

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

Nama Mahasiswa:
Mazaya Z D

NIM:
1301154508

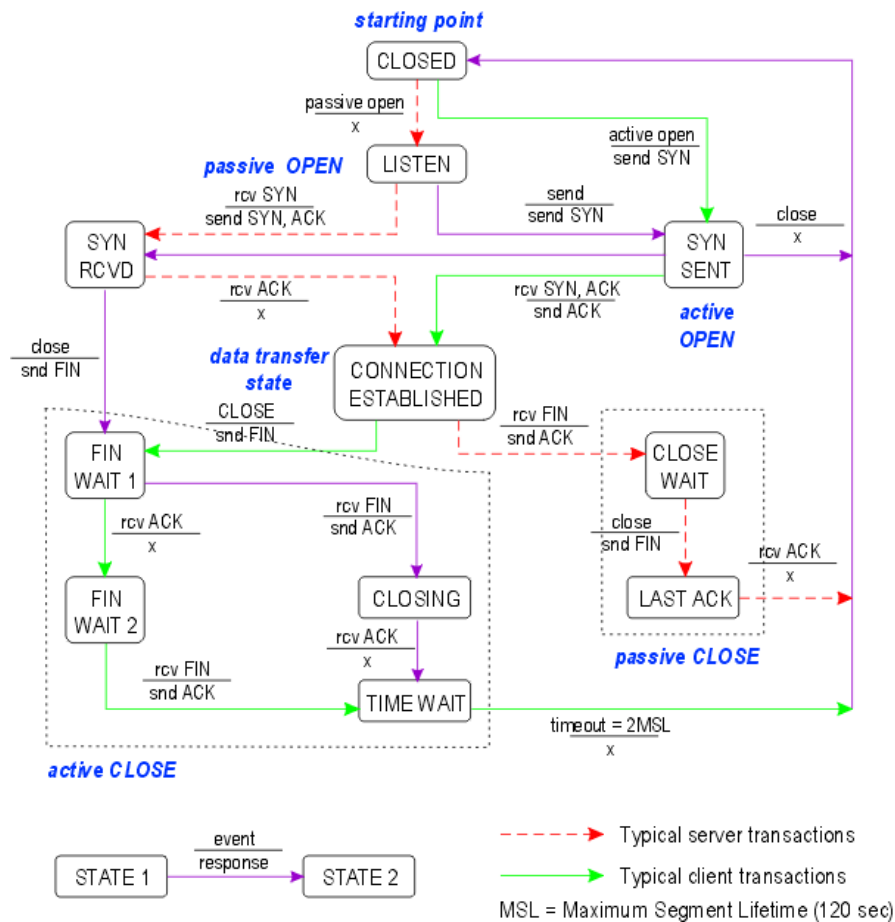
Nilai:

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. Disarankan untuk 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. Kumpulkan link repository github tersebut sebagai tanda bahwa anda mengerjakan tugas modul ini.

Soal No 1

TCP finite state machine



Jelaskan maksud diagram finite state machine dari TCP Connection diatas!

Jawaban:

SYN (*Synchronize*) -> digunakan untuk memulai dan membuat koneksi. Dinamakan demikian karena salah satu fungsinya adalah untuk menyinkronkan nomor urut antar perangkat.

FIN (*Final*) -> yang merupakan segmen TCP dengan set bit *FIN*, menunjukkan bahwa perangkat ingin memutuskan koneksi.

ACK (*Acknowledgment*) -> menunjukkan tanda terima pesan seperti *SYN* atau *FIN*.

State	State Description	Event and Transition
CLOSED	Ini adalah status default yang digunakan oleh setiap koneksi sebelum	Passive Open: Server memulai proses pengaturan koneksi dengan melakukan

Nama: Mazaya Z D	NIM: 1301154508	Nilai:
------------------	-----------------	--------

