Go-Lang Embed

Eko Kurniawan Khannedy

Eko Kurniawan Khannedy

- Technical architect at one of the biggest ecommerce company in Indonesia
- 10+ years experiences
- www.programmerzamannow.com
- youtube.com/c/ProgrammerZamanNow

Eko Kurniawan Khannedy

- Telegram : @khannedy
- Facebook : fb.com/ProgrammerZamanNow
- Instagram : instagram.com/programmerzamannow
- Youtube : youtube.com/c/ProgrammerZamanNow
- Telegram Channel : t.me/ProgrammerZamanNow
- Email : echo.khannedy@gmail.com

Sebelum Belajar

- Go-Lang Dasar
- Go-Lang Modules
- Go-Lang Unit Test
- https://www.udemy.com/course/pemrograman-go-lang-pemula-sampai-mahir/?referralCode=C9C831DC7A42D8714259

Agenda

- Pengenalan Embed Package
- Embed File ke String
- Embed File ke Byte[]
- Embed Multiple File
- Hasil Embed di Compile
- Dan lain-lain

Pengenalan Embed Package

Embed Package

- Sejak Golang versi 1.16, terdapat package baru dengan nama embed
- Package embed adalah fitur baru untuk mempermudah membaca isi file pada saat compile time secara otomatis dimasukkan isi file nya dalam variable
- https://golang.org/pkg/embed/

Cek Versi Golang

go version

Cara Embed File

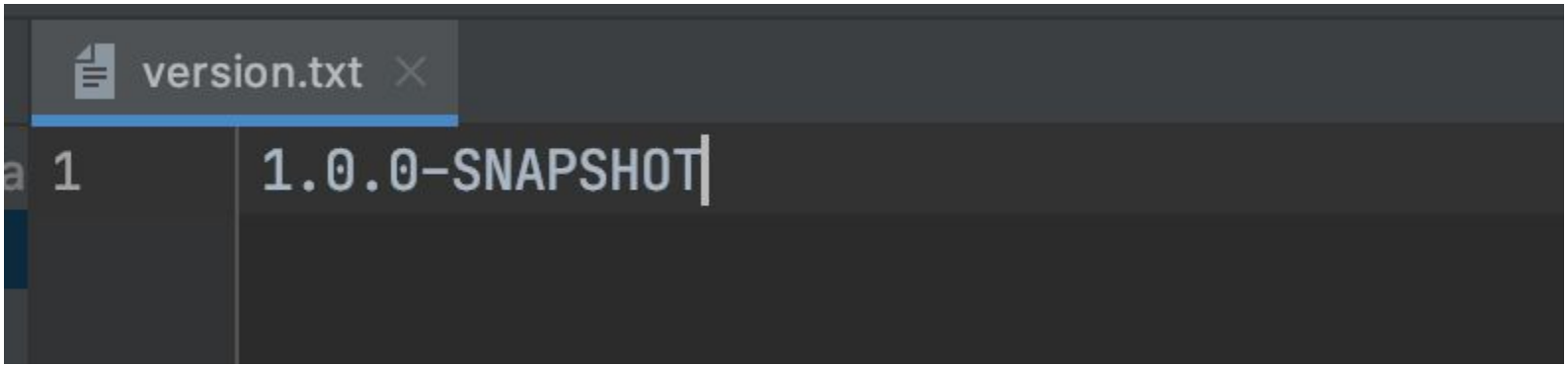
- Untuk melakukan embed file ke variable, kita bisa mengimport package embed terlebih dahulu
- Selajutnya kita bisa tambahkan komentar //go:embed diikuti dengan nama file nya, diatas variable yang kita tuju
- Variable yang dituju tersebut nanti secara otomatis akan berisi konten file yang kita inginkan secara otomatis ketika kode golang di compile
- Variable yang dituju tidak bisa disimpan di dalam function

Embed File ke String

Embed File ke String

- Embed file bisa kita lakukan ke variable dengan tipe data string
- Secara otomatis isi file akan dibaca sebagai text dan masukkan ke variable tersebut

Kode : File version.txt



Kode : Embed File ke String

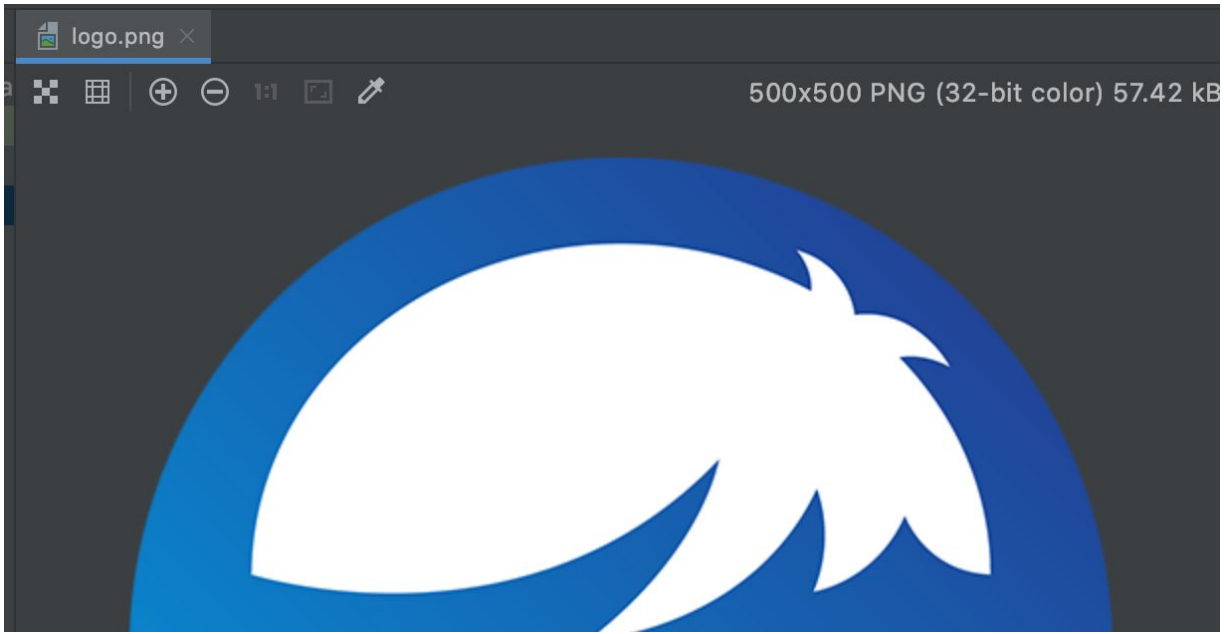
```go
import (
    _ "embed"
    "fmt"
    "testing"
)

//go:embed version.txt
var version string

func TestString(t *testing.T) {
    fmt.Println(version)
}
```

Embed File ke Byte[]

Embed File ke []byte

- Selain ke tipe data String, embed file juga bisa dilakukan ke variable tipe data []byte
- Ini cocok sekali jika kita ingin melakukan embed file dalam bentuk binary, seperti gambar dan lain-lain

Gambar Logo



Kode : Embed File ke Byte[]

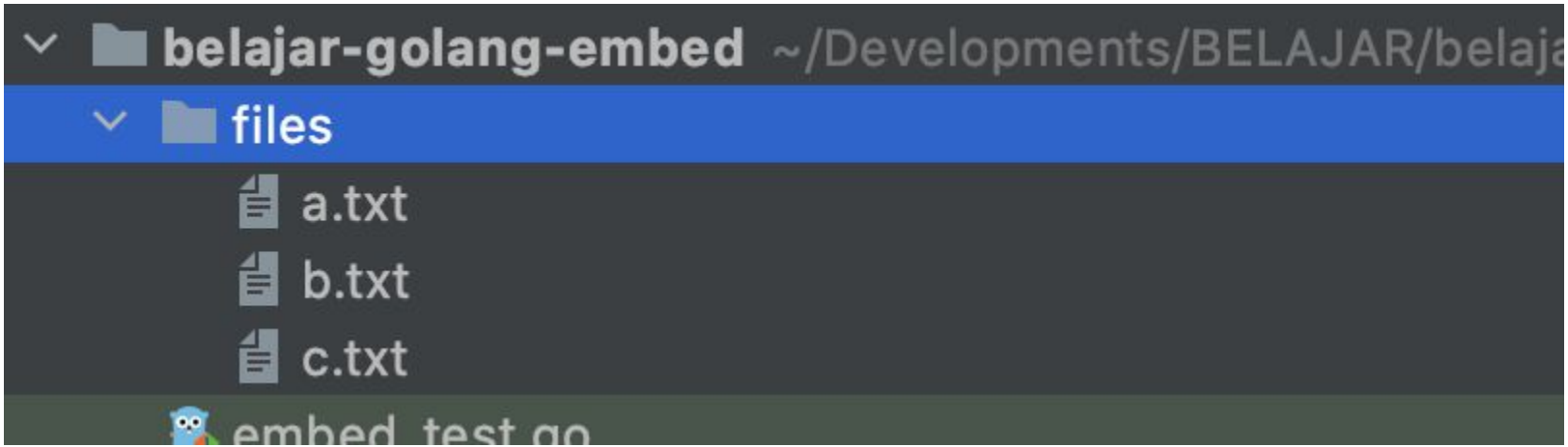
```go
//go:embed logo.png
var logo []byte

func TestByteArray(t *testing.T) {
    err := ioutil.WriteFile("logo_next.png", logo, fs.ModePerm)
    if err != nil {
        panic(err)
    }
}
```

