

LATIHAN PERTEMUAN 10
PRAKTIKUM PEMOGRAMAN BERBASIS WEB
Untuk Memenuhi Praktikum Pemograman Berbasis Web



Oleh:

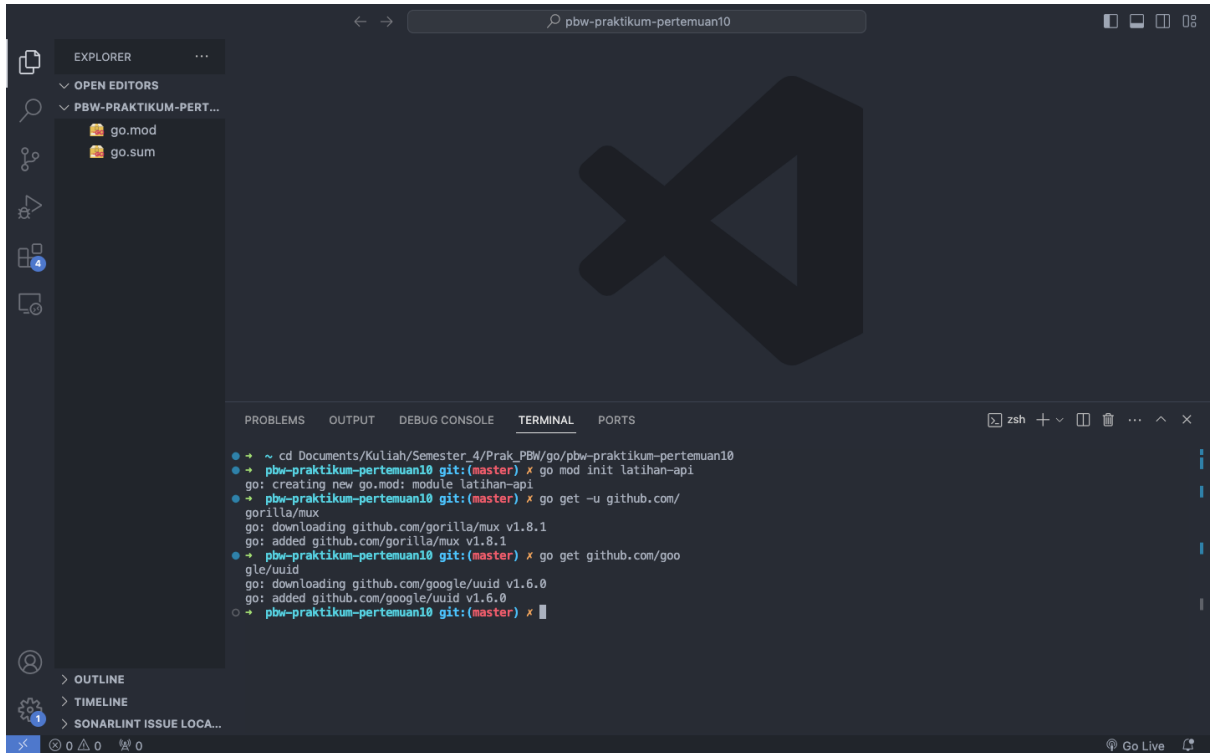
Nama : Siti Ghumaisa
NPM : 4522210131
Kelas : A
Semester : 4 (Genap)

Dosen :

ADI WAHYU PRIBADI ,S.SI.,M.KOM
S1-Teknik Informatika
Fakultas Teknik Universitas Pancasila
2023/2024

