

LAPORAN PRAKTIKUM
PEMROGRAMAN BERBASIS WEB
TUGAS 03

Untuk memenuhi Tugas 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/tree/master>

Package

1. Membuat folder untuk dijadikan project, misal progo

\$ mkdir progo

```
MINGW64:/d/RPL/golang_kuliah/progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ mkdir progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
progo/
```

2. Pindah ke folder progo

\$ cd progo

```
MINGW64:/d/RPL/golang_kuliah/progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ mkdir progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
progo/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ cd progo/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
```

3. Menjadikan folder progo sebagai module

\$ go mod init progo

```
MINGW64:/d/RPL/golang_kuliah/progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ mkdir progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
progo/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ cd progo/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ go mod init progo
go: creating new go.mod: module progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod
```

4. Membuat folder ratarata

\$ mkdir ratarata

```
MINGW64:/d/RPL/golang_kuliah/progo

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ mkdir progo

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
progo/

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ cd progo/

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ go mod init progo
go: creating new go.mod: module progo

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ mkdir ratarata

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod  ratarata/
```

5. Pindah ke folder ratarata

\$ cd ratarata

```
MINGW64:/d/RPL/golang_kuliah/progo

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ mkdir progo

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
progo/

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ cd progo/

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ go mod init progo
go: creating new go.mod: module progo

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod

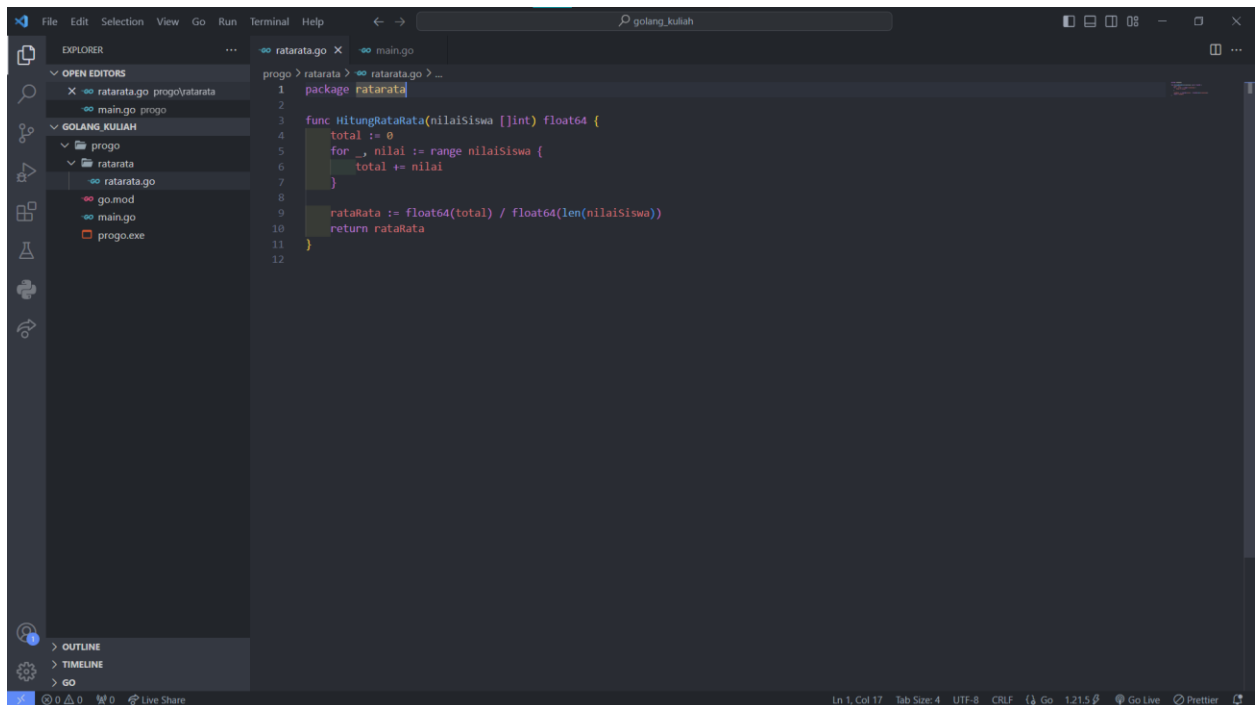
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ mkdir ratarata

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod  ratarata/

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ cd ratarata/

muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo/ratarata
$ ls
```

