

# **TUGAS PEMROGRAMAN BERBASIS WEB**

## **KELAS A**

Disusun Untuk Memenuhi Tugas Mata Kuliah Pemrograman Berbasis Web



Disusun oleh:

Gibran Kahlil      4522210128

Dosen Pengampu:

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

**Program Studi Teknik Informatika**

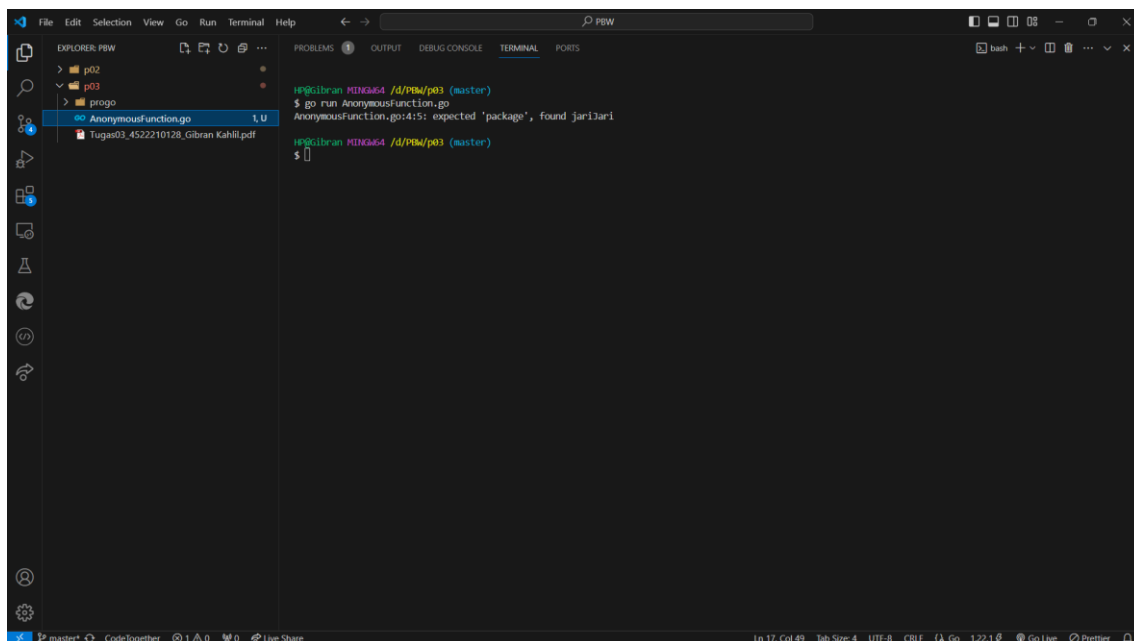
**Fakultas Teknik Universitas Pancasila**

**2024**

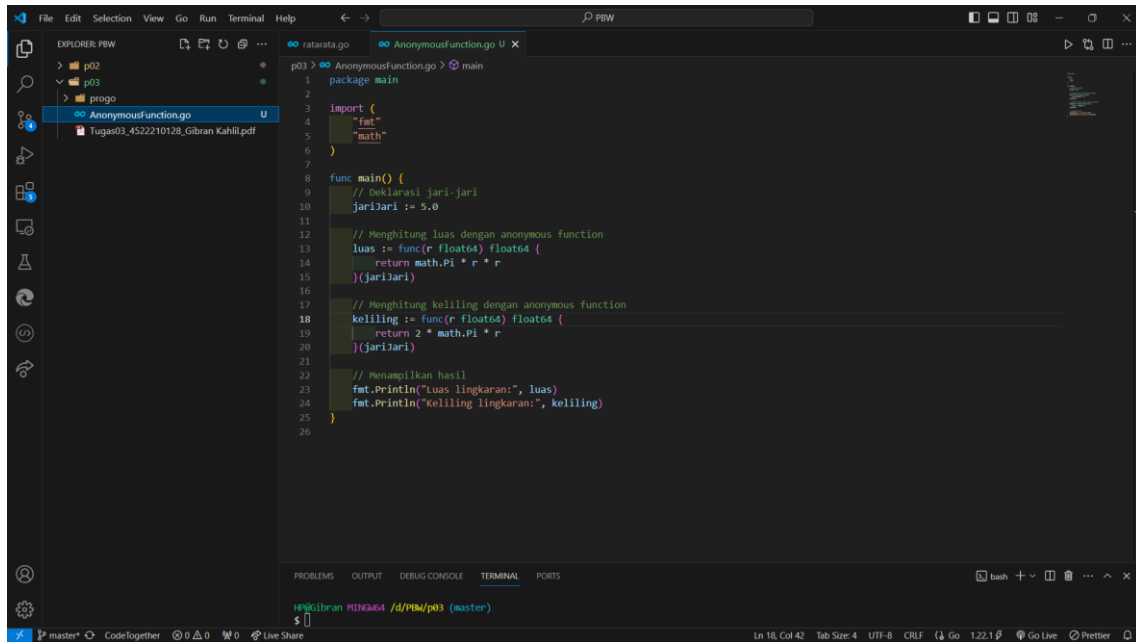
## Anonymous Function

```
p03 > AnonymousFunction.go
1  /******/
2  // anonymous function
3  /******/
4  jariJari := 5.0
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)
```