Link git-hub <https://github.com/Ghumaisa/PBW>

1. Initiate project --> install library gorilla dan



```
pbw-praktikum-pertemuan10

EXPLORER
  OPEN EDITORS
  PBW-PRAKTIKUM-PERT...
    go.mod
    go.sum

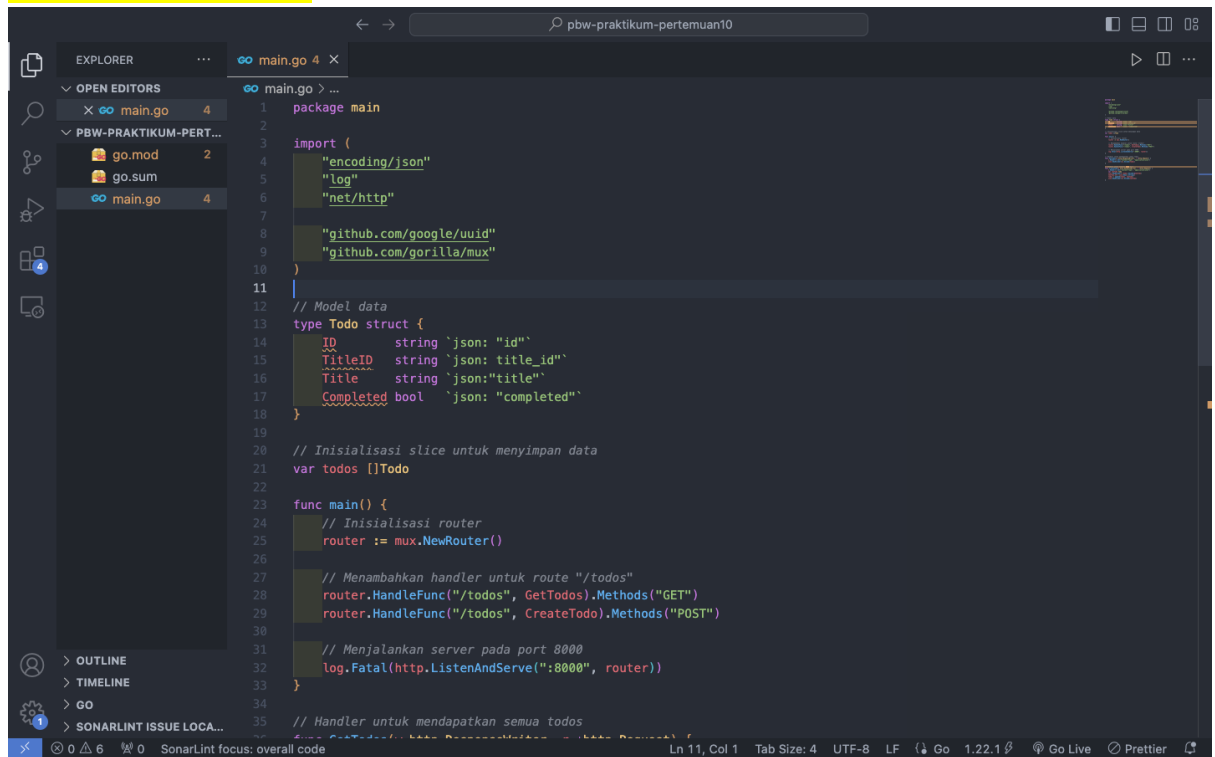
TERMINAL
  zsh
  ~ Documents/Kuliah/Semester_4/Prak_PBW/go/pbw-praktikum-pertemuan10
  pbw-praktikum-pertemuan10 git:(master) x go mod init latihan-api
  go: creating new go.mod: module latihan-api
  pbw-praktikum-pertemuan10 git:(master) x go get -u github.com/gorilla/mux
  go: downloading github.com/gorilla/mux v1.8.1
  go: added github.com/gorilla/mux v1.8.1
  pbw-praktikum-pertemuan10 git:(master) x go get github.com/google/uuid
  go: downloading github.com/google/uuid v1.6.0
  go: added github.com/google/uuid v1.6.0
  pbw-praktikum-pertemuan10 git:(master) x
```