6. Membuat file bernama ratarata.go

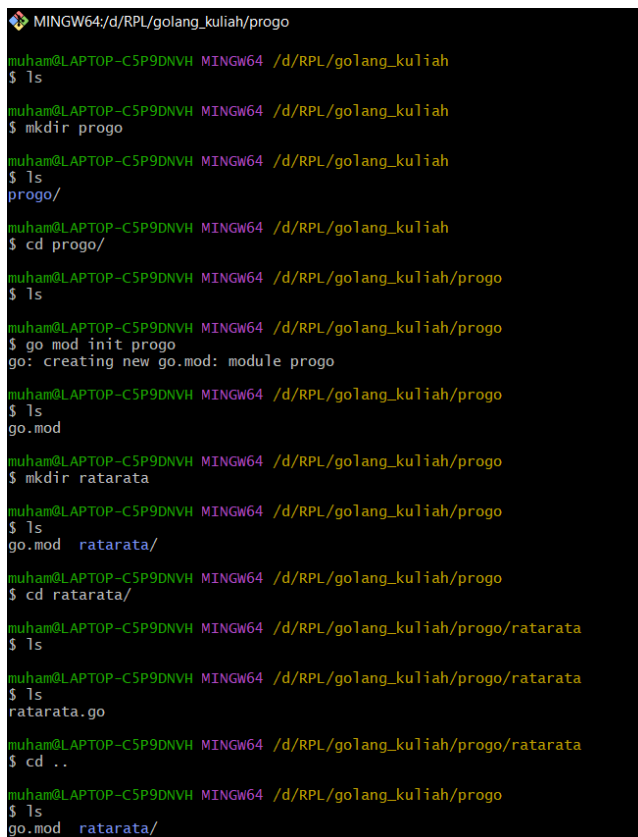


The screenshot shows the Visual Studio Code editor interface. The Explorer sidebar on the left shows a project structure with folders 'progo' and 'ratarata'. Inside 'ratarata', there is a file 'ratarata.go'. The main editor window displays the code for 'ratarata.go'.

```
1 package ratarata
2
3 func HitungRataRata(nilaiSiswa []int) float64 {
4     total := 0
5     for _, nilai := range nilaiSiswa {
6         total += nilai
7     }
8
9     rataRata := float64(total) / float64(len(nilaiSiswa))
10    return rataRata
11 }
12
```

7. Pindah ke folder progo

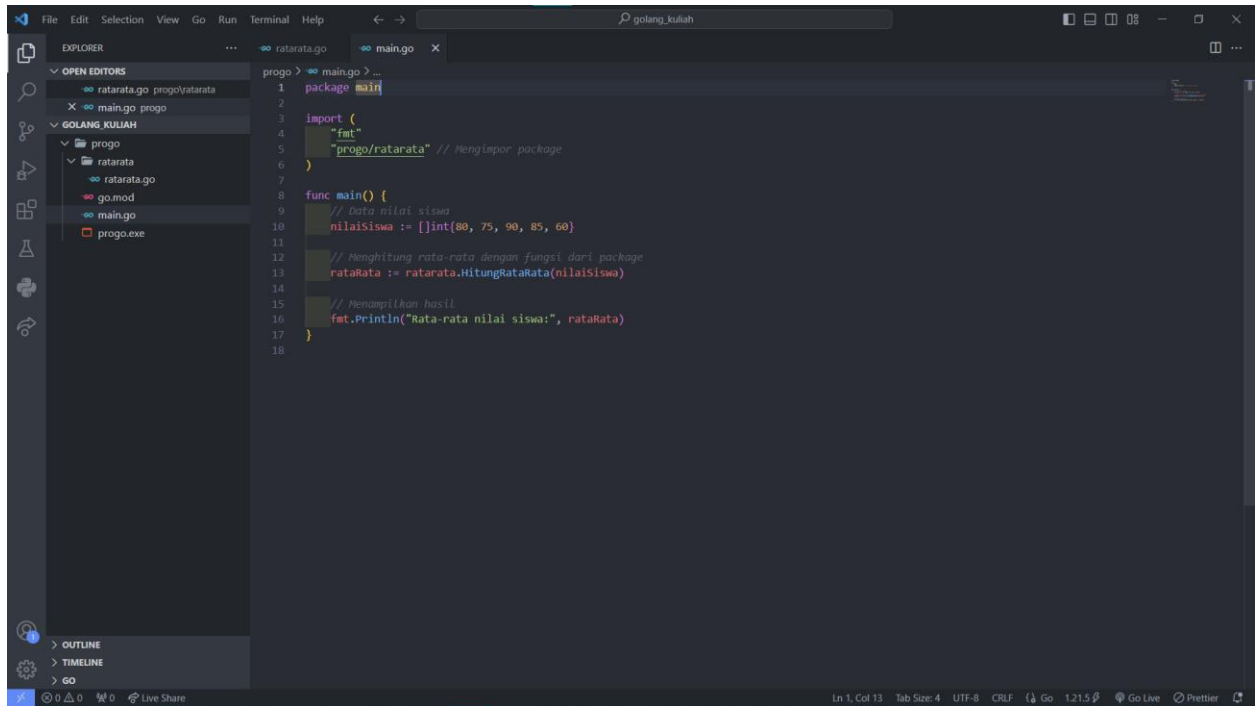
\$ cd ..



