dari function init")
7 }
8
```

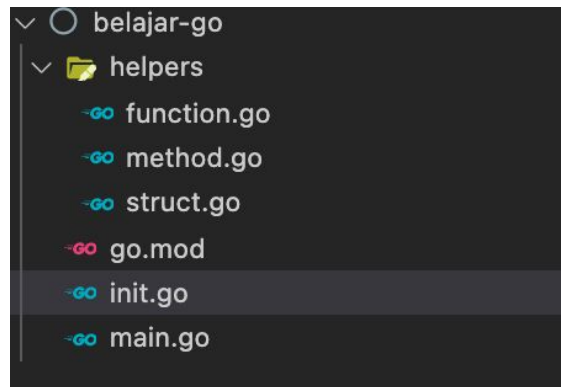
```
Hallo!!!! Ini berasal dari function init
Hallo dari function Greet
Hallo dari function greet
```



Init function

Jika kita ingin menjalankan seluruh file yang berada di root direktori kita, maka kita menggunakan perintah `go run *.go`. Jika sudah dijalankan maka hasilnya akan seperti pada gambar ketiga.

Bisa kita lihat dari hasilnya, tulisan pertama yang berada pada gambar ketiga merupakan tulisan yang berasal dari file `init.go` yang menggunakan function `init`. Maka ini membuktikan bahwa function bernama `init` akan otomatis dieksekusi terlebih dahulu oleh bahasa Go sebelum function `main`.



```
1 package main
2
3 import "fmt"
4
5 func init() {
6     fmt.Println("Hallo!!!! Ini berasal dari function init")
7 }
8
```

```
go run *.go
Hallo!!!! Ini berasal dari function init
Hallo dari function Greet
Hallo dari function greet
```