2. Jelaskan library gorilla dan UUID

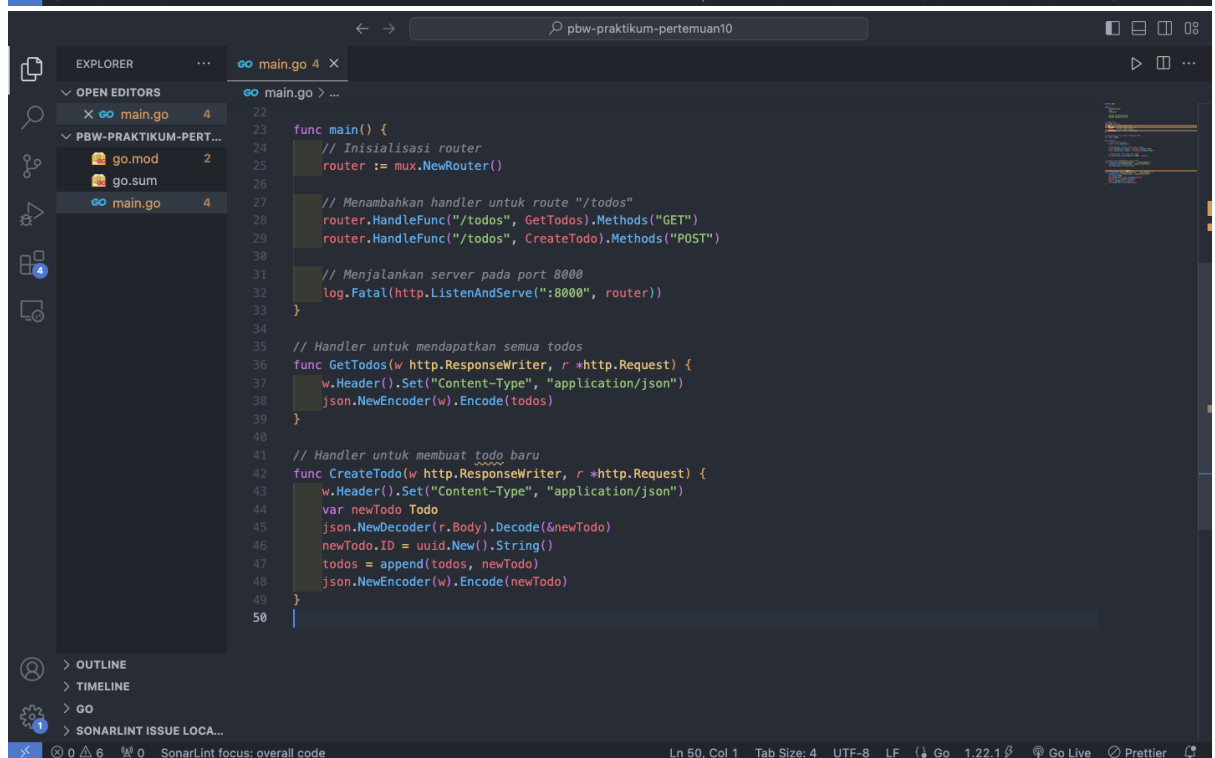
- **Gorilla:** pustaka Python yang menyediakan cara mudah untuk mengubah atau mengelola objek, khususnya metode dalam kelas, dengan melakukan monkey patching. Monkey patching adalah praktik memodifikasi atau memperluas kode program pada runtime tanpa mengubah sumber asli. Gorilla mempermudah proses ini dengan menyediakan antarmuka yang bersih dan sederhana, sehingga pengembang dapat menambah, mengganti, atau menghapus metode dari objek atau kelas tanpa harus menulis banyak kode boilerplate.
- **UUID (Universally Unique Identifier):** pustaka yang tersedia di banyak bahasa pemrograman, termasuk Python dan Java. UUID digunakan untuk menghasilkan pengenal unik yang dapat diandalkan untuk pengelolaan objek dalam sistem terdistribusi. UUID memiliki format standar yang terdiri dari 128 bit, yang biasanya direpresentasikan sebagai rangkaian 32 karakter heksadesimal. UUID sangat berguna dalam konteks di mana pengenal unik diperlukan, seperti dalam pembuatan kunci utama untuk database, penamaan sumber daya secara unik di jaringan, dan banyak aplikasi lain yang membutuhkan identifikasi yang unik dan tidak mudah diulang. Library Gorilla dan UUID digunakan dalam pengembangan perangkat lunak dengan bahasa pemrograman yang berbeda.

