

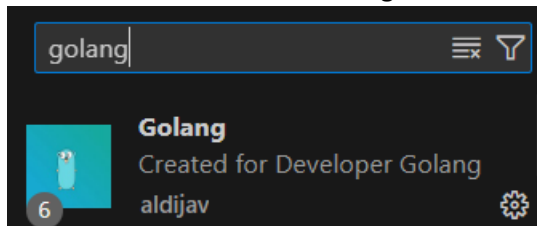
Modul Fiber 1

Persiapan:

1. Silahkan buat sebuah directory sebagai root directory project anda, beri nama “project-crud”.
Contoh, anda meletakkan root directory pada C://golang/project-crud

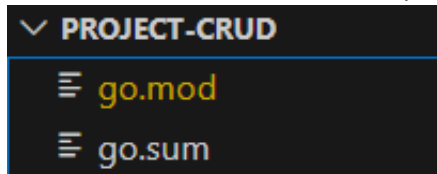
```
PS C:\golang> mkdir project-crud
PS C:\golang> cd .\project-crud\
PS C:\golang\project-crud> go mod init project-crud
PS C:\golang\project-crud> go get github.com/gofiber/fiber/v2
PS C:\golang\project-crud> code .
```

2. Install vscode extension “Golang”

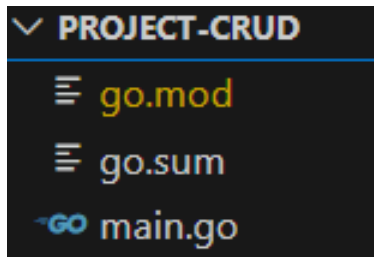


Project-crud

1. Pada saat ini isi dari root directory anda adalah:



2. Silahkan tambahkan file baru “main.go” pada root directory anda



3. Isi main.go dengan code berikut:

```
1 package main
2
3 import "github.com/gofiber/fiber/v2"
4
5 func main() {
6     //Fiber instance
7     app := fiber.New()
8
9     //Routes
10    app.Get("/", func(c *fiber.Ctx) error {
11        return c.JSON(fiber.Map{
12            "message": "Hello, World!",
13        })
14    })
15
16    //Start server
17    app.Listen(":3000")
18 }
```

Perhatikan cara anda menuliskan diatas. Anda tidak perlu mengetikan line 3, jika extension golang anda terinstall, line 3 seharusnya akan secara otomatis terisi atau melalui fitur "fix it".

4. Kembali ke terminal anda, compile dan run script go anda.

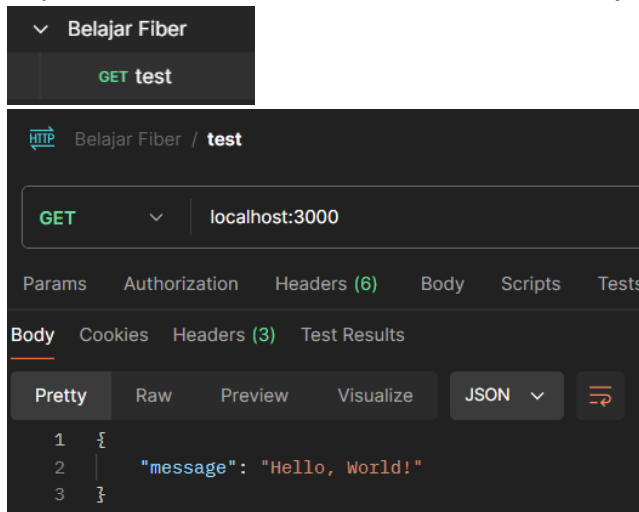
```
PS C:\golang\project-crud> go run main.go

Fiber v2.52.5
http://127.0.0.1:3000
(bound on host 0.0.0.0 and port 3000)

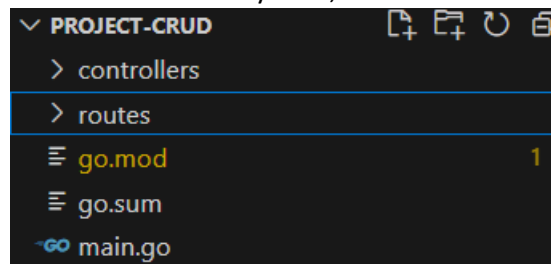
Handlers ..... 2   Processes ..... 1
Prefork ..... Disabled   PID ..... 27488
```

Pilih "Allow" jika firewall anda meminta permission untuk menjalankan script exe anda!

5. Silahkan buka aplikasi “Postman” anda. Buatlah sebuah collection baru dan isi dengan satu request baru. Beri nama collection baru anda “Belajar Fiber”!



6. Tata ulang code anda, sehingga script anda terbagi kedalam 2 modul, yaitu routes dan controllers.
- a. Buatlah dua directory baru, routes dan controllers, didalam root directory anda.



- b. Buat file myController.go didalam directory controllers dan isi dengan code berikut:

```
1 package controllers
2
3 import "github.com/gofiber/fiber/v2"
4
5 func HomeFunc(c *fiber.Ctx) error {
6     return c.JSON(fiber.Map{
7         "message": "Hello, World baru nih!",
8     })
9 }
```

Perhatikan, nama function harus diawali dengan huruf kapital. Hal ini merupakan semantic dari go, yaitu jika function akan diakses diluar package nya, maka huruf pertama function harus huruf kapital.

Ubah isi dari message!

- c. Buat file route.go didalam directory routes dan isi dengan code berikut:

```
1 package routes
2
3 import (
4     "project-crud/controllers"
5
6     "github.com/gofiber/fiber/v2"
7 )
8
9 func RouterApp(app *fiber.App) {
10     app.Get("/", controllers.HomeFunc)
11 }
```

Line 3 – 7 seharusnya terisi secara otomatis

- d. Ubah isi dari main.go sebagai berikut:

```
1 package main
2
3 import (
4     "project-crud/routes"
5     "github.com/gofiber/fiber/v2"
6 )
7
8 func main() {
9     //Fiber instance
10    app := fiber.New()
11
12    //Routes
13    routes.RouterApp(app)
14
15    //Start server
16    app.Listen(":3000")
17 }
```

7. Uji point 6 menggunakan postman!

8. Grouping Routes.

- a. Ubah isi dari routes/route.go sebagai berikut:

```
1 package routes
2
3 import (
4     "project-crud/controllers"
5
6     "github.com/gofiber/fiber/v2"
7 )
8
9 func RouterApp(app *fiber.App) {
10     api := app.Group("/api")
11     api.Get("/", controllers.HomeFunc)
12 }
```

Perhatikan variable route **api!!**

9. Uji point 8 menggunakan postman! Apa url yang baru? _____

10. Selanjutnya Project-crud akan dibangun dengan struktur directory yang mendekati MVC

```
- project-crud
|
|- models
|- controllers
|- routers
|- config
|- main.go
|- ....
```

11. Koneksikan dengan MongoDB

a. Install mongoDB driver pada project anda

```
PS C:\golang\project-crud> go get go.mongodb.org/mongo-driver/mongo
```

b. Buatlah DB dan collection baru:

DB: unairsatu

Collection : users

c. Ambil url untuk koneksi ke MongoDB

Connect to your application



Connecting with MongoDB Driver

1. Select your driver and version

We recommend installing and using the latest driver version.

Driver	Version
Go	1.6 or later

2. Install your driver

Run the following on the command line

```
go get go.mongodb.org/mongo-driver/mongo
```

[View MongoDB Go Driver installation instructions.](#)

3. Add your connection string into your application code

Use this connection string in your application

☐ View full code sample

```
mongodb+srv://belajarnodejs:<db_password>@cluster0.aiolomr.mongodb.net/?
retryWrites=true&w=majority&appName=Cluster0
```

d. Buat Directory “config” pada root directory anda dan buatlah file “database.go” didalamnya.

(Anda boleh menggunakan code lain, tidak harus sesuai dengan modul. Silahkan googling atau gunakan chatGPT atau AI sejenisnya)

Project-crud/config/database.go

```

1  package config
2
3  import (
4      "context"
5      "log"
6      "time"
7
8      "go.mongodb.org/mongo-driver/mongo"
9      "go.mongodb.org/mongo-driver/mongo/options"
10 )
11
12 var DB *mongo.Database
13
14 func ConnectDB() {
15     log.Println("Connecting to MongoDB...")
16
17     ctx, cancel := context.WithTimeout(context.Background(), 10*time.Second)
18     defer cancel()
19
20     client, err := mongo.Connect(ctx, options.Client().ApplyURI("mongodb://localhost:27017"))
21     if err != nil {
22         log.Fatal("Error connecting to MongoDB:", err)
23     }
24
25     DB = client.Database("unairsatu")
26     log.Println("Connected to MongoDB!")
27 }
28
29 func GetCollection(collectionName string) *mongo.Collection {
30     // ConnectDB()
31
32     if DB == nil {
33         // log.Fatal("Database connection is not initialized")
34         ConnectDB()
35     }
36
37     return DB.Collection(collectionName)
38 }

