

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS WEB
LATIHAN 10

Untuk Memenuhi Latihan Praktikum Pemrograman Berbasis Web



Dosen: Adi Wahyu Pribadi, S.Si., M.Kom.

Disusun oleh:

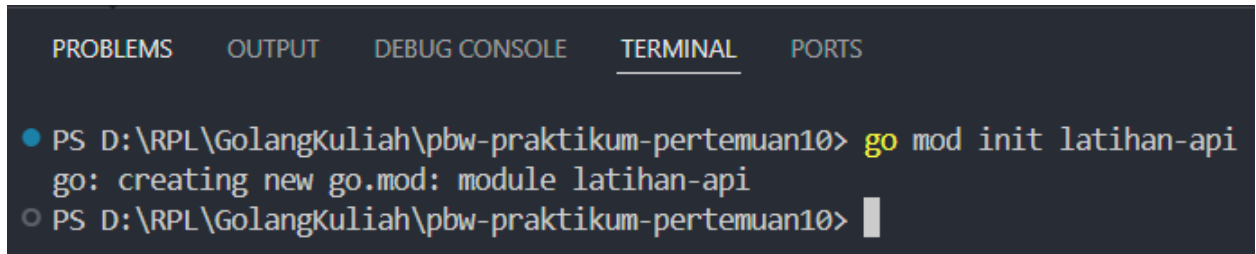
Nama : Muhammad Abduh Harry Malhotra
NIM : 4522210133
Semester : 4
Kelas : A

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PANCASILA
JAKARTA SELATAN
2023/2024

Link Github: <https://github.com/hotraa/PBW/>

Latihan Web API menggunakan Gorilla

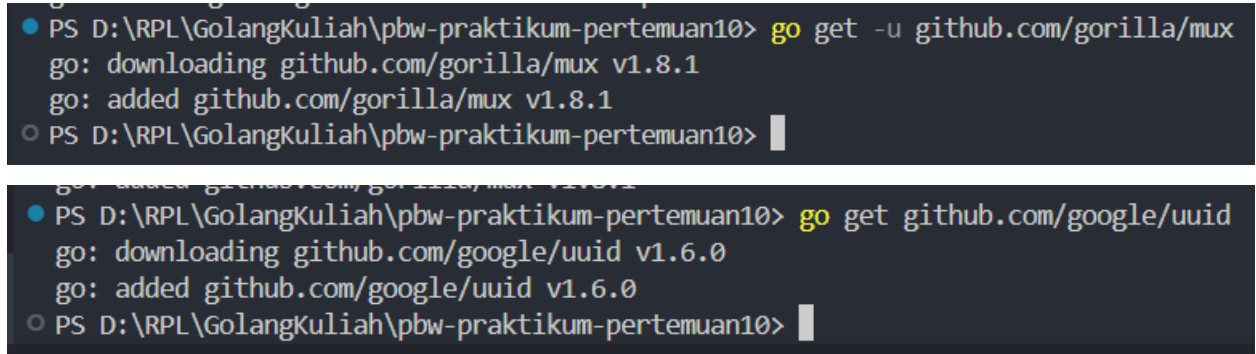
1. Initiate Project



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> go mod init latihan-api
go: creating new go.mod: module latihan-api
○ PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> 
```

2. Install Library Gorilla dan UUID



```
● PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> go get -u github.com/gorilla/mux
go: downloading github.com/gorilla/mux v1.8.1
go: added github.com/gorilla/mux v1.8.1
○ PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10>

● PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> go get github.com/google/uuid
go: downloading github.com/google/uuid v1.6.0
go: added github.com/google/uuid v1.6.0
○ PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> 
```

Jelaskan kegunaan Library Gorilla dan UUID

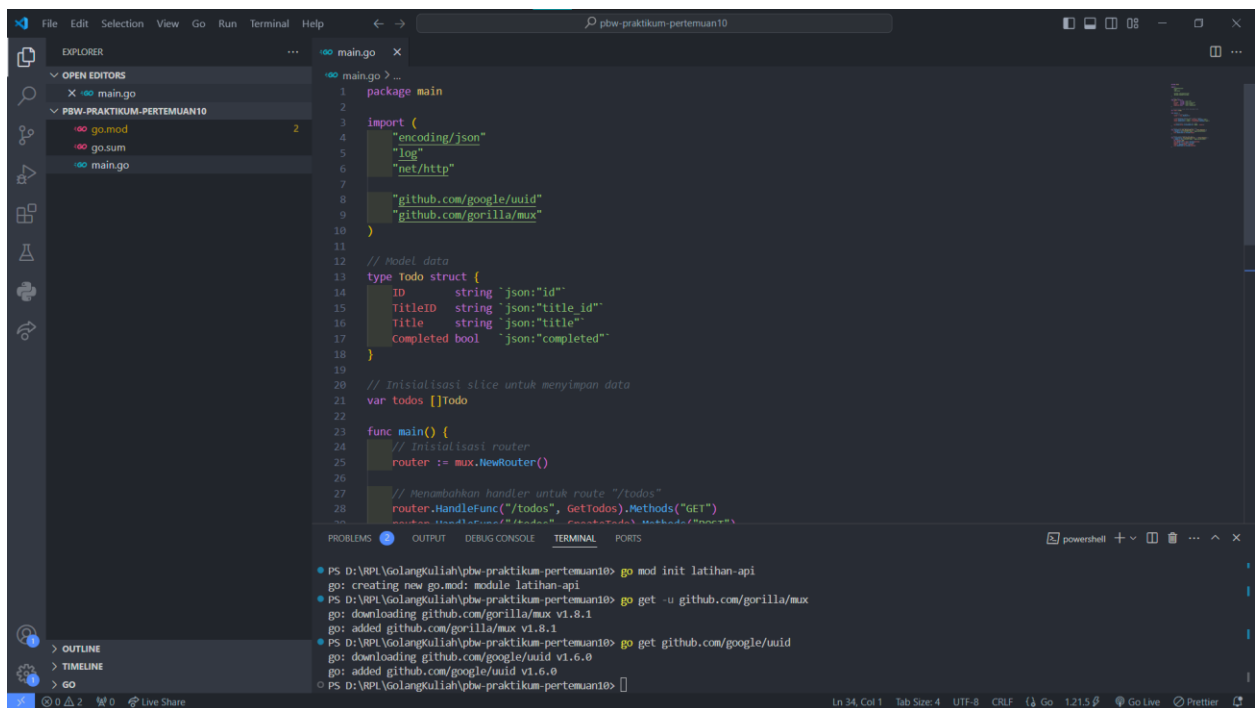
Gorilla adalah toolkit web untuk Golang yang terdiri dari beberapa paket utama yang digunakan dalam pengembangan aplikasi web:

- Gorilla Mux: Router HTTP yang kuat dan fleksibel untuk pencocokan URL. Mendukung pencocokan rute berdasarkan host URL, jalur, prefiks jalur, skema, nilai header dan query, metode HTTP, atau menggunakan pencocok kustom.
- Gorilla Sessions: Mengelola sesi pengguna dalam aplikasi web, termasuk penyimpanan sesi dalam cookie atau penyimpanan berbasis file atau database, membantu mengelola state pengguna dengan aman.
- Gorilla WebSocket: Implementasi protokol WebSocket yang memungkinkan komunikasi dua arah antara klien dan server, berguna untuk aplikasi real-time seperti chat atau notifikasi.

UUID (Universally Unique Identifier) adalah standar untuk pengidentifikasi yang unik secara global, sangat berguna dalam pengembangan web API untuk:

- Identifikasi Unik: Menghasilkan pengidentifikasi yang unik dan tidak mungkin bertabrakan, penting untuk identifikasi sumber daya dalam API seperti ID pengguna, produk, atau transaksi.
- Keamanan: Digunakan dalam konteks keamanan seperti token otentikasi atau kunci API, karena sulit diprediksi dan dihasilkan secara acak.
- Konsistensi Data: Memastikan setiap entitas memiliki identifikasi yang konsisten dan unik, mengurangi risiko duplikasi atau kebingungan dalam integrasi data dari berbagai sumber atau layanan.

3. Membuat file main.go



The screenshot shows the Visual Studio Code editor with a Go project named 'pbw-praktikum-pertemuan10'. The Explorer sidebar on the left shows the file structure with 'main.go' selected. The main editor displays the content of 'main.go', which includes package declarations, imports for JSON encoding, logging, and HTTP routing, a struct definition for 'Todo', and a 'main' function that initializes a router and sets up a GET endpoint for '/todos'. The terminal at the bottom shows the execution of Go commands to initialize the module, download dependencies (gorilla/mux and google/uuid), and run the application.

