


Nama: Sella Tresnasari	NIM: 1301188565	Nilai:
------------------------	-----------------	--------

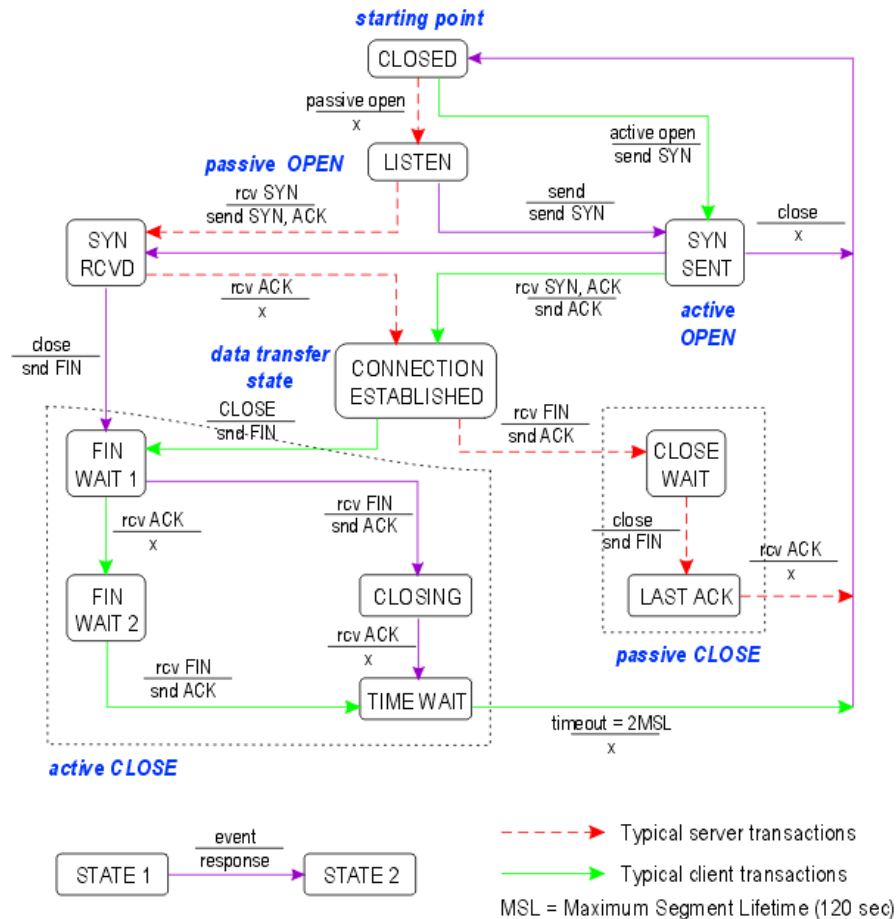
	<b>Tugas 1</b> <b>Pemrograman Jaringan (CSH4V3)</b> ----- <b>Semester Ganjil 2019 - 2020</b> <b>Dosen: Aulia Arif Wardana, S.Kom., M.T. (UIW)</b>	
	<p><i><b>Berdo'alah sebelum mengerjakan. Dilarang berbuat curang.</b></i></p> <p><i><b>Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.</b></i></p> <p><i><b>Selamat belajar, semoga sukses !</b></i></p>	
<b>Nama Mahasiswa:</b> <b>Sella Tresnasari</b> .....	<b>NIM:</b> <b>1301188565</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.

## Soal No 1

## TCP finite state machine



14.

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

Jawaban:

Finite State Machine adalah sebuah metode perancangan system control yang menggambarkan prinsip kerja system dengan menggunakan state (keadaan), event (kejadian), action (aksi).

# Server

- Server dimulai pada Starting Point
- Server menerima SYN dan mengirim SYN, ACK ke SYN RCVD
- Server menerima ACK di Connection Established
- Server menerima FIN dan mengirim ACK ke Close Wait
- Server mengirim FIN ke Last ACK
- Server menerima ACK lalu server CLOSE

Nama: Sella Tresnasari	NIM: 1301188565	Nilai:
------------------------	-----------------	--------

#### # Client

- Client mulai pada Starting Point
- Client melakukan Active Open
- Client mengirim SYN ke SYN SENT
- Client menerima SYN, ACK dan mengirim ACK ke Connection Established
- Client close dan mengirim FIN ke FIN WAIT 1
- Client menerima ACK di FIN Wait 2
- Client menerima FIN dan mengirim ACK ke Time Wait
- Client Time Out lalu ke CLOSE (berakhir)

**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!

**# FOR**

- Melakukan perulangan dengan kondisi yang ditentukan (i=1)
- Selama i <= 3 program melakukan looping
- Perulangan berhenti ketika kondisi false
- Perulangan dapat dihentikan menggunakan break dan dilanjutkan dengan continue

**# IF ELSE**

- Jika IF memiliki kondisi true/ benar maka program dijalankan
- Jika kondisi berbeda maka program akan menjalankan kondisi di else

Nama: Sella Tresnasari	NIM: 1301188565	Nilai:
------------------------	-----------------	--------

Jawaban:

# Printscreen 1

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N02A.go
1
2
3
7
8
9
loop
1
3
5
```

# Printscreen 2

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N02B.go
7 is odd
8 divisible by 4
9 has 1 digit
```

### Soal No 3 (array dan function)

```
28. package main
import "fmt"
29. func main() {
30.     var a [5]int
    fmt.Println("emp:", a)
31.
    a[4] = 100
32.     fmt.Println("set:", a)
    fmt.Println("get:", a[4])
33.     fmt.Println("len:", len(a))
34.     b := [5]int{1, 2, 3, 4, 5}
    fmt.Println("dcl:", b)
35.     var twoD [2][3]int
    for i := 0; i < 2; i++ {
        for j := 0; j < 3; j++ {
36.             twoD[i][j] = i + j
        }
    }
    fmt.Println("2d: ", twoD)
}
```

```
package main
import "fmt"
func plus(a int, b int) int {
    return a + b
}
func plusPlus(a, b, c int) int {
    return a + b + c
}
func main() {
    res := plus(1, 2)
    fmt.Println("1+2 =", res)
    res = plusPlus(1, 2, 3)
    fmt.Println("1+2+3 =", res)
}
```

Nama: Sella Tresnasari	NIM: 1301188565	Nilai:
------------------------	-----------------	--------

37.

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

Jawaban:

# Array

- Array merupakan kumpulan data bertipe sama, yang disimpan dalam sebuah variabel
- Array juga dapat digunakan untuk mengukur suatu panjang dari array tersebut

# Function

- Fuction/fungsi sekumpulan blok kode yang dibungkus dengan nama tertentu, seperti function plus akan menjalankan perintah a+b sedangkan function plusPlus akan menjalankan perintah a+b+c

# Printscreen 1

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N03A.go
emp : [0 0 0 0 0]
set : [0 0 0 0 100]
get : 100
len : 5
dc1 : [1 2 3 4 5]
2d : [[0 1 2] [1 2 3]]
```

# Printscreen 2

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N03B.go
1+2 = 3
1+2+3 = 6
```

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

```

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

    sp.age = 51
    fmt.Println(sp.age)
    }

```

```

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())
    fmt.Println("perim:", rp.perim())
}

```

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

# Printscreen 1

```

localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N04A.go
{Bob 20}
{alice 30}
{Fred 0}
&{Ann 40}
sean
50
51

```

# Printscreen 2

```

localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N04B.go
area : 50
perim : 30
area : 50
perim : 30

```

Nama: Sella Tresnasari	NIM: 1301188565	Nilai:
------------------------	-----------------	--------

Jawaban:

#### # Struct

- Struct merupakan kumpulan definisi dari variabel dan fungsi, yang dibungkus sebagai tipe data baru dengan nama tertentu
- Variabel dalam struct dapat memiliki tipe data yang bervariasi.
- Pada function main terdapat beberapa perintah yang bias menampilkan 2 tipe data yg berbeda seperti:  
fmt.Println(person{"Bob", age: 30})  
output : {Bob 20}

#### # Method

- Method adalah suatu fungsi yang menempel pada tipe data seperti struct atau tipe data lainnya
- Method bisa diakses lewat variabel objek.
- Method memiliki akses ke property struct.
- Pada soal kedua struct telah membuat inisialisasi pada variabel width dan height, dan method memanggil struct.
- Pada soal yang kedua method memanggil "rect" pada struct untuk menjalankan method perkalian width dengan height (line 39-41)

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

```

50. package main
51. import "fmt"
52. func vals() (int, int) {
53.     return 3, 7
54. }
55. func main() {
56.     a, b := vals()
57.     fmt.Println(a)
58.     fmt.Println(b)
59.     _, c := vals()
60.     fmt.Println(c)
61. }

```

```

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

```

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



Nama: Sella Tresnasari	NIM: 1301188565	Nilai:
------------------------	-----------------	--------

Jawaban:

#### # Multiple Return

- Multiple Return merupakan sebuah fungsi yang bisa mengembalikan lebih dari 1 nilai yang dilakukan sebuah fungsi.
- Pada func vals () (int,int) {return 3,7} lalu pada func main() nilai 3 dan 7 diambil lalu ada pengembalian nilai c maka mengeluarkan output an 3 7 7

#### # Command Line (flag)

- Command line dapat mendeklarasi flag yang berupa string, integer, dan Boolean
- Flag di deklarasi kan di dalam sebuah function
- Pada soal kedua terdapat beberapa jenis flag yaitu : string, integer, Boolean
- Pada pemanggilan program menampilkan output yang sesuai dengan jenis flag seperti pada line 53 flag membuat perintah string dan meampilkan output "numb: 42"

#### # Printscreen 1

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N05A.go
3
7
7
```

#### # Printscreen 2

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/sellatr/TUGAS1_N05B.go
word foo
numb 42
bool false
svar bar
tail []
```

#### Soal No 6 (simple web application)

```
60. package main
61. import (
62.     "fmt"
63.     "net/http"
64. )
65. func main() {
66.     http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
67.         fmt.Fprintf(w, "Hello, you've requested: %s\n", r.URL.Path)
68.     })
69.     http.ListenAndServe(":80", nil)
70. }
```

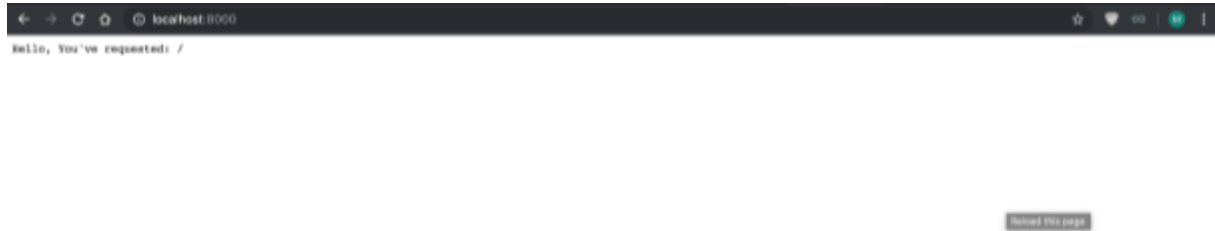
67. Sebelum menjalankan program diatas, gantilah port 80 ke port 8000. Buka browser kemudian ketikkan alamat localhost:8000.

Nama: Sella Tresnasari	NIM: 1301188565	Nilai:
------------------------	-----------------	--------

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

Jawaban:

Terdapat program client server dimana program yang berjalan sebagai server menunggu client request untuk mengakses program melalui localhost:8000.  
Client memanggil localhost:8000 sebagai berikut:



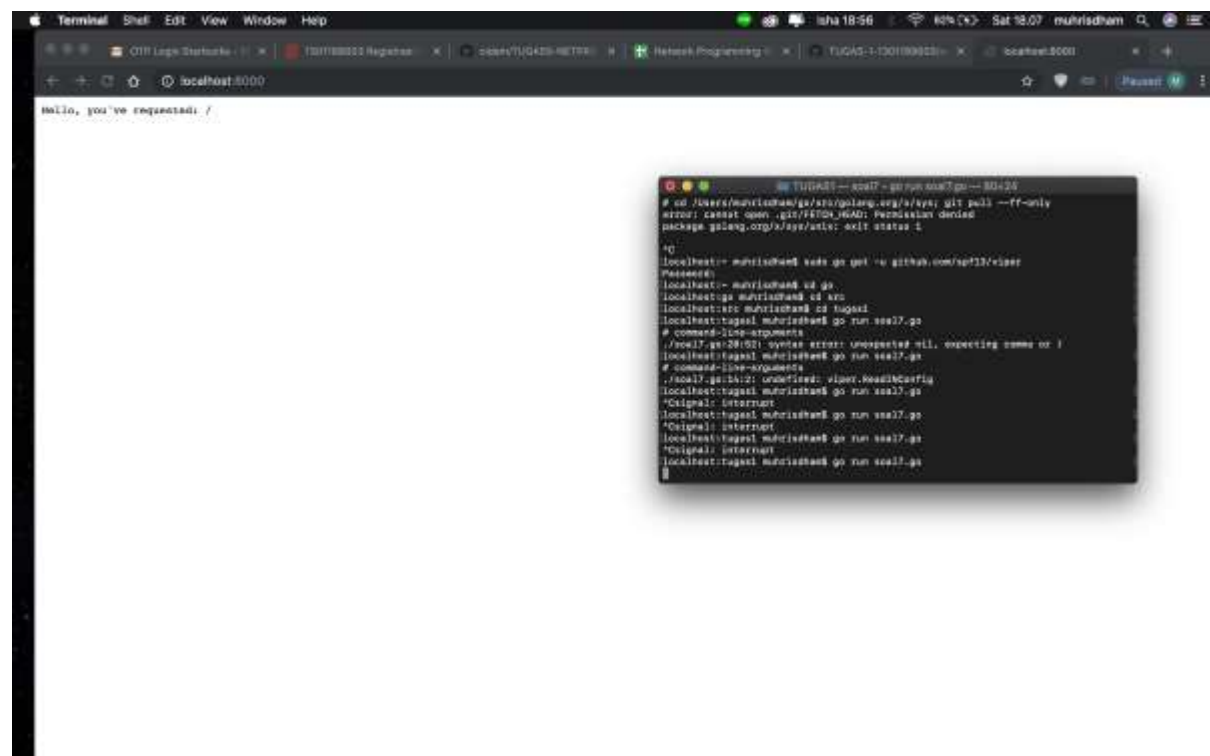
#### Soal No 7 (create config file)

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!

Jawaban:



```
Terminal Shell Edit View Window Help
localhost:8000
hallo, you've requested: /

TUGAS-1-1301188565
# cd /home/muhridham/go/src/golang.org/x/ys; git pull --ff-only
error: cannot open git/FETCH_HEAD: Permission denied
package golang.org/x/ys/ank: exit status 1

localhost:~ muhridham$ cd go get -u github.com/spf13/viper
# command-line-arguments
./ss27.go:28:21: syntax error: unexpected nil, expecting comma or {
localhost:~ muhridham$ cd go
localhost:~ muhridham$ cd src
localhost:src muhridham$ cd tugas1
localhost:tugas1 muhridham$ go run ss27.go
# command-line-arguments
./ss27.go:28:21: syntax error: unexpected nil, expecting comma or {
localhost:tugas1 muhridham$ go run ss27.go
# command-line-arguments
./ss27.go:28:21: syntax error: unexpected nil, expecting comma or {
localhost:tugas1 muhridham$ go run ss27.go
^Csignal: interrupt
localhost:tugas1 muhridham$ go run ss27.go
^Csignal: interrupt
localhost:tugas1 muhridham$ go run ss27.go
^Csignal: interrupt
localhost:tugas1 muhridham$ go run ss27.go
^Csignal: interrupt
localhost:tugas1 muhridham$ go run ss27.go
```

untuk program config.json digunakan untuk konfigurasi. Pada file config.json terdapat tipe data untuk server dan memiliki atribut yang berupa port. Fungsi http pada Go dapat digunakan untuk mengambil nomor port dari file config.json dengan menggunakan viper (sintaks `viper.GetString("server.port")`), yang nantinya akan mengembalikan nilai port dari tipe data server didalam file config.json.