	<p>proses pembentukannya dimulai. Dalam standarnya <i>state</i> disebut “fictional”. Alasannya adlh bahwa keadaan ini mewakili situasi di mana tidak ada koneksi antara perangkat — itu belum dibuat, atau baru saja dihancurkan.</p>	<p>buka pasif pada port TCP. Pada saat yang sama, mengatur struktur data (<i>transmission control block or TCB</i>) yang diperlukan untuk mengelola koneksi. Kemudian transisi ke <i>LISTEN state</i>.</p> <p>Active Open, Send SYN: Klien memulai pengaturan koneksi dengan mengirim pesan <i>SYN</i> , dan juga mengatur TCB untuk koneksi ini. Kemudian transisi ke keadaan <i>SYN-SENT</i> .</p>
LISTEN	<p>Perangkat (biasanya server) sedang menunggu untuk menerima pesan sinkronisasi (<i>SYN</i>) dari klien. Belum mengirim pesan <i>SYN</i> -nya sendiri.</p>	<p>Receive Client SYN, Send SYN+ACK: Perangkat server menerima <i>SYN</i> dari klien. Lalu mengirim kembali pesan yang berisi <i>SYN</i> sendiri dan juga <i>ACK</i> yang diterima. Server pindah ke status <i>SYN-RECEIVED</i> .</p>
SYN-SENT	<p>Perangkat (biasanya klien) telah mengirim pesan sinkronisasi (<i>SYN</i>) dan sedang menunggu <i>SYN</i> yang cocok dari perangkat lain (biasanya server).</p>	<p>Receive SYN, Send ACK: Jika perangkat yang telah mengirim pesan <i>SYN</i> -nya menerima <i>SYN</i> dari perangkat lain, tetapi bukan <i>ACK</i> untuk <i>SYN</i> -nya sendiri, ia mengakui <i>SYN</i> yang diterimanya dan kemudian beralih ke <i>SYN-RECEIVED</i> untuk menunggu pengakuan untuk <i>SYN</i> -nya.</p> <p>Receive SYN+ACK, Send ACK: Jika perangkat yang mengirim <i>SYN</i> menerima baik <i>ACK</i> untuk <i>SYN</i> dan juga <i>SYN</i> dari perangkat lain, itu mengakui <i>SYN</i> yang diterima dan kemudian bergerak langsung ke keadaan <i>ESTABLISHED</i> .</p>
SYN-RECEIVED	<p>Perangkat telah menerima <i>SYN</i> (permintaan koneksi) dari mitranya dan mengirim <i>SYN</i> sendiri. Sekarang sedang menunggu <i>ACK</i> ke <i>SYN</i> -nya untuk menyelesaikan pengaturan koneksi.</p>	<p>Receive ACK: Ketika perangkat menerima <i>ACK</i> ke <i>SYN</i> yang dikirimnya, ia beralih ke status <i>ESTABLISHED</i> .</p>
ESTABLISHED	<p>"Kondisi stabil" dari koneksi TCP terbuka. Data dapat dipertukarkan secara bebas begitu kedua perangkat dalam koneksi memasuki kondisi ini. Ini akan berlanjut sampai koneksi ditutup karena satu dan lain alasan.</p>	<p>Close, Send FIN: Perangkat dapat menutup koneksi dengan mengirim pesan dengan <i>FIN</i> (<i>selesai</i>) yang dikirim dan beralih ke status <i>FIN-WAIT-1</i> .</p> <p>Receive FIN: Perangkat dapat menerima pesan <i>FIN</i> dari mitra</p>

Nama: Mazaya Z D	NIM: 1301154508	Nilai:
------------------	-----------------	--------

		koneksinya yang meminta koneksi ditutup. ACK akan mengirim pesan dan transisi ke CLOSE-WAIT state .
CLOSE-WAIT	Perangkat telah menerima permintaan tutup (<i>FIN</i>) dari perangkat lain. Sekarang harus menunggu aplikasi pada perangkat lokal untuk mengakui permintaan ini dan menghasilkan permintaan yang cocok.	Close, Send FIN: Aplikasi yang menggunakan TCP, yang telah diberitahu proses lain yang ingin dimatikan, mengirim permintaan dekat ke lapisan TCP pada mesin yang menjalankannya. TCP kemudian mengirimkan <i>FIN</i> ke perangkat jarak jauh yang sudah diminta untuk memutuskan koneksi. Perangkat ini sekarang transisi ke LAST-ACK .
LAST-ACK	Perangkat yang telah menerima permintaan penutupan dan mengakuinya, telah mengirim <i>FIN</i> nya sendiri dan sedang menunggu ACK untuk permintaan ini.	Receive ACK for FIN: Perangkat menerima pemberitahuan untuk permintaan penutupannya. Sekarang telah mengirim <i>FIN</i> kami dan ACK, serta menerima <i>FIN</i> perangkat lain dan mengakuinya, jadi kami langsung menuju ke CLOSED state .
FIN-WAIT-1	Perangkat dalam kondisi ini sedang menunggu ACK untuk <i>FIN</i> yang dikirimkannya, atau sedang menunggu permintaan penghentian koneksi dari perangkat lain.	Receive ACK for FIN: Perangkat menerima pemberitahuan untuk permintaan penutupannya. Ini transisi ke status FIN-WAIT-2 . Receive FIN, Send ACK: Perangkat tidak menerima ACK untuk <i>FIN</i> -nya sendiri, tetapi menerima <i>FIN</i> dari perangkat lain. Setelah mengakuinya, dan pindah ke CLOSED state .
FIN-WAIT-2	Perangkat dalam kondisi ini telah menerima ACK untuk permintaannya untuk memutuskan koneksi dan sekarang menunggu <i>FIN</i> yang cocok dari perangkat lain.	Receive FIN, Send ACK: Perangkat menerima <i>FIN</i> dari perangkat lain. Ia mengakuinya dan pindah ke status TIME-WAIT .
CLOSING	Perangkat telah menerima <i>FIN</i> dari perangkat lain dan mengirim ACK untuknya, tetapi belum menerima ACK untuk pesan <i>FIN</i> -nya sendiri.	Receive ACK for FIN: Perangkat menerima pemberitahuan untuk permintaan penutupannya. Lalu transisi ke TIME-WAIT state .
TIME-WAIT	Perangkat sekarang telah menerima <i>FIN</i> dari perangkat lain dan mengakuinya, dan mengirim <i>FIN</i> sendiri dan menerima ACK untuk itu. Semua sudah selesai, kecuali bila menunggu untuk memastikan ACK diterima dan mencegah potensi tumpang tindih dengan koneksi baru.	Timer Expiration: Setelah periode tunggu yang ditentukan, perangkat bertransisi ke status 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 screenshot) dan jelaskan cara kerjanya!

