


Nama:	NIM:	Nilai:
-------	------	--------

	<b>Tugas 1</b> <b>Pemrograman Jaringan (CSH4V3)</b> ----- <b>Semester Ganjil 2019 - 2020</b> <b>Dosen: Aulia Arif Wardana, S.Kom., M.T. (UIW)</b>	
	<p><i><b>Berdo'alah sebelum mengerjakan. Dilarang berbuat curang.</b></i></p> <p><i><b>Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.</b></i></p> <p><i><b>Selamat belajar, semoga sukses !</b></i></p>	
<b>Nama Mahasiswa:</b> <b>Reno Butar Butar</b> .....	<b>NIM:</b> <b>1301164724</b> .....	<b>Nilai:</b> .....

**Siapkan tools berikut sebelum mengerjakan:**

1. Go Programming Language (<https://golang.org/dl/>).
2. Visual Studio Code (<https://code.visualstudio.com/>) atau LiteIDE (<https://github.com/visualfc/liteide>).
3. Harus menggunakan linux dengan distro fedora (<https://getfedora.org/id/workstation/>).
4. Buatlah git repository pada <https://github.com/> kemudian push semua kode dan hasil laporan anda ke dalam repository github yang sudah anda buat.
5. Kumpulkan link repository github tersebut sebagai tanda bahwa anda mengerjakan tugas modul ini.
6. Link repository harus berbeda untuk setiap tugasnya. Buatlah markdown yang rapi di setiap repository tugas yang anda kumpulkan.
7. Printscreen program harus dari desktop anda sendiri, dan harus dari linux yang sudah diinstall. Jika tidak, maka harus mengulang pengerjaan tugasnya.
8. Jangan lupa untuk menuliskan NAMA dan NIM pada laporan.
9. Laporan berbentuk PDF dan dikumpulkan pada link repository github beserta kodenya.

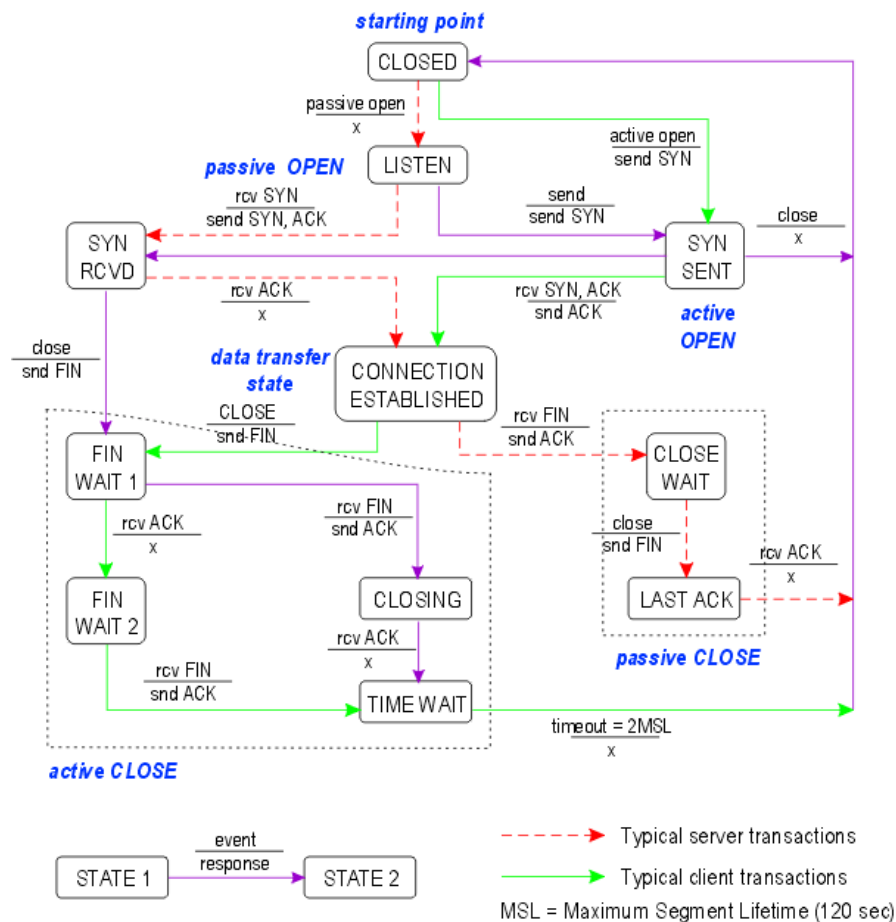
Nama:

NIM:

Nilai:

### Soal No 1

#### TCP finite state machine



14.