3. Buat file main.go

Screenshoot source code

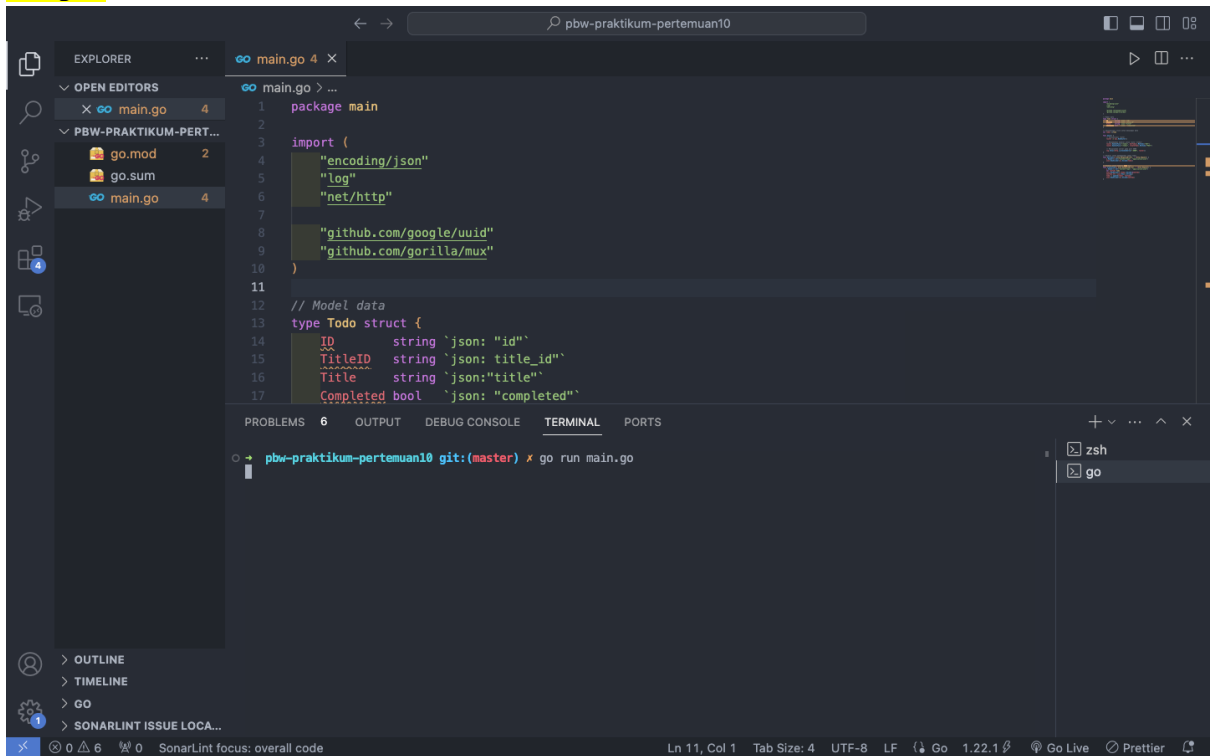


```
1 package main
2
3 import (
4     "encoding/json"
5     "log"
6     "net/http"
7
8     "github.com/google/uuid"
9     "github.com/gorilla/mux"
10 )
11
12 // Model data
13 type Todo struct {
14     ID string `json: "id"`
15     TitleID string `json: "title_id"`
16     Title string `json: "title"`
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     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
```



```
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     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 }
40
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 }
50
```