```

- e. Ubah main.go anda untuk test koneksi ke MongoDB.

```

1  package main
2
3  import (
4      "project-crud/config"
5  )
6
7  func main() {
8      //Fiber instance
9      // app := fiber.New()
10
11      //Routes
12      // routes.RouterApp(app)
13
14      //Start server
15      // app.Listen(":3000")
16
17      config.ConnectDB()
18
19  }

```

- f. Compile dan run

```
PS C:\golang\project-crud> go run main.go
2024/10/22 22:15:24 Connecting to MongoDB...
2024/10/22 22:15:24 Connected to MongoDB!
```

Hasil diatas menunjukan bahwa, code anda dapat membangun koneksi ke MongoDB

- g. Kembalikan isi main.go seperti sebelumnya:

```
1 package main
2
3 import (
4     "project-crud/routes"
5
6     "github.com/gofiber/fiber/v2"
7 )
8
9 func main() {
10     //Fiber instance
11     app := fiber.New()
12
13     //Routes
14     routes.RouterApp(app)
15
16     //Start server
17     app.Listen(":3000")
18
19 }
```

12. Buat Model

- a. Buat directory "model" didalam root directory dan isi dengan file users.go
project-crud/model/users.go

```
1 package models
2
3 import "go.mongodb.org/mongo-driver/bson/primitive"
4
5 type User struct {
6     ID          primitive.ObjectID `json:"id,omitempty"`
7     Username    string             `json:"username"`
8     Nm_user     string             `json:"nm_user"`
9     Pass        string             `json:"pass"`
10    Email        string             `json:"email"`
11    Role_aktif   int8               `json:"role_aktif"`
12    Created_at   primitive.DateTime `json:"created_at"`
13    Jenis_kelamin int8               `json:"jenis_kelamin"`
14    Photo        string             `json:"photo"`
15    Phone        string             `json:"phone"`
16    Token        string             `json:"token"`
17    Id_jenis_user int8               `json:"id_jenis_user"`
18    Pass_2       string             `json:"pass_2"`
19 }
```

Kita anulir beberapa kolom pada cdm yang diberikan pada soal UTS.