Jawaban:

1. FOR

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soal
1
2
3
7
8
9
loop
1
3
5
[mazayaazd@localhost gp_projects]$
```

Pernyataan penjumlahan ini diulang sebanyak N kali, yaitu sebanyak angka yang dijumlahkan. Jika angka sekarang adalah i

Yang diisi 1, maka output diawali 1 dan diulang sebanyak 3 kali dimana setiap diulang i ditambah 1.

Pengulangan kedua j diisi dengan 7, setiap pengulangan ditambah 1 dan pengulangan akan berhenti jika j = 9

Pengulangan ketiga n diisi 0 dan akan berhenti jika n = 5. Hasil yg dikeluarkan jika sisa dari n dibagi 2 sama dengan 0.

2. IF/ELSE

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soalsatu
7 is odd
8 is divisible 4
9 has 1 digit
[mazayaazd@localhost gp_projects]$
```

Pernyataan if/else pertama contoh masalah dengan dua kasus, yang berarti bahwa aksi₁ dikerjakan jika kondisi bernilai benar, sebaliknya jika kondisi bernilai salah, maka aksi₂ yang akan dilaksanakan. Perhatikan bahwa else menyatakan ingkaran (negation) dari kondisi.

Pernyataan if/else kedua contoh masalah

dengan satu kasus, yang berarti bahwa aksi hanya dilaksanakan bila kondisi bernilai benar (*true*). Bila kondisi bernilai salah (*false*), tidak ada aksi apa pun yang dikerjakan.

Pernyataan if/else ketiga contoh masalah tiga kasus, yang berarti bahwa aksi₁ dikerjakan jika kondisi benar, jika kondisi bernilai salah masuk ke aksi₂ dan jika kondisi menjadi bernilai benar aksi₂ akan dilaksanakan. Jika kondisi masih bernilai salah maka aksi₃ yang akan dilaksanakan.

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 screenshot) dan jelaskan cara kerjanya!

Jawaban:

1. ARRAY

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soaldua
emp: [0 0 0 0 0]
set: [0 0 0 0 100]
get: 100
len: 5
dcl: [1 2 3 4 5]
2d: [[0 1 2] [1 2 3]]
[mazayaazd@localhost gp_projects]$
```

Program array yang mendefinisikan banyaknya elemen. Dimana variable a untuk "emp" disimpan di a dengan ukuran lariknya 5.

Pada baris selanjutnya variable a di set dimana pada penyimpanan setelah indeks 4 yaitu indeks 5 akan menyimpan nilai 100 didalamnya.

Lalu untuk array bernama "get" memanggil array yang berisi nilai 100 untuk dihasilkan. Selanjutnya array bernama "len" akan menghasilkan panjang array yang di inputkan.

Array bernama "dcl" yang terdiri dari 5 bilangan decimal akan dimasukkan di wadah b lalu dipanggil kembali untuk hasil.

2. FUNCTION

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soaltiga
1+2 = 3
1+2+3 = 6
[mazayaazd@localhost gp_projects]$
```

Fungsi plus akan mengembalikan nilai ke titik pemanggilan dengan pernyataan a + b

Fungsi plusPlus akan mengembalikan nilai ke titik pemanggilan dengan pernyataan a + b + c

Program utama melakukan input variable plus dan plusPlus kemudian hasil dikirimkan lewat res. Dengan demikian hasil perhitungan penjumlahan dalam program utama dengan menggunakan variable res.