The screenshot shows a terminal window with the following commands and output:

```
MINGW64/d/RPL/golang_kuliah/progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ mkdir progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ ls
progo/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah
$ cd progo/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ go mod init progo
go: creating new go.mod: module progo
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ mkdir ratarata
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod  ratarata/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ cd ratarata/
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo/ratarata
$ ls
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo/ratarata
$ ls
ratarata.go
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo/ratarata
$ cd ..
muham@LAPTOP-C5P9DNVH MINGW64 /d/RPL/golang_kuliah/progo
$ ls
go.mod  ratarata/
```

8. Membuat file main.go



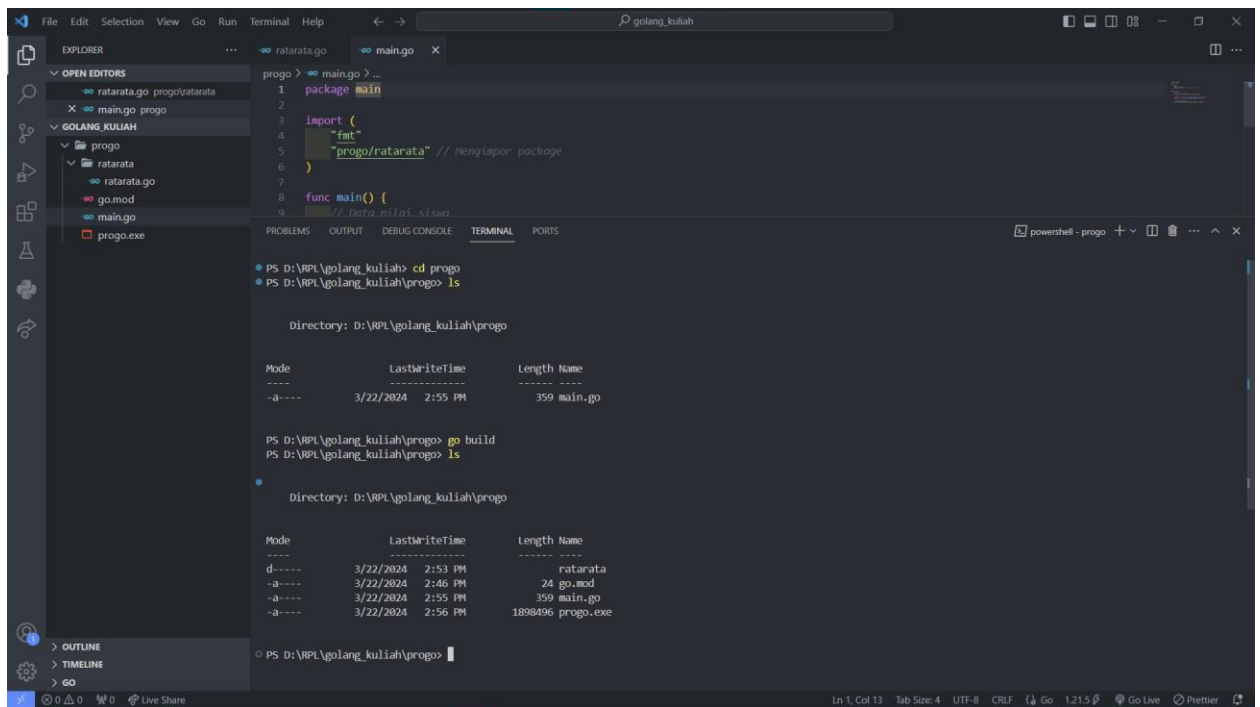
The screenshot shows the Visual Studio Code editor with a project named 'golang_kuliah'. The Explorer sidebar on the left shows the file structure: 'ratarata.go' and 'main.go' are open in the editor. The 'main.go' file is being edited, showing the following code:

```
1 package main
2
3 import (
4     "fmt"
5     "progo/ratarata" // Mengimpor package
6 )
7
8 func main() {
9     // Data nilai siswa
10    nilaiSiswa := []int{80, 75, 90, 85, 60}
11
12    // Menghitung rata-rata dengan fungsi dari package
13    rataRata := ratarata.HitungRataRata(nilaiSiswa)
14
15    // Menampilkan hasil
16    fmt.Println("Rata-rata nilai siswa:", rataRata)
17 }
18
```

The status bar at the bottom indicates the file is at line 1, column 13, with a tab size of 4, UTF-8 encoding, and CRLF line endings. It also shows the Go version 1.21.5 and the Prettier formatter.