13. userController.go

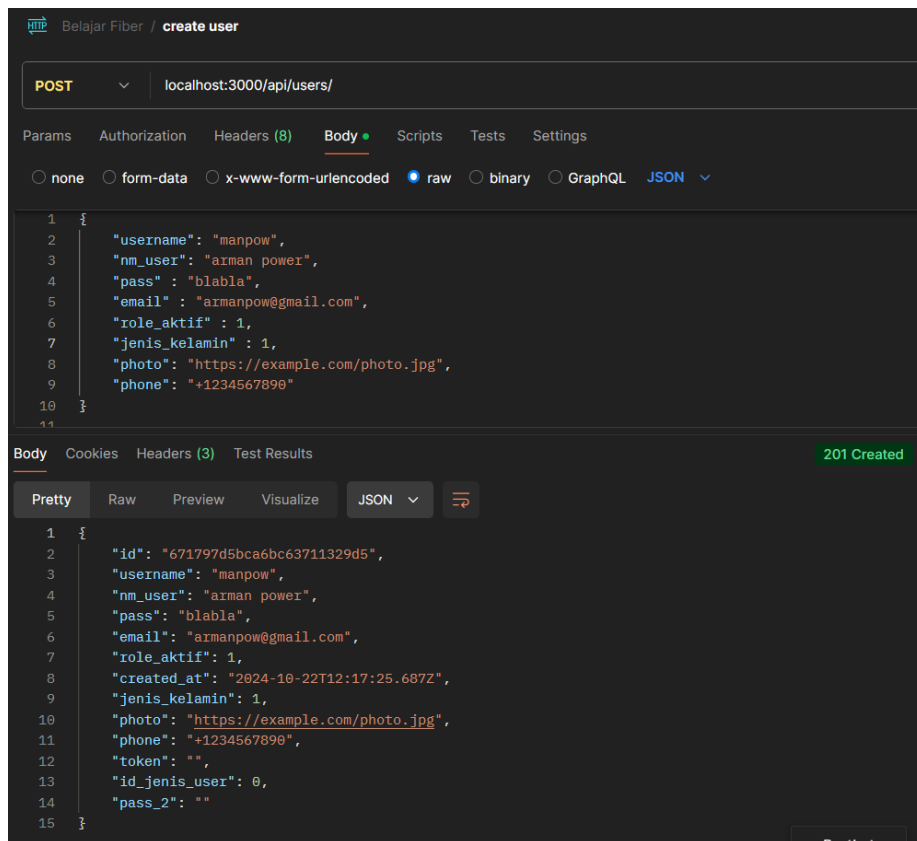
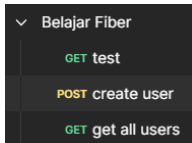
- a. Didalam directory “controllers” tambahkan file “userController.go”
Project-crud/controllers/userController.go

```
1 package controllers
2
3 import (
4     "context"
5     "net/http"
6     "time"
7
8     "github.com/gofiber/fiber/v2"
9     "go.mongodb.org/mongo-driver/bson"
10    "go.mongodb.org/mongo-driver/bson/primitive"
11    "go.mongodb.org/mongo-driver/mongo"
12
13    "project-crud/config"
14    "project-crud/models"
15 )
16
17 var userCollection *mongo.Collection = config.GetCollection("users")
18
19 // Create User
20 func CreateUser(c *fiber.Ctx) error {
21     ctx, cancel := context.WithTimeout(context.Background(), 10*time.Second)
22     defer cancel()
23
24     var user models.User
25     if err := c.BodyParser(&user); err != nil {
26         return c.Status(http.StatusBadRequest).JSON(fiber.Map{"error": err.Error()})
27     }
28
29     loc, err := time.LoadLocation("Asia/Jakarta")
30     if err != nil {
31         return c.Status(http.StatusInternalServerError).JSON(fiber.Map{"error": err.Error()})
32     }
33     user.Created_at = primitive.NewDateTimeFromTime(time.Now().In(loc))
34
35     newUser := models.User{
36         ID:          primitive.NewObjectID(),
37         Username:    user.Username,
38         Nm_user:     user.Nm_user,
39         Pass:        user.Pass,
40         Email:       user.Email,
41         Role_aktif:  user.Role_aktif,
42         Created_at:  user.Created_at,
43         Jenis_kelamin: user.Jenis_kelamin,
44         Photo:       user.Photo,
45         Phone:       user.Phone,
46     }
47
48     _, errIns := userCollection.InsertOne(ctx, newUser)
49     if errIns != nil {
50         return c.Status(http.StatusInternalServerError).JSON(fiber.Map{"error": errIns.Error()})
51     }
52
53     return c.Status(http.StatusCreated).JSON(newUser)
54 }
55
56 // Get All Users
57 func GetUsers(c *fiber.Ctx) error {
58     ctx, cancel := context.WithTimeout(context.Background(), 10*time.Second)
59     defer cancel()
60
61     var users []models.User
62     cursor, err := userCollection.Find(ctx, bson.M{})
63     if err != nil {
64         return c.Status(http.StatusInternalServerError).JSON(fiber.Map{"error": err.Error()})
65     }
66
67     if err = cursor.All(ctx, &users); err != nil {
68         return c.Status(http.StatusInternalServerError).JSON(fiber.Map{"error": err.Error()})
69     }
70
71     return c.Status(http.StatusOK).JSON(users)
72 }
```

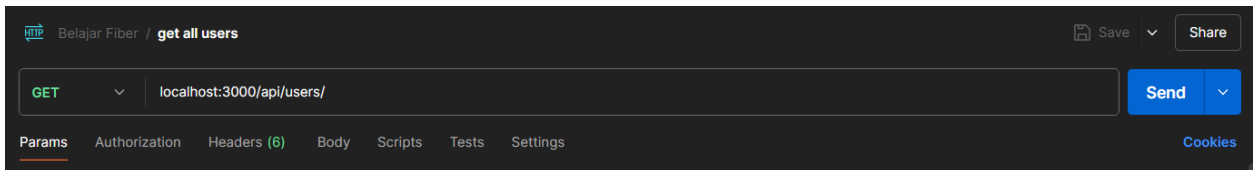

14. Tambahkan route

```
9 func RouterApp(app *fiber.App) {
10     api := app.Group("/api")
11     api.Get("/", controllers.HomeFunc)
12
13     users := api.Group("/users")
14     users.Post("/", controllers.CreateUser)
15     users.Get("/", controllers.GetUsers)
16 }
```

15. Test dengan Postman



Get All users



Tugas

1. Buatlah API untuk:
 - a. Select 1 user berdasarkan ID nya
 - b. Update data user berdasarkan ID nya
2. Test dan screenshot hasilnya. Yang anda screenshot:
 - a. Code anda
 - b. Request pada postman
 - c. Collection pada mongoDB sebelum dan sesudah API dipanggil