```
1 package main
2
3 import (
4     "encoding/json"
5     "log"
6     "net/http"
7
8     "github.com/google/uuid"
9     "github.com/gorilla/mux"
10 )
11
12 // Model data
13 type Todo struct {
14     ID          string `json:"id"`
15     TitleID     string `json:"title_id"`
16     Title       string `json:"title"`
17     Completed   bool   `json:"completed"`
18 }
19
20 // Inisialisasi slice untuk menyimpan data
21 var todos []Todo
22
23 func main() {
24     // Inisialisasi router
25     router := mux.NewRouter()
26
27     // Menambahkan handler untuk route "/todos"
28     router.HandleFunc("/todos", GetTodos).Methods("GET")
29 }
```

Terminal Output:

```
PS D:\VPL\Golangkuliah\pbw-praktikum-pertemuan10> go mod init latihan-api
go: creating new go.mod: module latihan-api
PS D:\VPL\Golangkuliah\pbw-praktikum-pertemuan10> go get -u github.com/gorilla/mux
go: downloading github.com/gorilla/mux v1.8.1
go: added github.com/gorilla/mux v1.8.1
PS D:\VPL\Golangkuliah\pbw-praktikum-pertemuan10> go get github.com/google/uuid
go: downloading github.com/google/uuid v1.6.0
go: added github.com/google/uuid v1.6.0
PS D:\VPL\Golangkuliah\pbw-praktikum-pertemuan10>
```

- Buat inialisasi slice melingkup router, handler router, server router

```

24 // Inialisasi router
25 router := mux.NewRouter()
26
27 // Menambahkan handler untuk route "/todos"
28 router.HandleFunc("/todos", GetTodos).Methods("GET")
29 router.HandleFunc("/todos", CreateTodo).Methods("POST")
30
31 // Menjalankan server pada port 8000
32 log.Fatal(http.ListenAndServe(":8000", router))
33 }
34
35 // Handler untuk mendapatkan semua todos
36 func GetTodos(w http.ResponseWriter, r *http.Request) {
37     w.Header().Set("Content-Type", "application/json")
38     json.NewEncoder(w).Encode(todos)
39 }

```

- Buat handler

```

41 // Handler untuk membuat todo baru
42 func CreateTodo(w http.ResponseWriter, r *http.Request) {
43     w.Header().Set("Content-Type", "application/json")
44     var newTodo Todo
45     json.NewDecoder(r.Body).Decode(&newTodo)
46     newTodo.ID = uuid.New().String()
47     todos = append(todos, newTodo)
48     json.NewEncoder(w).Encode(newTodo)
49 }