15. Jelaskan maksud diagram finite state machine dari TCP Connection diatas!

Jawaban:

Finite State Machine diatas menjelaskan alur proses pembuatan koneksi TCP yang disebut dengan three-way handshaking.

Dari sisi client :

1. Client mengirim SYN (SYN SENT)
2. Client menerima SYN dan ACK, kemudian mengirim ACK (CONNECTION ESTABLISHED)
3. Client mengirimkan FIN (FIN WAIT 1)
4. Client menerima ACK (FIN WAIT 2)
5. Client menerima FIN dan mengirim ACK (TIME WAIT)
6. Terjadi timeout, maka koneksi berakhir (CLOSED)

Dari sisi server :

1. Server membuat koneksi secara passive open (LISTEN)

Nama:	NIM:	Nilai:
-------	------	--------

2. Server menerima SYN, kemudian mengirim SYN, ACK (SYN RCVD)
3. Server menerima ACK (CONNECTION ESTABLISHED)
4. Server menerima FIN, kemudian mengirimkan ACK (CLOSE WAIT)
5. Server mengirim FIN (LAST ACK)
6. Server mengakhiri koneksi (CLOSED)

Nama:	NIM:	Nilai:
-------	------	--------

### Soal No 2 (for dan if/else)

```

16. package main
17. import "fmt"
18. func main() {
19.     i := 1
20.     for i <= 3 {
21.         fmt.Println(i)
22.         i = i + 1
23.     }
24.     for j := 7; j <= 9; j++ {
25.         fmt.Println(j)
26.     }
27.     for {
28.         fmt.Println("loop")
29.         break
30.     }
31.     for n := 0; n <= 5; n++ {
32.         if n%2 == 0 {
33.             continue
34.         }
35.         fmt.Println(n)
36.     }
37. }

```

```

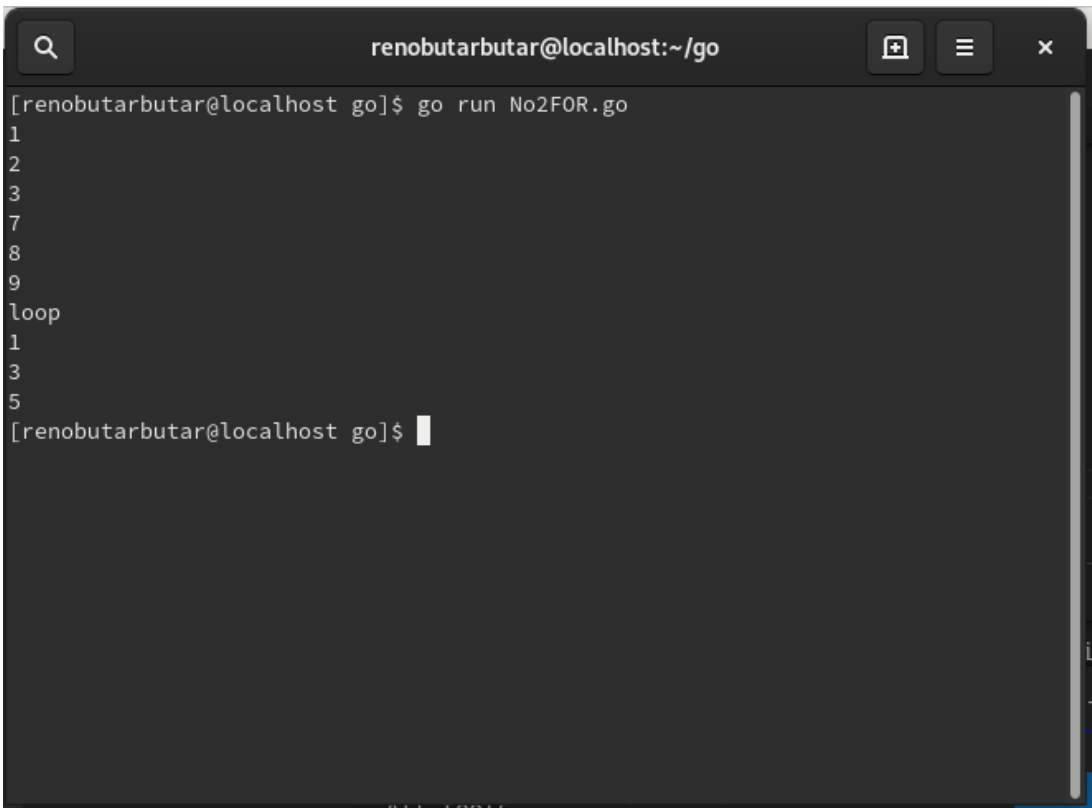
package main
import "fmt"
func main() {
    if 7%2 == 0 {
        fmt.Println("7 is even")
    } else {
        fmt.Println("7 is odd")
    }
    if 8%4 == 0 {
        fmt.Println("8 is divisible by 4")
    }
    if num := 9; num < 0 {
        fmt.Println(num, "is negative")
    } else if num < 10 {
        fmt.Println(num, "has 1 digit")
    } else {
        fmt.Println(num, "has multiple digits")
    }
}

