|  |  |  |  |
| --- | --- | --- | --- |
| C:\Users\Nabila Elshanum\Downloads\kad.jpg | **Tugas 1**  **Pemrograman Jaringan (CSH4V3)**  **------------------------------**  **Semester Genap 2018 - 2019**  **Dosen: Aulia Arif Wardana, S.Kom., M.T. (UIW)** | | |
| ***Berdo’alah sebelum mengerjakan. Dilarang berbuat curang.***  ***Tugas ini untuk mengukur kemampuan anda, jadi kerjakan dengan sepenuh hati.***  ***Selamat belajar, semoga sukses !*** | | | |
| **Nama Mahasiswa:**  **Taufik Rahmat Kurniawan** | | **NIM:**  **1301178601** | **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** |
| http://telescript.denayer.wenk.be/~hcr/cn/idoceo/images/tcpfsm.gif  Jelaskan maksud diagram finite state machine dari TCP Connection diatas! |
| Jawaban:  CLOSED Tidak ada koneksi  LISTEN Server menunggu call dari client  SYN-SENT Request koneksi dikirim, menunggu ack  STN-RCVD Request koneksi diterima  ESTABLISHED Koneksi terjalin  FIN-WAIT-1 Aplikasi menginginkan penutupan koneksi  FIN-WAIT-2 Sisi lain menerima penutupan koneksi  CLOSING Kedua sisi memutuskan untuk menutup koneksi bersama-sama  TIME-WAIT Menunggu pengiriman ulang segmen2 sampai selesai  CLOSE-WAIT Server menunggu aplikasi ditutup  LAST-ACK Server menunggu ack terakhir |

|  |
| --- |
| **Soal No 2 (for dan if/else)** |
| Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya! |
| Jawaban:    package main  import "fmt"  func main() {      i := 1      for i <= 3 { // for yang pertama untuk tampilkan 1-3          fmt.Println(i)          i = i + 1      }      for j := 7; j <= 9; j++ { // for yang kedua untuk tampilkan 7-9          fmt.Println(j)      }      for { // for yang ketiga untuk tampilkan kata “loop”          fmt.Println("loop")          break      }      for n := 0; n <= 5; n++ { // for yang keempat untuk tampilkan bilangan ganjil 0-5          if n%2 == 0 {              continue          }          fmt.Println(n)      }  }    package main  import "fmt"  func main() {      if 7%2 == 0 { // If yang pertama menampilkan “7 is odd” karena 7 tidak habis dibagi 2          fmt.Println("7 is even")      } else {          fmt.Println("7 is odd")      }      if 8%4 == 0 { // if yang kedua menampilkan “8 is divisible by 4” karena 8 habis dibagi 2          fmt.Println("8 is divisible by 4")      }      if num := 9; num < 0 { // If yang ketiga menampilkan “9 has 1 digit” karena 9 kurang dari 10          fmt.Println(num, "is negative")      } else if num < 10 {          fmt.Println(num, "has 1 digit")      } else {          fmt.Println(num, "has multiple digits")      }  } |
| **Soal No 3 (array dan function)** |
| Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya! |
| Jawaban:    package main  import "fmt"  func main() {      var a [5]int      fmt.Println("emp:", a) // tampilkan emp denga nisi array 0-4 yakni 0      a[4] = 100      fmt.Println("set:", a) //tampilkan isi array a dengan array ke 4 diisi 100      fmt.Println("get:", a[4]) //tampilkan data array ke 4      fmt.Println("len:", len(a)) // tampilkan jumlahh array      b := [5]int{1, 2, 3, 4, 5}      fmt.Println("dcl:", b) // tampilkan array yang sudah di set isinya      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) //tampilkan I untuk baris dan j untuk kolom, serta jumlah kan tiap isi sesuai dengan baris + kolom  }    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) // tampilkan 1 + 2 = 3 (yang dipanggil dari fungsi)      res = plusPlus(1, 2, 3)      fmt.Println("1+2+3 =", res) // tampilkan 1 + 2 + 3 = 6 (yang dipanggil dari fungsi)  } |
| **Soal No 4 (struct dan method)** |
| Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya! |
| Jawaban:    package main  import "fmt"  type person struct {      name string      age int  }  func main() {      fmt.Println(person{"bob", 20}) // tampilkan bob 20      fmt.Println(person{name: "Alice", age: 30}) // tampilkan alice 30      fmt.Println(person{name: "Fred"}) // tampilkan fred 0 (int di set 0)      fmt.Println(&person{name: "Ann", age: 40}) // tampilkan ann 40      s := person{name: "Sean", age: 50}      fmt.Println(s.name) // tampilkan 50      sp := &s      fmt.Println(sp.age)// tampilkan nothing      sp.age = 51      fmt.Println(sp.age) // tampilkan 51  }    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()) // width \* height = 50 (area)      fmt.Println("perim:", r.perim()) // 2\*10+2\*5= 30 (perim)      rp := &r      fmt.Println("area: ", rp.area()) // width \* height = 50 (area)      fmt.Println("perim:", rp.perim()) // 2\*10+2\*5= 30 (perim)  } |
| **Soal No 5 (multiple return value dan command line)** |
| Jalankan masing-masing program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya! |
| Jawaban:    package main  import "fmt"  func vals() (int, int) {      return 3, 7  }  func main() {      a, b := vals()      fmt.Println(a) // tampilkan nilai a yakni 3      fmt.Println(b) // tampilkan nilai b yakni 7      \_, c := vals()      fmt.Println(c)// tampilkan nilai c yakni 7  }    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.Print("word:", \*wordPtr)      fmt.Print("numb:", \*numbPtr)      fmt.Print("fork:", \*boolPtr)      fmt.Print("svar:", svar)      fmt.Print("tail:", flag.Args())  } |
| **Soal No 6 (simple web application)** |
| 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:    package main  import (      "fmt"      "net/http"  )  func main() {      http.HandleFunc("/", func(w http.ResponseWriter, r \*http.Request) {          fmt.Fprint(w, "Hello, you've requested: %s\n", r.URL.Path)      })      http.ListenAndServe(":8000", nil)  } |
| **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: |