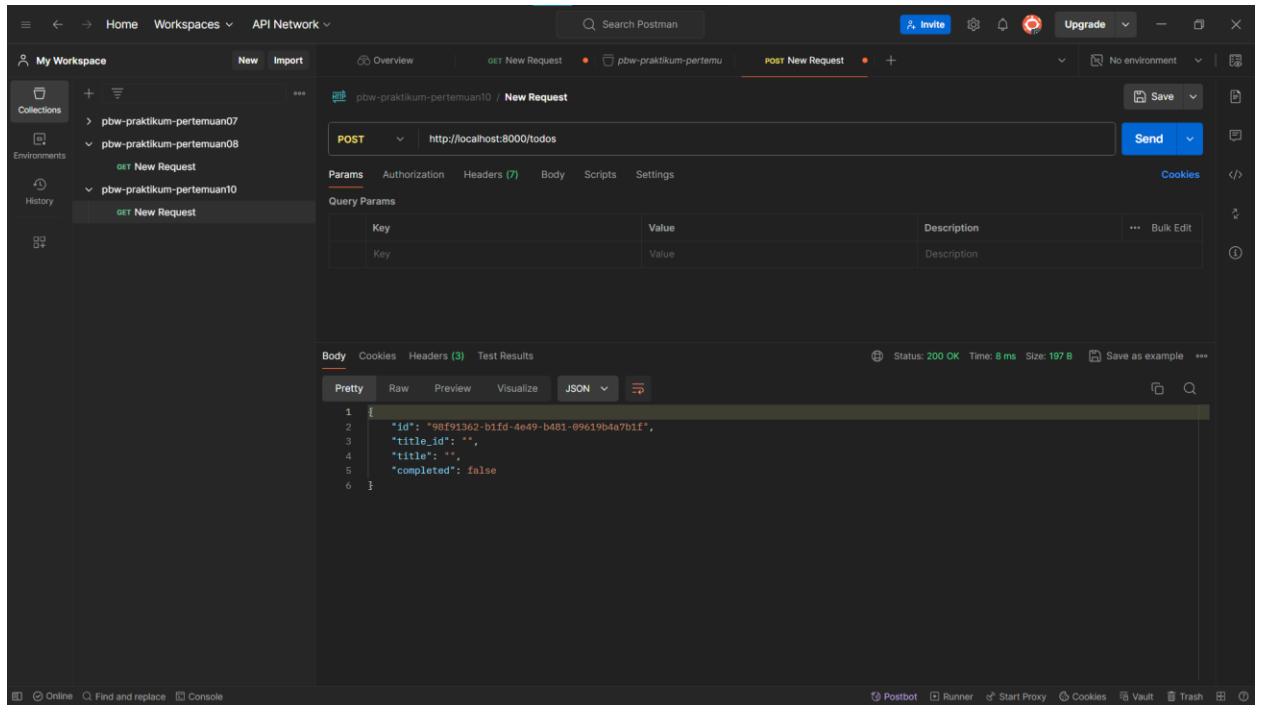
```

4. Jalankan local dan buka postman

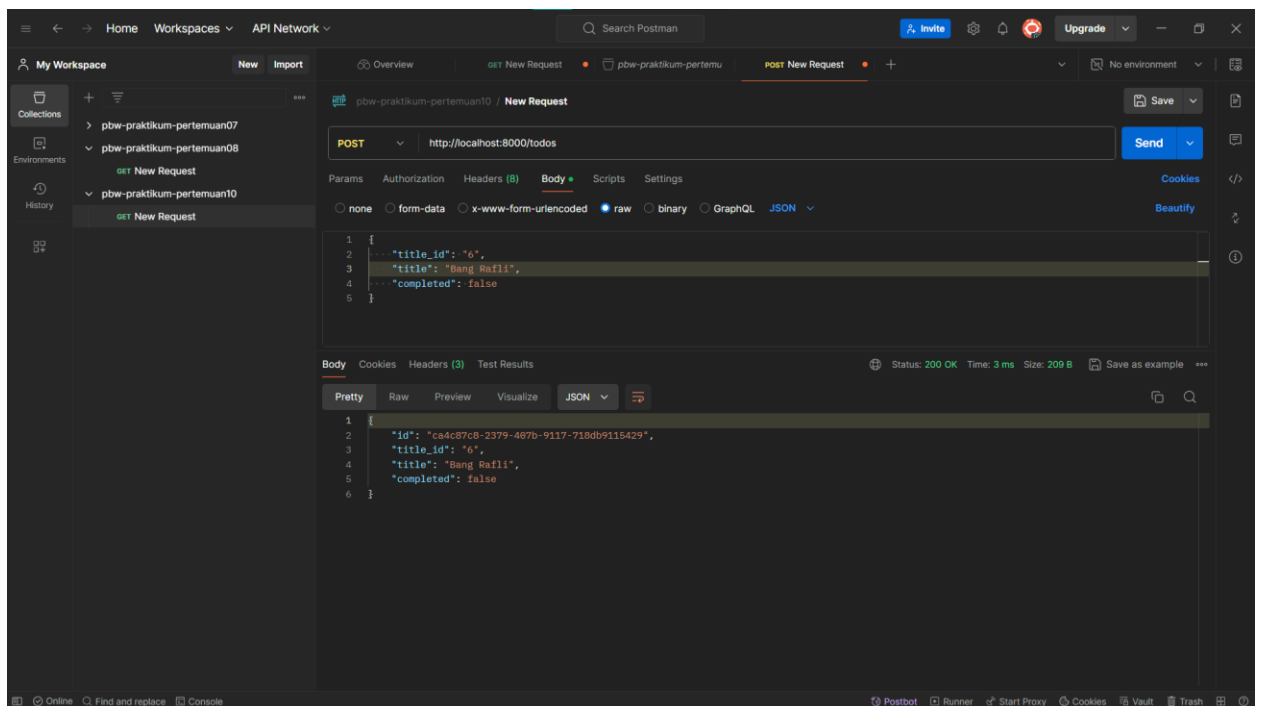
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
• PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> go mod init latihan-api
go: creating new go.mod: module latihan-api
• PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> go get -u github.com/gorilla/mux
go: downloading github.com/gorilla/mux v1.8.1
go: added github.com/gorilla/mux v1.8.1
• PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> go get github.com/google/uuid
go: downloading github.com/google/uuid v1.6.0
go: added github.com/google/uuid v1.6.0
○ PS D:\RPL\GolangKuliah\pbw-praktikum-pertemuan10> go run main.go

```



5. Masukan title dengan body JSON dengan metode POST



6. Gunakan GET untuk melihat title yang sudah kita buat