9. Build main.go dan muncul file progo.exe

\$ go build



The screenshot shows the Visual Studio Code editor with the same project. The 'main.go' file is still open. The Terminal panel at the bottom shows the following commands and output:

```
PS D:\RPL\golang_kuliah> cd progo
PS D:\RPL\golang_kuliah\progo> ls

Directory: D:\RPL\golang_kuliah\progo

Mode                LastWriteTime         Length Name
----                -
-a----             3/22/2024   2:55 PM             359 main.go

PS D:\RPL\golang_kuliah\progo> go build
PS D:\RPL\golang_kuliah\progo> ls

Directory: D:\RPL\golang_kuliah\progo

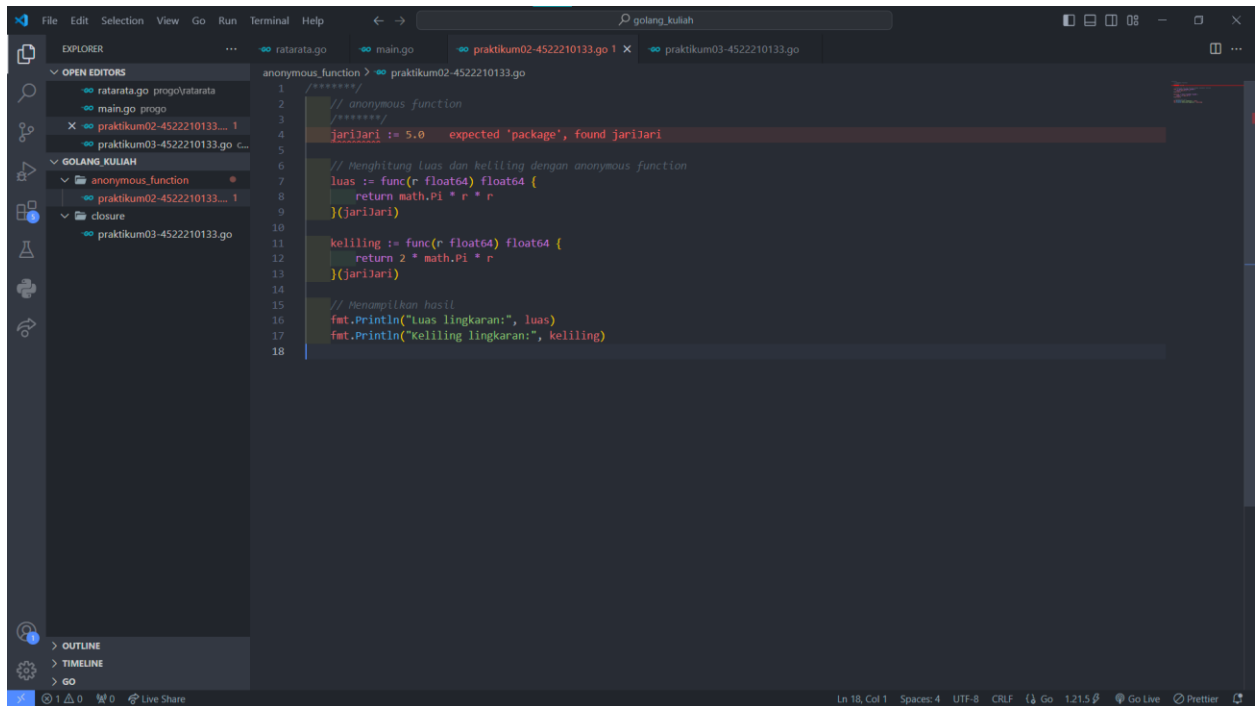
Mode                LastWriteTime         Length Name
----                -
d-----             3/22/2024   2:53 PM             ratarata
-a----             3/22/2024   2:46 PM              24 go.mod
-a----             3/22/2024   2:55 PM             359 main.go
-a----             3/22/2024   2:56 PM          1898496 progo.exe

PS D:\RPL\golang_kuliah\progo>
```

The status bar at the bottom indicates the file is at line 1, column 13, with a tab size of 4, UTF-8 encoding, and CRLF line endings. It also shows the Go version 1.21.5 and the Prettier formatter.