Pada source code di atas terdapat beberapa kesalahan, yaitu pada saat memanggil Anonymous Function dengan melewati argumen “jariJari” sebagai parameter, tetapi cara ini tidak benar maka tidak bisa di running. Berikut adalah hasil running pada source code yang belum di perbaiki:

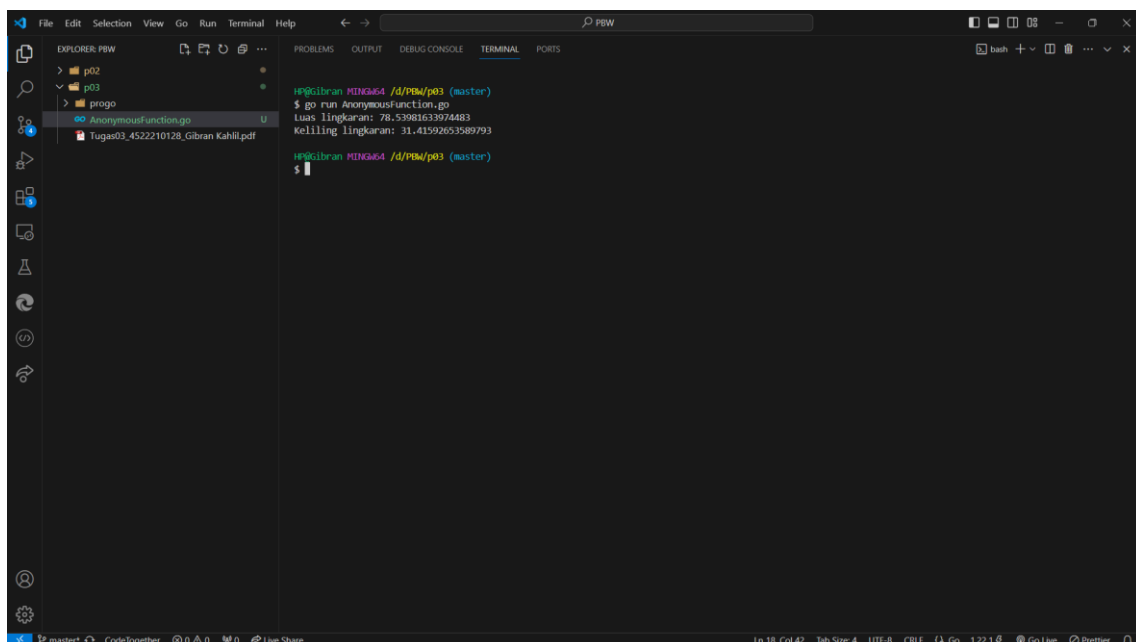


```
File Edit Selection View Go Run Terminal Help
Explorer: PBW
  > p03
  > p03
  > prog
  > AnonymousFunction.go
  > Tugas03_4522210120_Gibran Kahili.pdf
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
H@Gibran MINK664 /d:/PBW/p03 (master)
$ go run AnonymousFunction.go
AnonymousFunction.go:15:5: expected 'package', found jariJari
H@Gibran MINK664 /d:/PBW/p03 (master)
$
```