Embed Multiple Files

Embed Multiple Files

- Kadang ada kebutuhan kita ingin melakukan embed beberapa file sekaligus
- Hal ini juga bisa dilakukan menggunakan embed package
- Kita bisa menambahkan komentar //go:embed lebih dari satu baris
- Selain itu variable nya bisa kita gunakan tipe data embed.FS

Kode : Multiple Files



Kode : Embed Multiple File

```go
//go:embed files/a.txt
//go:embed files/b.txt
//go:embed files/c.txt
var files embed.FS

func TestMultipleFiles(t *testing.T) {
    a, _ := files.ReadFile("files/a.txt")
    fmt.Println(string(a))

    b, _ := files.ReadFile("files/b.txt")
    fmt.Println(string(b))

    c, _ := files.ReadFile("files/c.txt")
    fmt.Println(string(c))
}
```

Path Matcher

Patch Matcher

- Selain manual satu per satu, kita bisa mengguakan patch matcher untuk membaca multiple file yang kita inginkan
- Ini sangat cocok ketika misal kita punya pola jenis file yang kita inginkan untuk kita baca
- Caranya, kita perlu menggunakan path matcher seperti pada package function path.Match

https://golang.org/pkg/path/#Match

func Match

```
func Match(pattern, name string) (matched bool, err error)
```

Match reports whether name matches the shell pattern. The pattern syntax is:

```
pattern:
        { term }
term:
        '*'         matches any sequence of non-/ characters
        '?'         matches any single non-/ character
        '[' [ '^' ] { character-range } ']'
                    character class (must be non-empty)
        c           matches character c (c != '*', '?', '\\', '[')
        '\\' c      matches character c

character-range:
        c           matches character c (c != '\\', '-', ']')
        '\\' c      matches character c
        lo '-' hi   matches character c for lo <= c <= hi
```

Kode : Patch Matcher

```go
//go:embed files/*.txt
var path embed.FS

func TestPathMatcher(t *testing.T) {
    dir, _ := path.ReadDir("files")
    for _, entry := range dir {
        if !entry.IsDir() {
            fmt.Println(entry.Name())
            content, _ := path.ReadFile("files/" + entry.Name())
            fmt.Println("Content:", string(content))
        }
```

Hasil Embed di Compile

Hasil Embed di Compile

- Perlu diketahui, bahwa hasil embed yang dilakukan oleh package embed adalah permanent dan data file yang dibaca disimpan dalam binary file golang nya
- Artinya bukan dilakukan secara realtime membaca file yang ada diluar
- Hal ini menjadikan jika binary file golang sudah di compile, kita tidak butuh lagi file external nya, dan bahkan jika diubah file external nya, isi variable nya tidak akan berubah lagi

Kode : Main Function

```go
//go:embed version.txt
var version string

//go:embed logo.png
var logo []byte

//go:embed files/*.txt
var path embed.FS

func main() {
    fmt.Println(version)

    ioutil.WriteFile("logo_next.png", logo, fs.ModePerm)
```

Compile

go build

Materi Selanjutnya

Materi Selanjutnya

- Go-Lang Web