Output



The screenshot shows the Visual Studio Code editor with a Go file named `main.go` open. The code defines a `main` package with imports for `encoding/json`, `log`, `net/http`, `github.com/google/uuid`, and `github.com/gorilla/mux`. It defines a `Todo` struct with fields `ID`, `TitleID`, `Title`, and `Completed`. The terminal shows the command `go run main.go` being executed, and the output is `zsh` and `go`.

```
package main

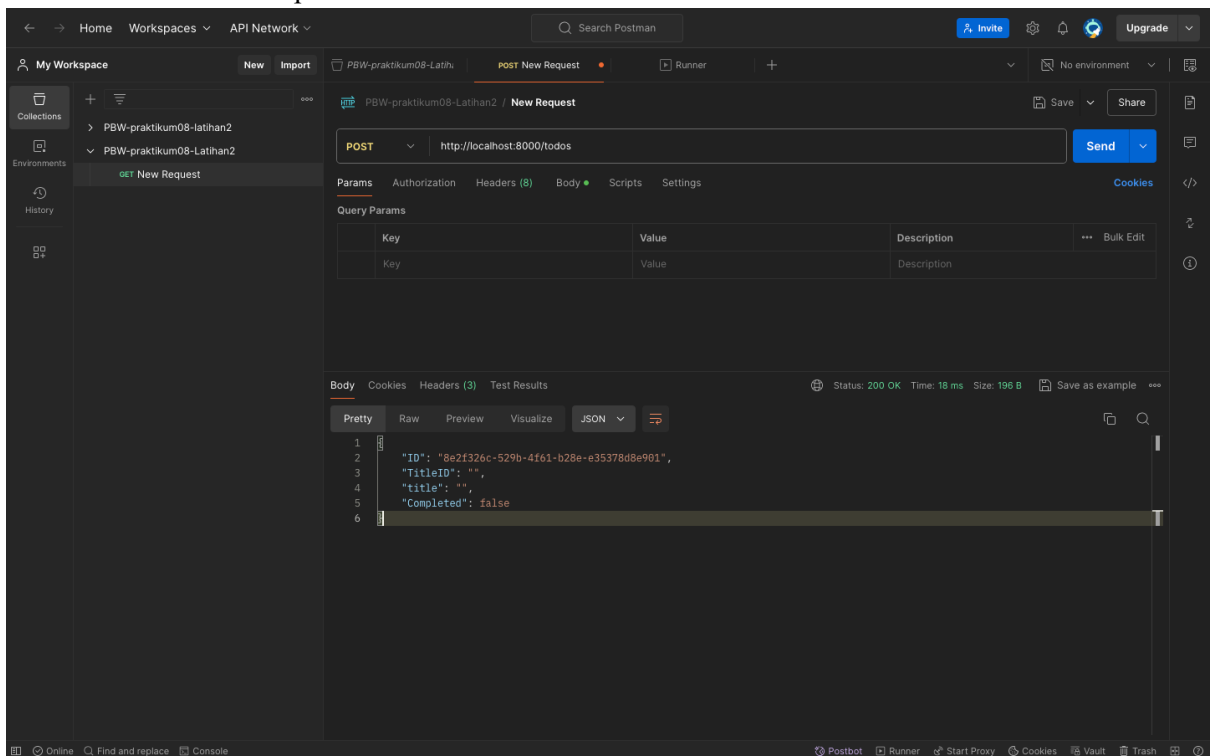
import (
    "encoding/json"
    "log"
    "net/http"
    "github.com/google/uuid"
    "github.com/gorilla/mux"
)

// Model data
type Todo struct {
    ID          string `json: "id"`
    TitleID     string `json: "title_id"`
    Title       string `json: "title"`
    Completed   bool  `json: "completed"`
}
```