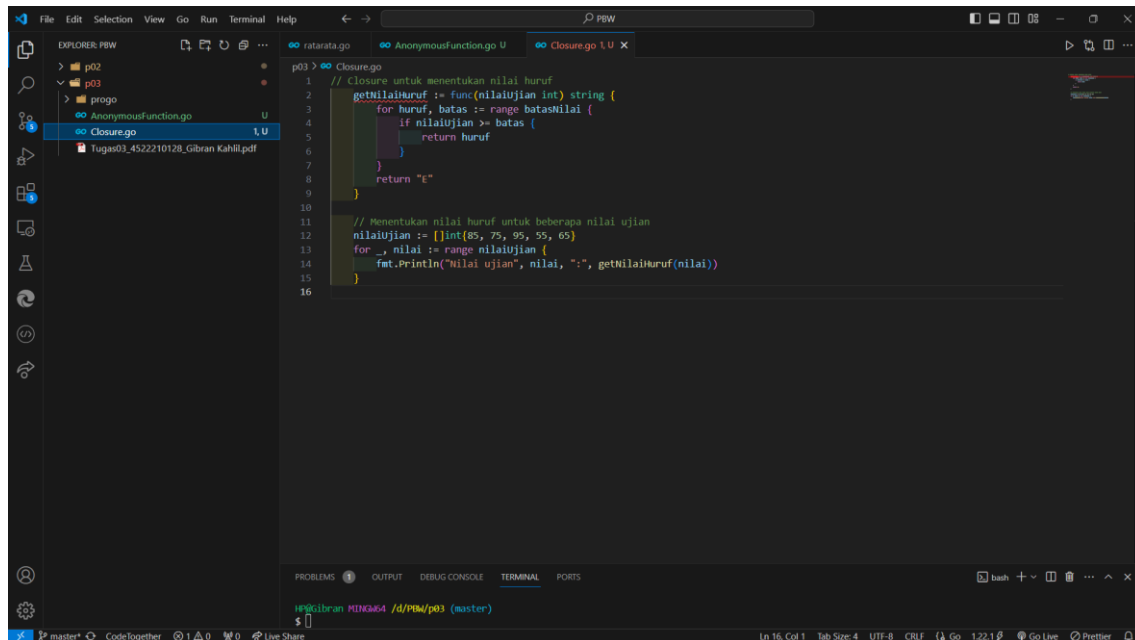
```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     // Deklarasi jari-jari
10    jariJari := 5.0
11
12    // Menghitung luas dengan anonymous function
13    luas := func(r float64) float64 {
14        return math.Pi * r * r
15    }(jariJari)
16
17    // Menghitung keliling dengan anonymous function
18    keliling := func(r float64) float64 {
19        return 2 * math.Pi * r
20    }(jariJari)
21
22    // Menampilkan hasil
23    fmt.Println("Luas lingkaran:", luas)
24    fmt.Println("Keliling lingkaran:", keliling)
25 }
26
```

Berikut adalah perbaikan dari source code sebelumnya, dimana fungsi anonim dipanggil terpisah dari deklarasinya dengan melewatkan nilai “jariJari” sebagai argument. Berikut adalah hasil running dari program yang sudah di perbaiki:



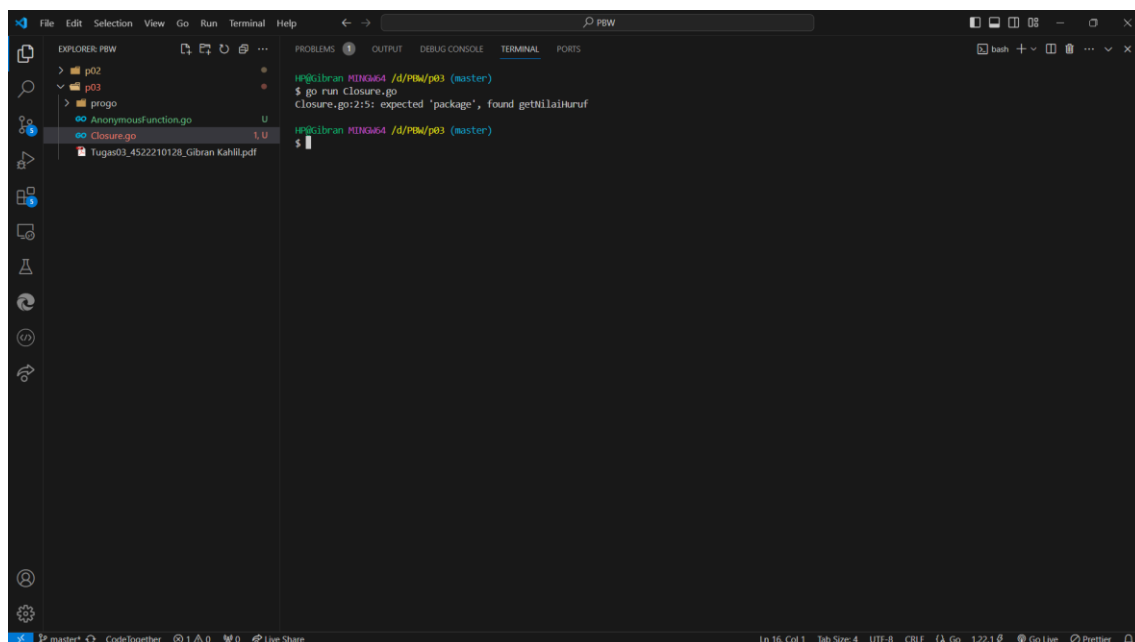
```
HP@Gibran MINGW64 /d/Program/p03 (master)
$ go run Anonymousfunction.go
Luas lingkaran: 78.53981633974483
Keliling lingkaran: 31.41592653589793
HP@Gibran MINGW64 /d/Program/p03 (master)
$
```

## Closure

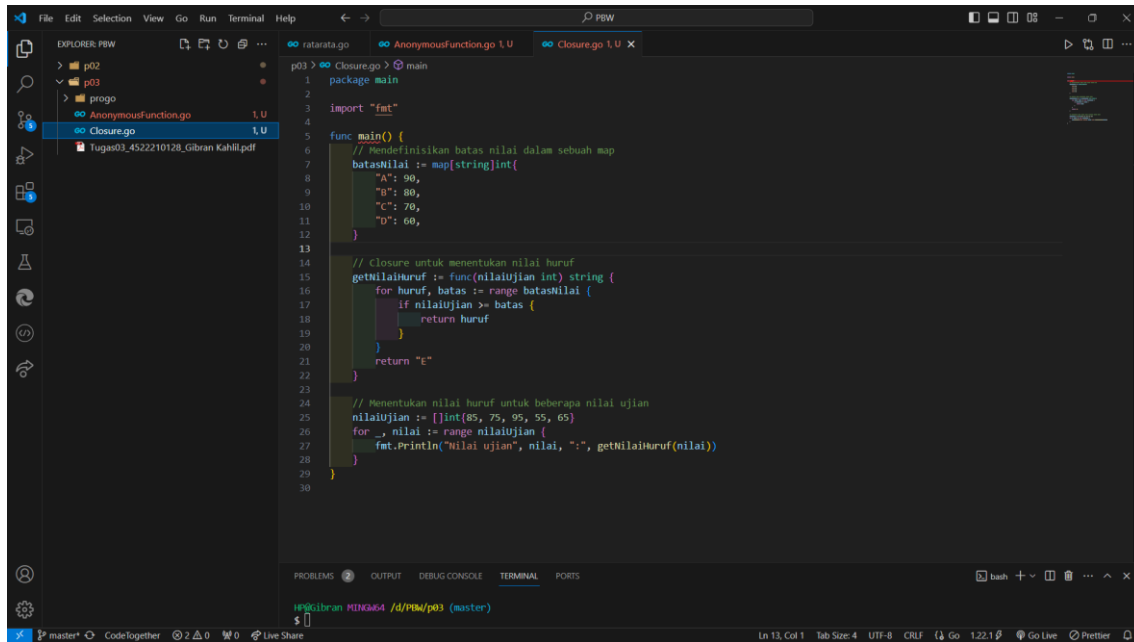


```
1 // Closure untuk menentukan nilai huruf
2 getNilaiHuruf := func(nilaiujian int) string {
3     for huruf, batas := range batasNilai {
4         if nilaiujian >= batas {
5             return huruf
6         }
7     }
8     return "E"
9 }
10
11 // Menentukan nilai huruf untuk beberapa nilai ujian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

Pada source code di atas terdapat sebuah variabel “batasNilai” yang digunakan dalam fungsi “getNilaiHuruf”, tetapi variabel ini tidak didefinisikan di dalam fungsi ataupun di luar fungsi “main”. Karena “batasNilai” adalah sebuah variabel global yang didefinisikan di tempat lain maka program tidak akan jalan. Karena fungsi “batasNilai” tidak memiliki akses variabel terhadap “getNilaiHuruf”. Berikut adalah hasil running pada source code yang belum di perbaiki:

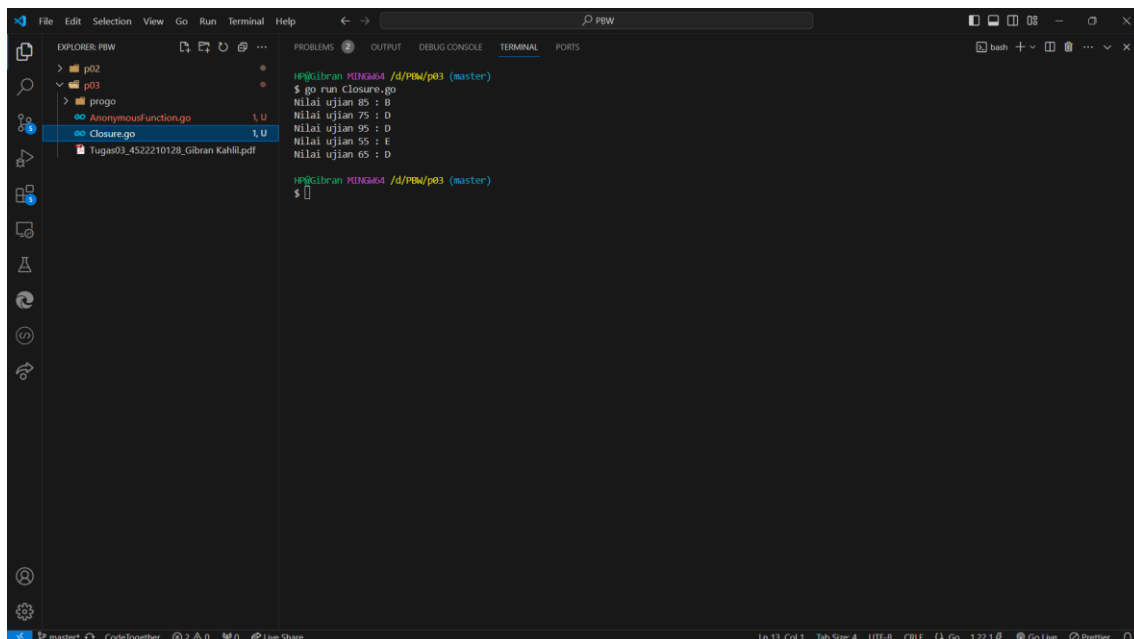


```
HP@Gibran MINGW64 /d/PPM/p03 (master)
$ go run Closure.go
Closure.go:2:5: expected 'package', found getNilaiHuruf
HP@Gibran MINGW64 /d/PPM/p03 (master)
$
```



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

Berikut ini adalah source code yang sudah diperbaiki dengan menambahkan deklarasi variabel “batasNilai” dengan mengubah fungsi “getNilaiHuruf” menjadi Closure. Mendefinisikan “batasNilai” sebagai sebuah map yang menyimpan batas nilai untuk setiap huruf, lalu digunakan dalam fungsi “getNilaiHuruf” sebagai closure. Berikut ini adalah hasil running dari source code yang sudah diperbaiki:



```
HP@Gibran MINGW64 /d/PPM/p03 (master)
$ go run Closure.go
Nilai ujian 85 : B
Nilai ujian 75 : D
Nilai ujian 95 : D
Nilai ujian 55 : E
Nilai ujian 65 : D
HP@Gibran MINGW64 /d/PPM/p03 (master)
$
```