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:

1. STRUCT

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soalempat
{Bob 20}
{Alice 30}
{Fred 0}
&{Ann 40}
Sean
50
51
[mazayaazd@localhost gp_projects]$
```

Fungsi `fmt.Println` untuk menampilkan nilai-nilai variable yang ditunjuk oleh variable `age`. Hal ini berguna untuk menunjukkan proses pertukaran data didalam fungsi.

Perintah 's' sebelumnya diisi oleh variable 'name' dan 'age' dimana hasil yang ditampilkan hanya nama nya saja yaitu 'sean'.

Perintah 'sp' akan meng-copy data terakhir yang ditunjukkan variable pointer 's' ke variable pointer 'sp'.

Perintah 'sp.age' akan meng-copy data yang di berikan pada hasil yg dikeluarkan.

2. METHOD

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soallima
area: 50
perim: 30
area: 50
perim: 30
[mazayaazd@localhost gp_projects]$
```

Untuk memecahkan program menjadi bagian-bagian yang sederhana. Hal yang perlu dilakukan yaitu mendeklarasikan variable `width`, `height`, method `area` dan method `perim`.

dengan parameter `width` dan `height` sebagai variable lokal. Kemudian 'r' sebagai variable global.

Selanjutnya pemanggilan fungsi, dimana fungsi yang sudah terbentuk dapat dipanggil dengan cara mengetikkan nama fungsi diikuti dengan parameter.

Selanjutnya fungsi `area` bertipe integer

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:

1. MULTIPLE RETURN VALUE

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soalenam
3
7
7
[mazayaazd@localhost gp_projects]$
```

Fungsi vals jelas mengembalikan nilai integer. Dimana nilai yang akan dikeluarkan yaitu 3 dan 7.

Variable yang digunakan yaitu a,b dan c. fungsi vals dipanggil untuk dimasukkan kedalam variable a dan b. hasil yang dikeluarkan a adalah 3 dan b adalah 7

Variable c memanggil fungsi vals dimana hasil yang diinginkan yaitu nilai setelah koma yang berarti 7.

2. COMMAND LINE

```
[mazayaazd@localhost gp_projects]$ $GOBIN/soaltujuh
word: foo
numb: 42
fork: false
svar: bar
tail: []
[mazayaazd@localhost gp_projects]$
```

Melakukan pengecekan disetiap perintah barisnya. Dimana isi dari fungsi tersebut terdapat variable wordPtr, numbPtr dan boolPtr, svar.

Hal pertama yang dilakukan variable variable tersebut akan diisi dengan sebuah nilai. Nantinya ketika fungsi tersebut dipanggil

maka fungsi akan mengeluarkan hasil sesuai dengan perintah perbarisnya. Atau kata lain hasil yang dikeluarkan sesuai pemanggilannya.

Nama: Mazaya Z D	NIM: 1301154508	Nilai:
------------------	-----------------	--------

Soal No 6 (simple web application)

```
package main

import (
    "fmt"
    "net/http"
)

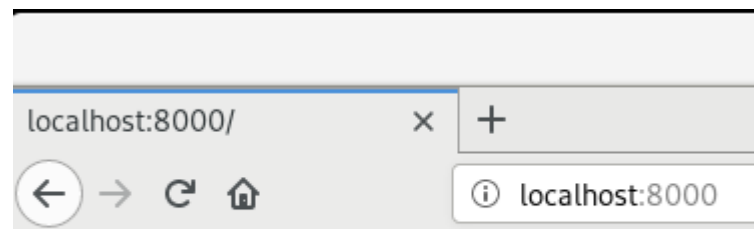
func main() {
    http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
        fmt.Fprintf(w, "Hello, you've requested: %s\n", r.URL.Path)
    })

    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:



Hello, you've requested: /

Pertama kali yang akan kita lakukan yaitu membuat koneksi dengan database yang ingin ditampilkan. Setelah koneksi berhasil terbentuk, diperlukan localhost untuk menjalankan fungsi sebagai server lokal agar dapat mengakses dan mengelola database yang ingin ditampilkan. Setelah itu http handle akan menunggu response permintaan yang telah dikirimkan sebelumnya agar teks dapat ditampilkan.

Sebelum program dijalankan port akan diarahkan ke 8000 (dapat berselancar secara sembunyi-sembunyi). Port akan mengidentifikasi sebuah proses di mana sebuah server dapat memberikan sebuah layanan kepada klien atau bagaimana sebuah klien dapat mengakses sebuah layanan yang ada dalam server. Sehingga port akan mengizinkan sebuah komputer untuk mendukung beberapa sesi koneksi.

Nama: Mazaya Z D	NIM: 1301154508	Nilai:
------------------	-----------------	--------