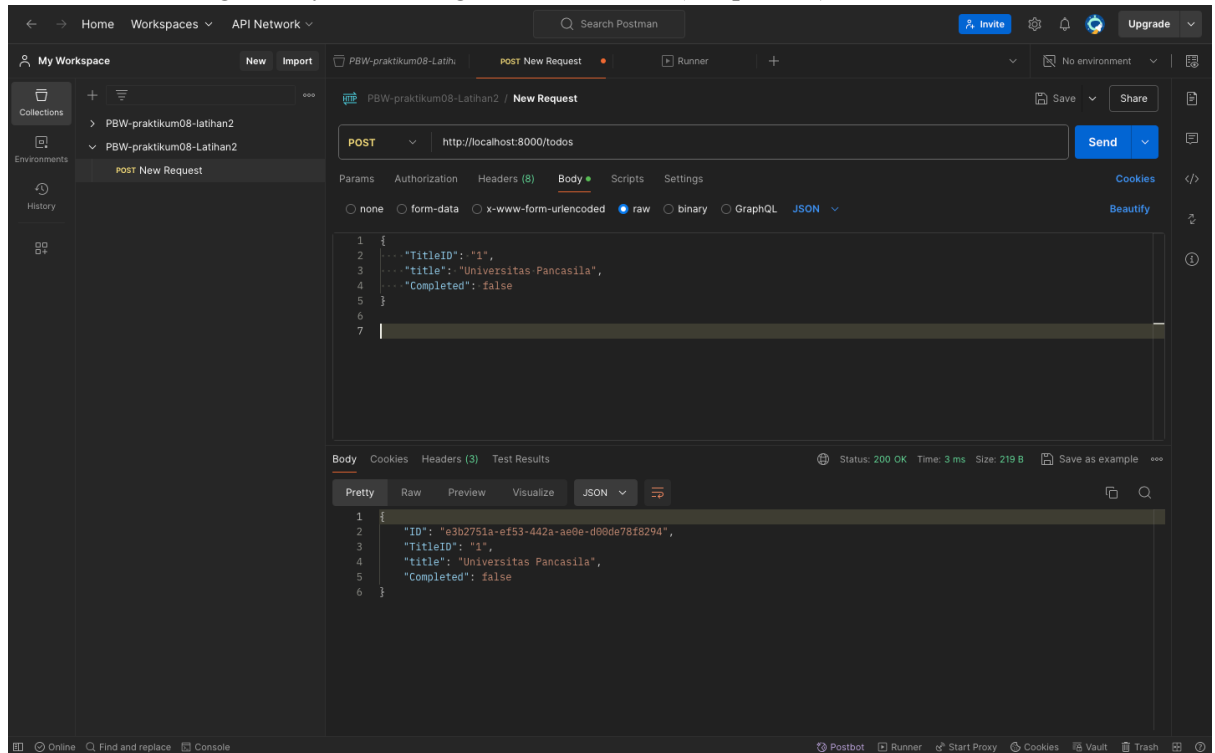
Terminal output:

```
pbw-praktikum-pertemuan10 git:(master) x go run main.go
zsh
go
```