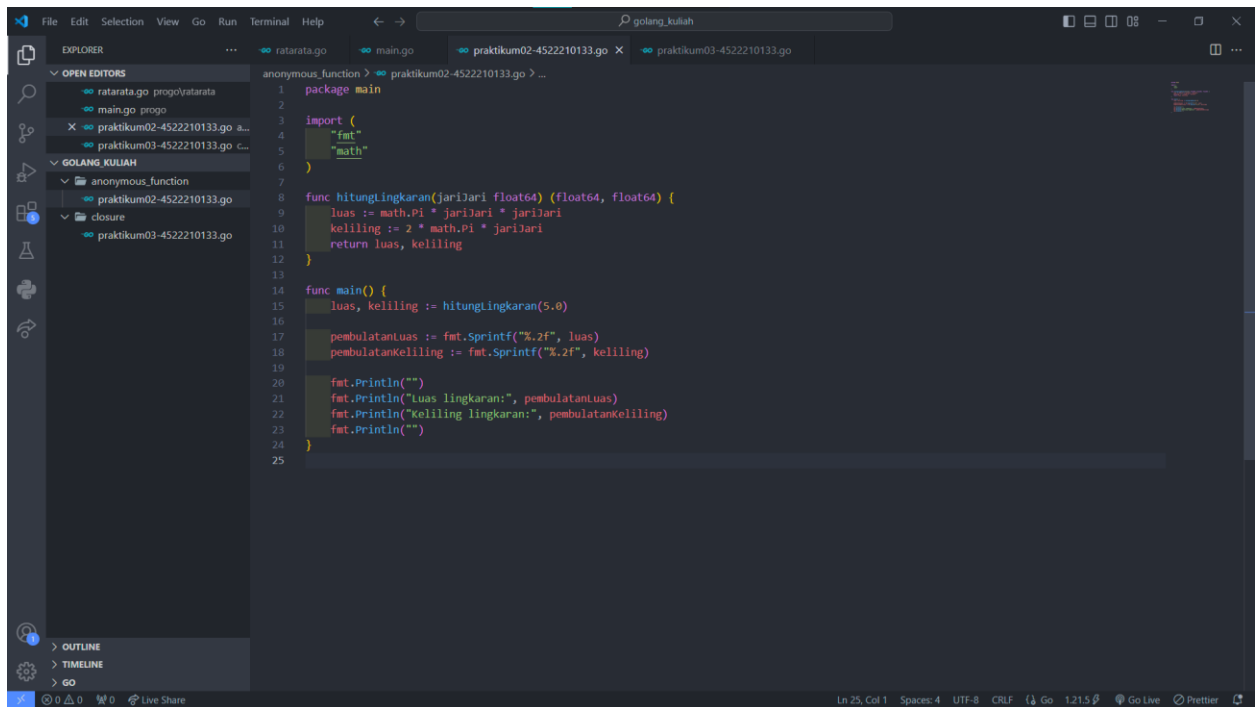
Anonymous Function

- Program awal



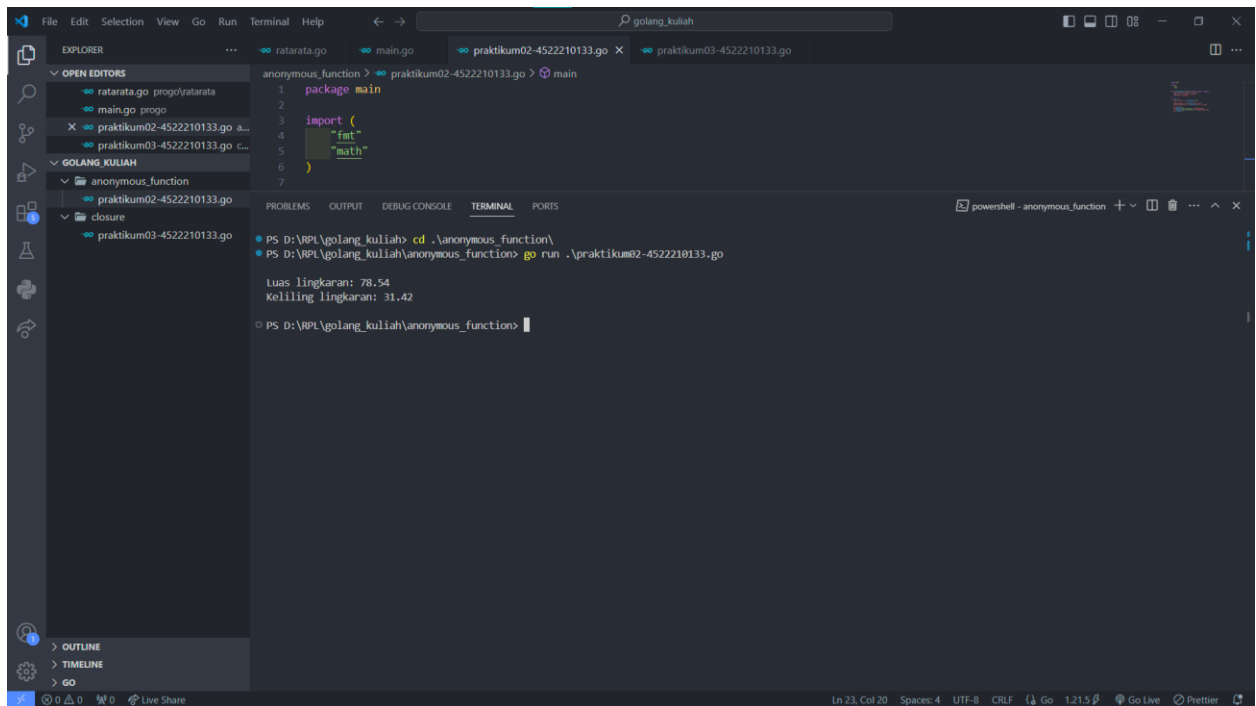
```
1 // *****
2 // anonymous function
3 // *****
4 jariJari := 5.0 // expected 'package', found jariJari
5
6 // Menghitung luas dan keliling dengan anonymous function
7 luas := func(r float64) float64 {
8     return math.Pi * r * r
9 }(jariJari)
10
11 keliling := func(r float64) float64 {
12     return 2 * math.Pi * r
13 }(jariJari)
14
15 // Menampilkan hasil
16 fmt.Println("luas lingkaran:", luas)
17 fmt.Println("keliling lingkaran:", keliling)
18
```