```

Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Nama:	NIM:	Nilai:
-------	------	--------

Jawaban:



```
renobutarbutar@localhost:~/go
[renobutarbutar@localhost go]$ go run No2FOR.go
1
2
3
7
8
9
loop
1
3
5
[renobutarbutar@localhost go]$
```

Cara kerja :

- For merupakan instruksi untuk melakukan perulangan sesuai dengan kondisi yang ditentukan
- Perulangan for dapat berhenti apabila kondisi sudah false, atau jika ada break
- Perulangan for juga dapat di-skip dengan menggunakan continue

Nama:	NIM:	Nilai:
-------	------	--------

```

renobutarbutar@localhost:~/go
[renobutarbutar@localhost go]$ go run No2IFELSE.go
7 is odd
8 is divisible by 4
9 has 1 digit
[renobutarbutar@localhost go]$

```

Cara kerja :

- If akan dijalankan jika kondisinya true
- Jika kondisi if false, maka akan berlanjut ke else if setelahnya
- Jika semua if kondisinya false, maka yang dijalankan adalah else

### Soal No 3 (array dan function)

```

27. package main
import "fmt"
func main() {
    var a [5]int
    fmt.Println("set:", a)

    a[4] = 100
    fmt.Println("set:", a)
}

```

```

package main
import "fmt"
func plus(a int, b int) int {
    return a + b
}
func plusPlus(a, b, c int) int {

```

Tug

020

dari 15

Nama:	NIM:	Nilai:
-------	------	--------

- 28.
- 29.
- 30.
- 31.
- 32.
- 33.
- 34.
- 35.
- 36.
37. Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Jawaban:



```

renobutarbutar@localhost:~/go
[renobutarbutar@localhost go]$ go run No3a.go
emp; [0 0 0 0 0]
set: [0 0 0 0 100]
get: 100
len: 5
dcl: [1 2 3 4 5]
2d: [[0 1 2] [1 2 3]]
[renobutarbutar@localhost go]$

```

Cara kerja :

Array merupakan sekumpulan variabel yang memiliki tipe data yang sama

Array mempunyai indeks untuk menentukan nilainya

Panjang dari suatu array dapat diketahui dengan fungsi `len(variabelArray)`

Array bisa berbentuk 2 dimensi atau lebih

Nama:	NIM:	Nilai:
-------	------	--------

```

renobutarbutar@localhost:~/go
[renobutarbutar@localhost go]$ go run No3b.go
1+2 = 3
1+2+3 = 6
[renobutarbutar@localhost go]$

```

Cara kerja :

Function adalah sekumpulan pernyataan yang akan dijalankan jika namanya dipanggil  
Function dapat mengembalikan bermacam-macam tipe data

#### Soal No 4 (struct dan method)

```

38 package main
   import "fmt"
39 type person struct {
   name string
   age int
40 }
41 func main() {
   fmt.Println(person{"Bob", 20})
42   fmt.Println(person{name: "Alice", age: 30})
43   fmt.Println(person{name: "Fred"})
   fmt.Println(&person{name: "Ann", age: 40})
   s := person{name: "Sean", age: 50}
   fmt.Println(s.name)
   sp := &s
   fmt.Println(sp.age)
   sp.age = 51

```

Tu

020

```

package main
import "fmt"
type rect struct {
width, height int
}
func (r *rect) area() int {
return r.width * r.height
}
func (r rect) perim() int {
return 2*r.width + 2*r.height
}
func main() {
r := rect{width: 10, height: 5}
fmt.Println("area: ", r.area())
fmt.Println("perim:", r.perim())
rp := &r
fmt.Println("area: ", rp.area())

```

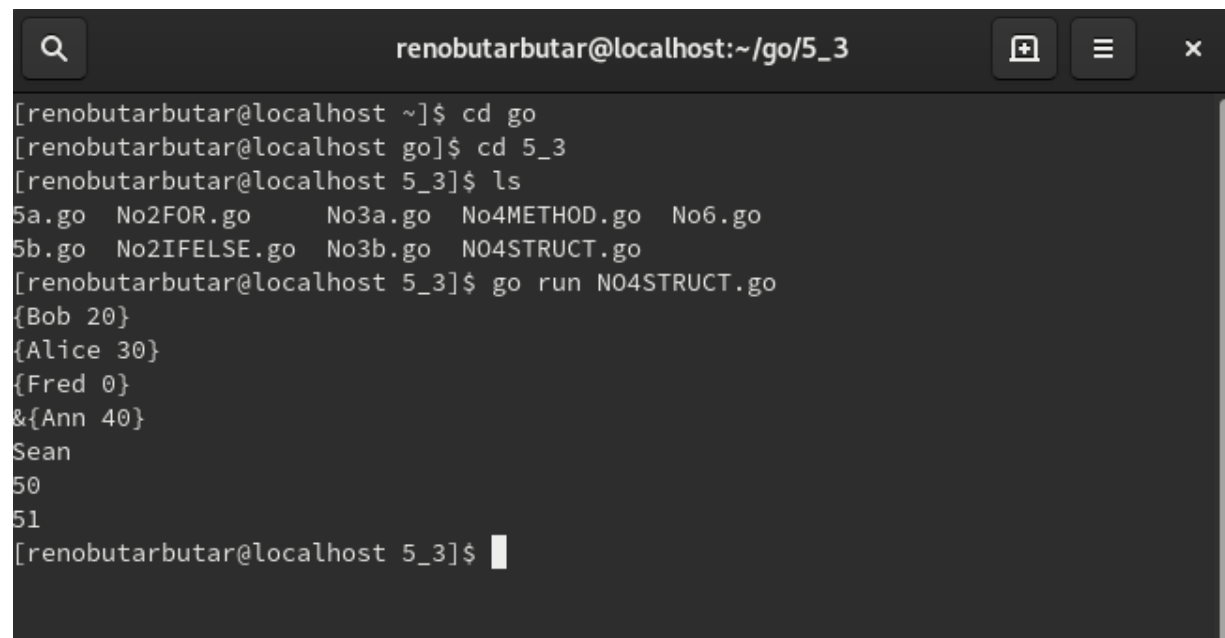
dari 15



Nama:	NIM:	Nilai:
-------	------	--------

- 44.
- 45.
- 46.
- 47.
48. Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Jawaban:



```

renobutarbutar@localhost:~/go/5_3
[renobutarbutar@localhost ~]$ cd go
[renobutarbutar@localhost go]$ cd 5_3
[renobutarbutar@localhost 5_3]$ ls
5a.go  No2FOR.go    No3a.go  No4METHOD.go  No6.go
5b.go  No2IFELSE.go  No3b.go  N04STRUCT.go
[renobutarbutar@localhost 5_3]$ go run N04STRUCT.go
{Bob 20}
{Alice 30}
{Fred 0}
&{Ann 40}
Sean
50
51
[renobutarbutar@localhost 5_3]$

```

Cara kerja :

Struct adalah instruksi untuk membuat tipe data bentukan.

Sebuah struct bisa mempunyai berbagai variabel yang tipe datanya berbeda.

Jika pada saat pemanggilan struct terdapat variabel yang tidak didefinisikan nilainya, maka nilainya menjadi 0 atau nil.

Nama:	NIM:	Nilai:
-------	------	--------

```

renobutarbutar@localhost:~/go/5_3
[renobutarbutar@localhost 5_3]$ ls
5a.go  No2FOR.go      No3a.go  No4METHOD.go  No6.go
5b.go  No2IFELSE.go    No3b.go  NO4STRUCT.go
[renobutarbutar@localhost 5_3]$ go run No4METHOD.go
area: 50
perim: 30
area: 50
perim: 30
[renobutarbutar@localhost 5_3]$

```

Cara kerja :

Method adalah fungsi yang memiliki akses ke properti struct.

Perbedaannya dengan fungsi, pada saat deklarasi method ditentukan struct dari method tersebut.

Untuk memanggilnya juga harus diawali dengan variabel struct terlebih dahulu.

#### Soal No 5 (multiple return value dan command line)

```

49. package main
50. import "fmt"
51. func vals() (int, int) {
52.     return 3, 7
53. }
54. func main() {
55.     a, b := vals()
    fmt.Println(a)
    fmt.Println(b)
    _, c := vals()
    fmt.Println(c)
}

```

```

package main

import "flag"
import "fmt"

func main() {
    wordPtr := flag.String("word", "foo", "a string")

    numbPtr := flag.Int("numb", 42, "an int")
    boolPtr := flag.Bool("fork", false, "a bool")

    var svar string
    flag.StringVar(&svar, "svar", "bar", "a string var")

    flag.Parse()

    fmt.Println("word:", *wordPtr)
    fmt.Println("numb:", *numbPtr)
    fmt.Println("fork:", *boolPtr)
    fmt.Println("svar:", svar)
    fmt.Println("tail:", flag.Args())
}

```

Tugas  
15

2020

ari

Nama:	NIM:	Nilai:
-------	------	--------

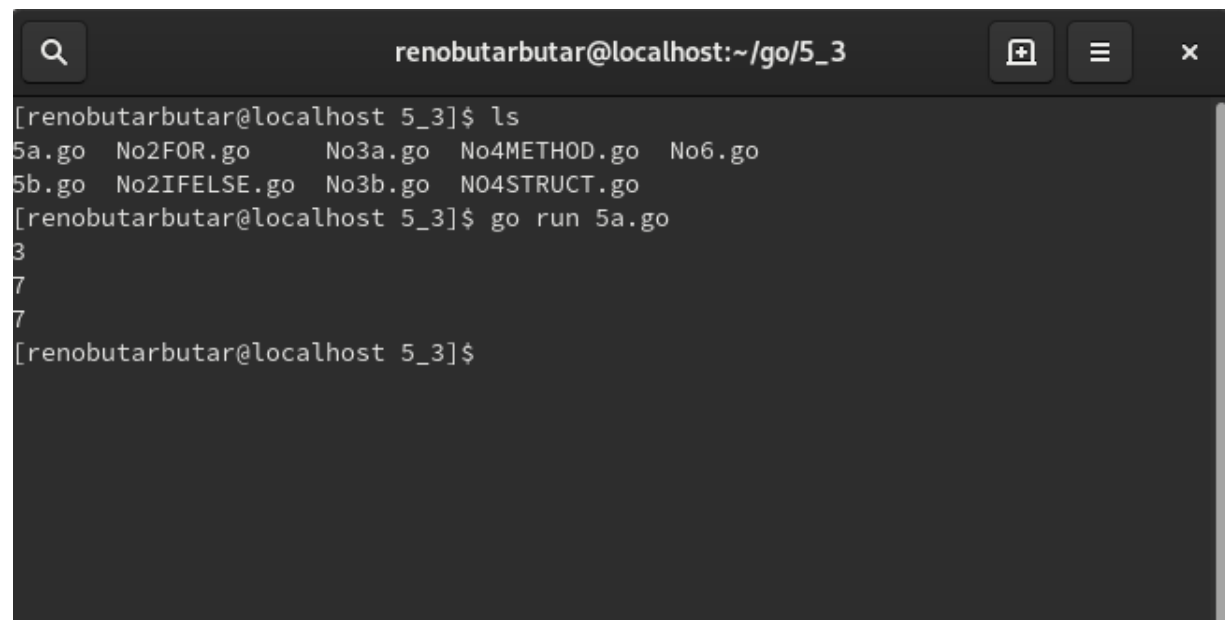
56.

57.

58.

59. Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Jawaban:



```

renobutarbutar@localhost:~/go/5_3
[renobutarbutar@localhost 5_3]$ ls
5a.go  No2FOR.go    No3a.go  No4METHOD.go  No6.go
5b.go  No2IFELSE.go No3b.go  NO4STRUCT.go
[renobutarbutar@localhost 5_3]$ go run 5a.go
3
7
7
[renobutarbutar@localhost 5_3]$

```

Cara kerja nya : dapat mengembalikan lebih dari 1 nilai.