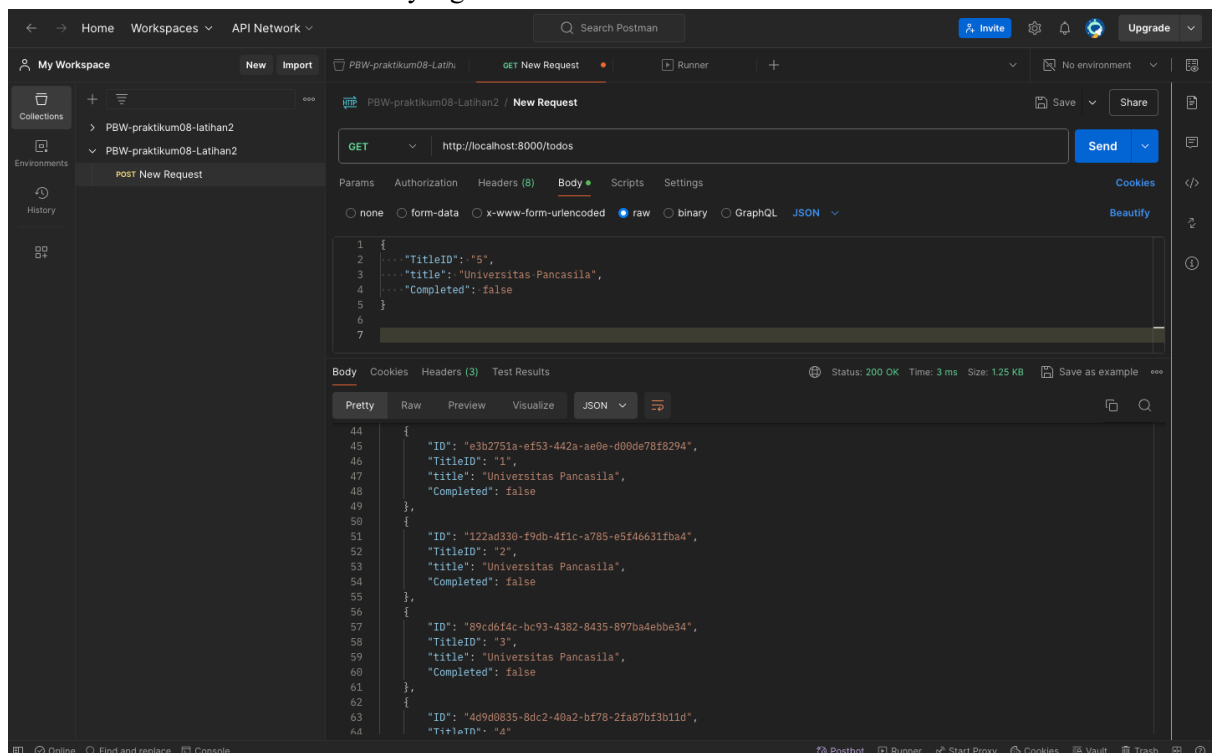
4. Jalankan local dan buka postman



5. Masukan title dengan body JSON dengan metode POST (sampe ke-5)



6. Gunakan GET untuk melihat title yang sudah kita buat



← → Home Workspaces ▾ API Network ▾

Search Postman

Invite

Upgrade ▾

My Workspace

New Import

PBW-praktikum08-Lathi.

GET New Request

Runner

+

No environment ▾

Collections

+

PBW-praktikum08-latihan2

▾ PBW-praktikum08-Latihan2

POST New Request

History

GET ▾

http://localhost:8000/todos

Send ▾

Params

Authorization

Headers (8)

Body •

Scripts

Settings

Cookies

Beautiful

☐ none

☐ form-data

☐ x-www-form-urlencoded

☒ raw

☐ binary

☐ GraphQL

JSON ▾

```
1 {
2   ...."TitleID": "5",
3   ...."title": "Universitas Pancasila",
4   ...."Completed": false
5 }
6
7
```

Body

Cookies

Headers (3)

Test Results

Status: 200 OK Time: 3 ms Size: 1.25 KB Save as example

Pretty

Raw

Preview

Visualize

JSON ▾

```
54 {
55   "Completed": false
56 },
57 {
58   "ID": "89cd614c-bc93-4382-8435-897ba4ebbe34",
59   "TitleID": "3",
60   "title": "Universitas Pancasila",
61   "Completed": false
62 },
63 {
64   "ID": "4d9d8835-8dc2-48a2-bf78-2fa87bf3b11d",
65   "TitleID": "4",
66   "title": "Universitas Pancasila",
67   "Completed": false
68 },
69 {
70   "ID": "99d9dffa-fbe5-4d4d-a44f-9734be467155",
71   "TitleID": "5",
72   "title": "Universitas Pancasila",
73   "Completed": false
74 }
```

Online

Find and replace

Console

Postbot

Runner

Start Proxy

Cookies

Vault

Trash