- Program setelah dilengkapi agar dapat dieksekusi



```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func hitungLingkaran(jariJari float64) (float64, float64) {
9     luas := math.Pi * jariJari * jariJari
10    keliling := 2 * math.Pi * jariJari
11    return luas, keliling
12}
13
14 func main() {
15    luas, keliling := hitungLingkaran(5.0)
16
17    pembulatanLuas := fmt.Sprintf("%.2f", luas)
18    pembulatanKeliling := fmt.Sprintf("%.2f", keliling)
19
20    fmt.Println("")
21    fmt.Println("luas lingkaran:", pembulatanLuas)
22    fmt.Println("keliling lingkaran:", pembulatanKeliling)
23    fmt.Println("")
24}
25
```

- Output



```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func hitungLingkaran(jari float64) (float64, float64) {
9     luas := math.Pi * jari * jari
10    keliling := 2 * math.Pi * jari
11    return luas, keliling
12 }
13
14 func main() {
15     jari := 5.0
16     luas, keliling := hitungLingkaran(jari)
17     fmt.Println("Luas lingkaran: ", luas)
18     fmt.Println("Keliling lingkaran: ", keliling)
19 }
```

PS D:\VPL\golang_kuliah> cd .\anonymous_function\
PS D:\VPL\golang_kuliah\anonymous_function> go run .\praktikum02-4522210133.go

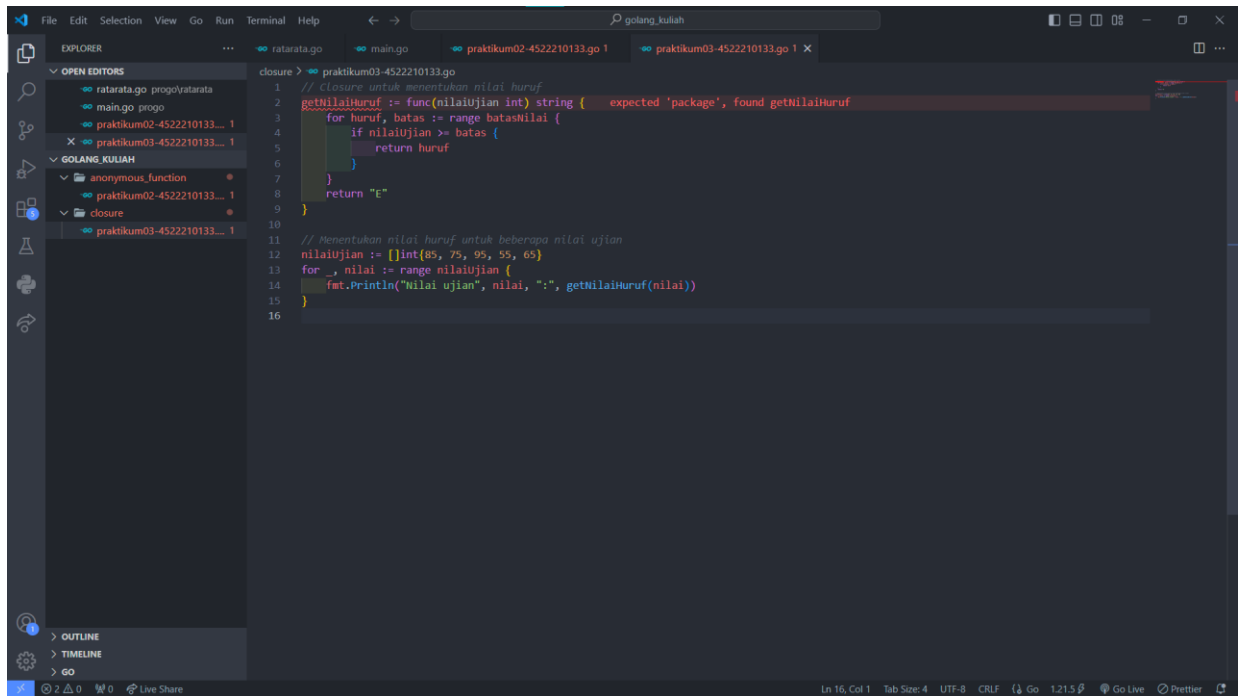
Luas lingkaran: 78.54
Keliling lingkaran: 31.42

- Penjelasan agar program dapat dieksekusi

1. Pertama-tama, saya mengubah struktur program dengan menambahkan package main, mengimport fmt dan math, serta menambahkan function main agar program dapat dijalankan.
2. Lalu, saya mendefinisikan function hitungLingkaran(jari float64) (float64, float64) yang menerima parameter jari-jari lingkaran dan me-return/mengembalikan dua nilai, yaitu luas dan keliling lingkaran.
3. Kemudian, di dalam function hitungLingkaran, saya menghitung nilai luas dan keliling lingkaran menggunakan rumus yang sesuai.
4. Selanjutnya, saya menggunakan fmt.Sprintf untuk membulatkan nilai luas dan keliling lingkaran menjadi dua angka di belakang koma.
5. Setelah itu, saya memanggil function hitungLingkaran dengan nilai jari-jari 5.0 dan menyimpan hasilnya dalam variabel luas dan keliling.
6. Terakhir, saya mencetak hasil perhitungan luas dan keliling lingkaran yang sudah dibulatkan menggunakan fmt.Println.

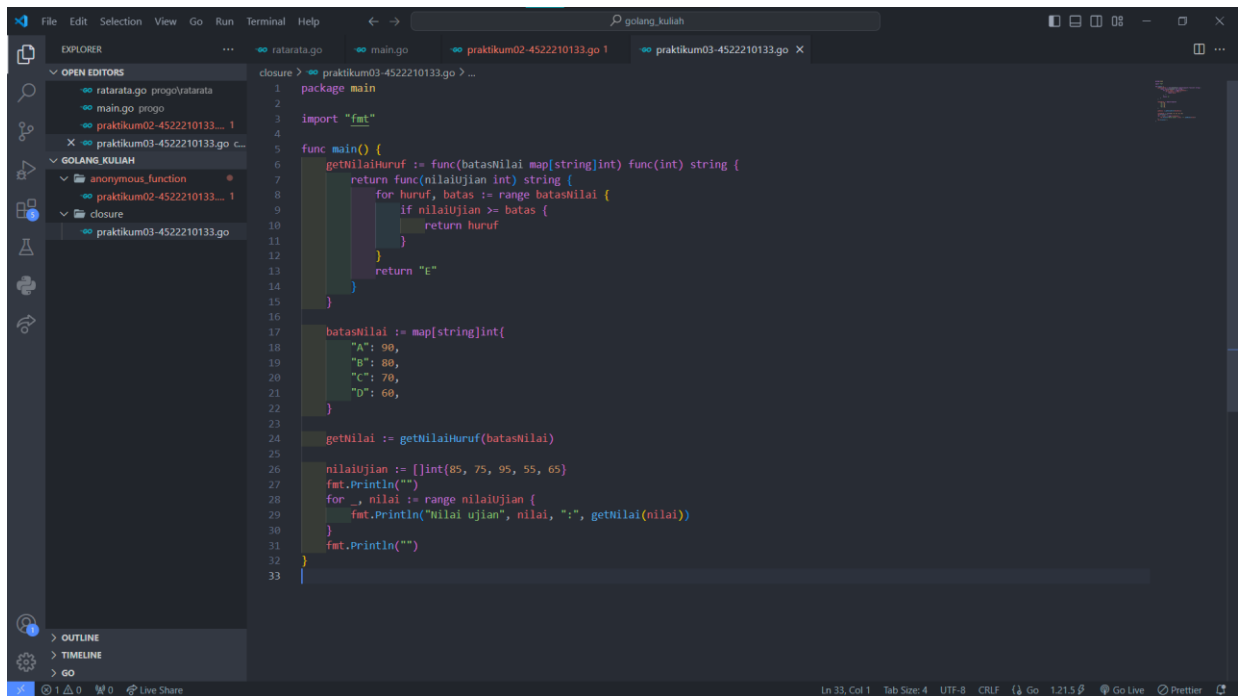
Closure

- Program awal



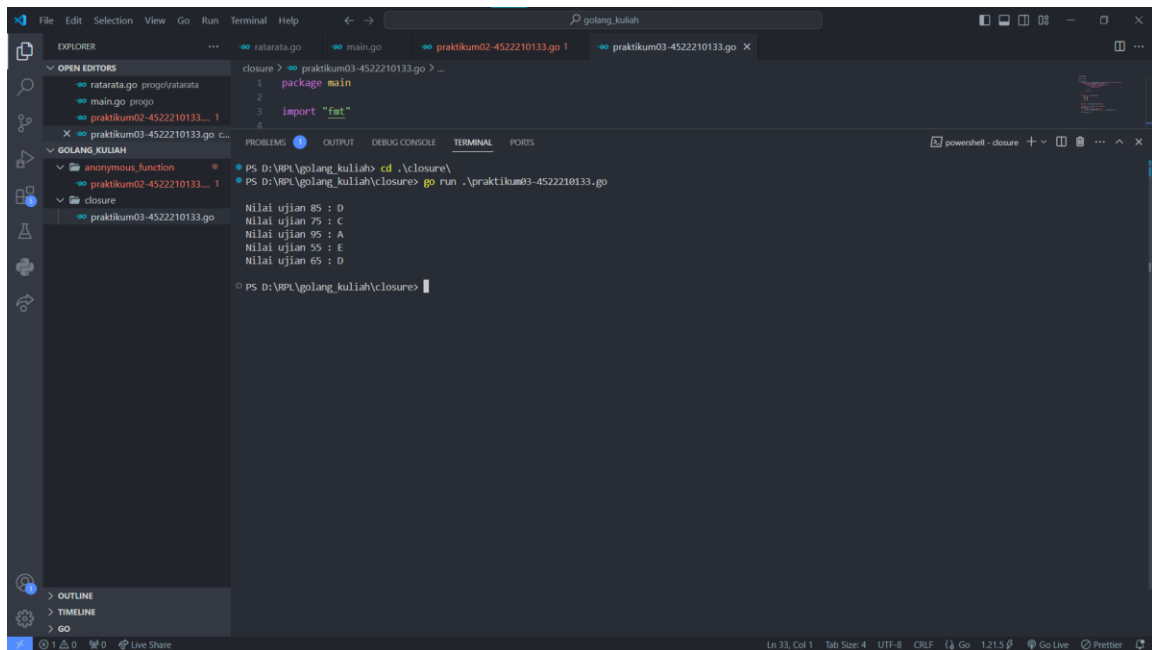
```
1 // Closure untuk menentukan nilai huruf
2 getNilaiHuruf := func(nilaiUjian int) string {
3     for huruf, batas := range batasNilai {
4         if nilaiUjian >= batas {
5             return huruf
6         }
7     }
8     return "E"
9 }
10
11 // Menentukan nilai huruf untuk beberapa nilai ujian
12 nilaiUjian := []int{85, 75, 95, 55, 65}
13 for _, nilai := range nilaiUjian {
14     fmt.Println("Nilai ujian", nilai, ":", getNilaiHuruf(nilai))
15 }
16
```

- Program setelah dilengkapi agar dapat dieksekusi



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     getNilaiHuruf := func(batasNilai map[string]int) func(int) string {
7         return func(nilaiUjian int) string {
8             for huruf, batas := range batasNilai {
9                 if nilaiUjian >= batas {
10                     return huruf
11                 }
12             }
13             return "E"
14         }
15     }
16
17     batasNilai := map[string]int{
18         "A": 90,
19         "B": 80,
20         "C": 70,
21         "D": 60,
22     }
23
24     getNilai := getNilaiHuruf(batasNilai)
25
26     nilaiUjian := []int{85, 75, 95, 55, 65}
27     fmt.Println("")
28     for _, nilai := range nilaiUjian {
29         fmt.Println("Nilai ujian", nilai, ":", getNilai(nilai))
30     }
31     fmt.Println("")
32 }
33
```


- Output



The screenshot shows a Go IDE with a terminal window displaying the output of a program. The terminal output is as follows:

```
PS D:\VRPL\golang_kuliah> cd .\closure\  
PS D:\VRPL\golang_kuliah\closure> go run .\praktikum03-4522210133.go  
Nilai ujian 85 : D  
Nilai ujian 75 : C  
Nilai ujian 95 : A  
Nilai ujian 55 : E  
Nilai ujian 65 : D  
PS D:\VRPL\golang_kuliah\closure>
```

- Penjelasan agar program dapat dieksekusi

1. Pertama-tama, saya menambahkan package main, meng-import fmt dan menambahkan function main agar program dapat dieksekusi.
2. Lalu, saya mengubah function getNilaiHuruf dengan menambahkan parameter batasNilai map[string]int agar dapat menerima parameter berupa map yang berisi batas nilai untuk setiap huruf.
3. Kemudian, saya menambahkan function di dalam function getNilaiHuruf yang me-return/mengembalikan function baru dengan tipe data func(int) string. Function ini digunakan untuk memperoleh nilai huruf berdasarkan nilai ujian.
4. Karena tadi saya menambahkan parameter batasNilai map[string]int, berarti saya juga membuat map tersebut untuk menentukan nilai huruf berdasarkan nilai ujian.
5. Saya menginisialisasi variabel getNilai dengan nilai pemanggilan function getNilaiHuruf(batasNilai) untuk mendapatkan function yang sesuai dengan batas nilai yang telah ditentukan.
6. Selanjutnya, saya melakukan iterasi nilaiUjian dan memanggil function getNilai untuk mendapatkan nilai huruf berdasarkan nilai ujian tersebut.
7. Terakhir, saya mencetak hasil perolehan nilai huruf untuk setiap nilai ujian dengan menggunakan fmt.Println.