Nama:	NIM:	Nilai:
-------	------	--------

```

renobutarbutar@localhost:~/go/5_3
[renobutarbutar@localhost 5_3]$ go run 5b.go
word: foo
numb: 42
fork: false
svar: bar
tail: []
[renobutarbutar@localhost 5_3]$

```

Cara kerja nya :

### Soal No 6 (simple web application)

```

60. package main
61. import (
    "fmt"
    "net/http"
)

func main() {
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintf(w, "Hello, you've requested: %s\n", r.URL.Path)
    })

    http.ListenAndServe(":80", nil)
}

```

Tugas  
15

ri

Nama:	NIM:	Nilai:
-------	------	--------

62.

63.

64.

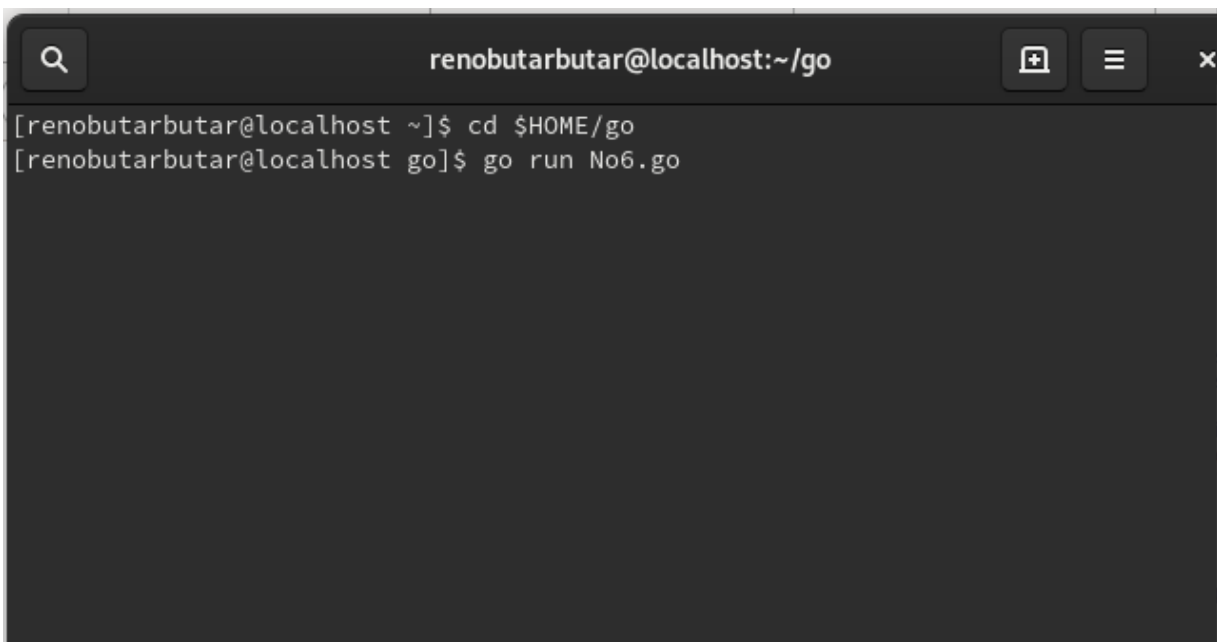
65.

66.

67. Sebelum menjalankan program diatas, gantilah port 80 ke port 8000. Buka browser kemudian ketikkan alamat localhost:8000.

68. Jalankan program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Jawaban:



```

renobutarbutar@localhost:~/go
[renobutarbutar@localhost ~]$ cd $HOME/go
[renobutarbutar@localhost go]$ go run No6.go

```

Nama:	NIM:	Nilai:
-------	------	--------



Cara kerja:

Go bisa digunakan untuk membuat aplikasi web.

HandleFunc berfungsi untuk menentukan route dan menentukan konten web ketika alamat diakses.

ListenAndServe berfungsi menentukan port dan menjalankan aplikasi pada port yang ditentukan.

#### Soal No 7 (create config file)

Buatlah sebuah config file untuk aplikasi web application pada soal no 6 dengan menggunakan library berikut: <https://github.com/spf13/viper> !

Jelaskan susunan directory dari program serta bagaimana cara untuk melakukan konfigurasi file config yang telah anda buat!

71. Printscreen hasil dan penjelasan kode untuk membuat file config disini!

Nama:	NIM:	Nilai:
-------	------	--------

Jawaban: