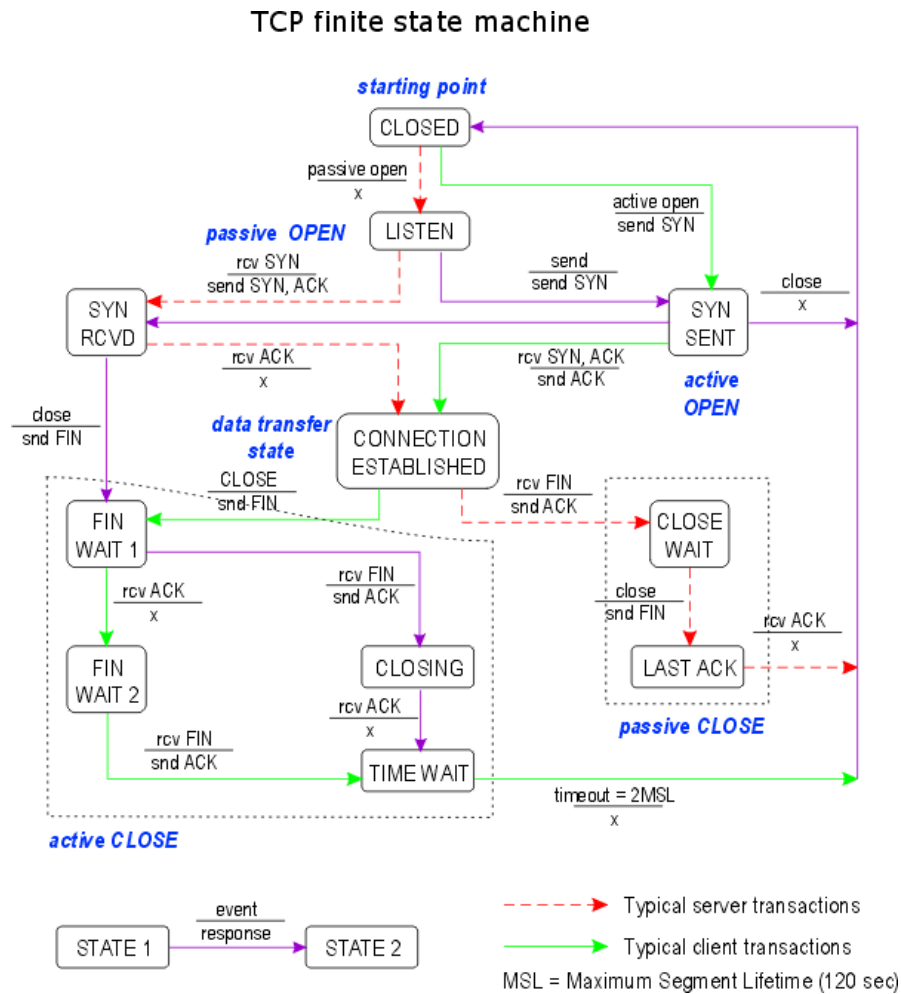


***Berdo'alah sebelum mengerjakan. Dilarang berbuat curang.
Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.
Selamat belajar, semoga sukses !***

Nama Mahasiswa:
I Putu Surya Baratha
.....**NIM:**
1301188566
.....**Nilai:**
.....**Siapkan tools berikut sebelum mengerjakan:**

1. Go Programming Language (<https://golang.org/dl/>).
2. VisualStudioCode (<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



Jelaskan maksud diagram finite state machine dari TCP Connection diatas!

Jawaban:

Finite State Machine adalah model logis untuk perilaku sistem yang kondisi internal berubah dengan terjadinya peristiwa yang ditentukan

Server:

- Server dimulai starting point
- Server membuat koneksi secara passive open(LISTEN)
- Server menerima SYN lalu mengirimkan SYN,ACK ke SYN RCVD
- Server menerima ACK di CONNECTION ESTABLISHED
- Server menerima FIN lalu mengirimkan ACK ke CLOSE WAIT
- Server mengirimkan FIN ke LAST ACK
- Server menerima ACK lalu mengakhiri koneksi(CLOSED)

Nama: I Putu Surya Baratha	NIM: 1301188566	Nilai:
<p>Client:</p> <ul style="list-style-type: none"> - Client active open - CLient mengirimkan SYN ke SYN SENT - Client menerima SYN,ACK kemudian mengirimkan ACK ke CONNECTION ESTABLISHED - Client mengirimkan FIN ke FIN WAIT 1 - Client menerima ACK ke FIN WAIT 2 - Client menerima FIN lalu mengirimkan ACK ke TIME WAIT - Timeout, maka koneksi berakhir(CLOSED) 		

Soal No 2 (for dan if/else)

```
package main
import "fmt"
func main() {
    i := 1
    for i <= 3 {
        fmt.Println(i)
        i = i + 1
    }

    for j := 7; j <= 9; j++ {
        fmt.Println(j)
    }

    for {
        fmt.Println("loop")
        break
    }

    for n := 0; n <= 5; n++ {
        if n%2 == 0 {
            continue
        }
        fmt.Println(n)
    }
}
```

```
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!

Jawaban:

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_N02A.go
1
2
3
7
8
9
loop
1
3
5
```

Melakukan perulangan for sesuai dengan nilai yang sudah ditentukan dengan nilai awal i=1. Didalam perulangan for yang pertama mencetak nilai i selama i<=3, maka hasilnya akan 1 2 3. Didalam perulangan for yang kedua mencetak nilai j selama j<=9 dimana nilai awal j adalah 7, maka hasilnya adalah 7 8 9. Perulangan for yang ketiga adalah loop forever yang mencetak string "loop".

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_N02B.go
7 is odd
8 divisible by 4
9 has 1 digit
```

Pada program if/else akan melakukan pengecekan pada suatu kondisi. Jika kondisi tersebut terpenuhi maka akan melakukan aksi dari kondisi tersebut.

Soal No 3 (array dan function)

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

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

    fmt.Println("len:", len(a))

    b := [5]int{1, 2, 3, 4, 5}
    fmt.Println("dcl:", b)

    var twoD [2][3]int
    for i := 0; i < 2; i++ {
        for j := 0; j < 3; j++ {
            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)
}
```

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

Jawaban:

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_N03A.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]]
```

Array adalah sekumpulan data yang sama, yang disimpan dalam sebuah variabel. Array memiliki kapasitas yang nilainya ditentukan pada saat pembuatan, menjadikan elemen/data yang disimpan di array tersebut jumlahnya tidak boleh melebihi yang sudah dialokasikan. Default nilai tiap elemen array pada awalnya tergantung dari tipe datanya. Jika int maka tiap element zero value-nya adalah 0, jika bool maka false, dan seterusnya. Setiap elemen array memiliki indeks berupa angka yang merepresentasikan posisi urutan elemen tersebut. Indeks array dimulai dari 0.

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_3B.go
1+2 = 3
1+2+3 = 6
```

Function merupakan sekumpulan pernyataan yang jika namanya dipanggil maka ia akan dijalankan. Didalam program, function memanggil func plus dan func plusPlus. Func plus mengembalikan nilai a+b, dan func plusPlus mengembalikan nilai a+b+c.

Soal No 4 (struct dan method)

```
package main

import "fmt"

type person struct {
    name string
    age  int
}

func main() {

    fmt.Println(person{"Bob", 20})

    fmt.Println(person{name: "Alice", age: 30})

    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
    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())
}
```

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

Jawaban:

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_4A.go
{Bob 20}
{alice 30}
{Fred 0}
&{Ann 40}
sean
50
51
```

Struct merupakan instruksi untuk membuat tipe data bentukan. Struct mempunyai berbagai macam variabel yang tipe datanya berbeda. Saat pemanggilan struct jika terdapat variabel yang tidak terdefinisi nilainya, maka nilai tersebut akan menjadi 0 atau nil

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_4B.go
area : 50
perim : 30
area : 50
perim : 30
```

Method merupakan fungsi yang memiliki akses terhadap properti struct. Untuk memanggilkannya harus diawali dengan variabel struct.

Soal No 5 (multiple return value dan command line)

```
package main

import "fmt"

func vals() (int, int) {
    return 3, 7
}

func main() {

    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())
}
```

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

Jawaban:

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_5A.go
3
7
7
```

Multiple Return adalah fungsi yang dapat mengembalikan lebih dari 1 nilai

```
localhost:go muhrisdham$ go run /Users/muhrisdham/go/src/IPUTUSURYA/NETPRO_5B.go
word foo
numb 42
bool false
svar bar
tail []
```

Command line merupakan interaksi dengan system operasi atau perangkat lunak computer dengan cara mengetik perintah untuk menjalankannya.

Soal No 6 (simple web application)

```
package main

import (
    "fmt"
    "net/http"
)

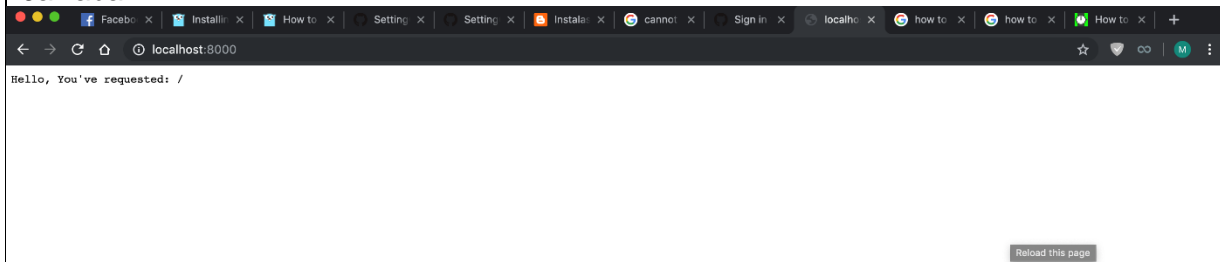
func main() {
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintf(w, "Hello, you've requested: %s\n", r.URL.Path)
    })

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

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

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

Jawaban:



Didalam program, handlefunc berfungsi untuk menentukan route/konten ketika akan diakses, dan listenAndServe berfungsi untuk menentukan port yang digunakan ketika ingin mengakses

Nama: I Putu Surya Baratha	NIM: 1301188566	Nilai:
----------------------------	-----------------	--------

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!

Printscreen hasil dan penjelasan kode untuk membuat file config disini!

Jawaban: