



Tugas 3
Pemrograman Jaringan (CSH4V3)

Semester Ganjil 2019 - 2020
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: Hilmi Triandi Nurachmad | NIM: 1301164286 | Nilai: |
| Nama Mahasiswa: Javiar Fasyah | NIM: 1301164477 | Nilai: |
| Nama Mahasiswa: Fahrur Rozi S | NIM: 1301164213 | 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. 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 kelompok 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.

10. Walaupun tugas berkelompok tapi pengumpulan link github harus individu, jika tidak mengumpulkan maka dianggap tidak mengerjakan.

| | | |
|------------------------|----------------|--------|
| Nama: hilmi triandi | NIM:1301164286 | Nilai: |
|------------------------|----------------|--------|

Soal No 1 (Host Lookup)

```

/* ResolveIP
*/

package main

import (
    "fmt"
    "net"
    "os"
)

func main() {
    if len(os.Args) != 2 {
        fmt.Fprintf(os.Stderr, "Usage: %s hostname\n", os.Args[0])
        fmt.Println("Usage: ", os.Args[0], "hostname")
        os.Exit(1)
    }
    name := os.Args[1]

    addr, err := net.ResolveIPAddr("ip", name)
    if err != nil {
        fmt.Println("Resolution error", err.Error())
        os.Exit(1)
    }

    fmt.Println("Resolved address is ", addr.String())
    os.Exit(0)
}

```

Jalankan program diatas (go run ResolveIP.go www.google.com), apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya menggunakan diagram FSM!

| | | |
|------------------------|----------------|--------|
| Nama: hilmi triandi | NIM:1301164286 | Nilai: |
|------------------------|----------------|--------|

Jawaban:

```
[hilmitn@localhost Tugas3]$ go run nomer1.go www.google.com  
Resolved address is 172.217.194.99
```

program ini mencari alamat melalui hostname yang dimasukan melalui argumen.
Dan akan menampilkan error apabila masukan tidak sesuai

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

Soal No 2 (Service Lookup)

```

/* LookupPort
*/

package main

import (
    "fmt"
    "net"
    "os"
)

func main() {
    if len(os.Args) != 3 {
        fmt.Fprintf(os.Stderr,
            "Usage: %s network-type service\n",
            os.Args[0])
        os.Exit(1)
    }
    networkType := os.Args[1]
    service := os.Args[2]

    port, err := net.LookupPort(networkType, service)
    if err != nil {
        fmt.Println("Error: ", err.Error())
        os.Exit(2)
    }

    fmt.Println("Service port ", port)
    os.Exit(0)
}

```

Jalankan program diatas (go run LookupPort.go tcp telnet), apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya menggunakan diagram FSM!

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

Jawaban:

```
[hilmitn@localhost Tugas3]$ go run nomer2.go tcp telnet
Service port 23
```

program ini melakukan lookupport dengan masukan networktype dan service sebagai masukan, kalau masukan sesuai program akan mengembalikan port yang sesuai juga.

Soal No 3 (TCP Client)

```
/* GetHeadInfo
*/
package main

import (
    "fmt"
    "io/ioutil"
    "net"
    "os"
)

func main() {
    if len(os.Args) != 2 {
        fmt.Fprintf(os.Stderr, "Usage: %s host:port ", os.Args[0])
        os.Exit(1)
    }
    service := os.Args[1]

    tcpAddr, err := net.ResolveTCPAddr("tcp4", service)
    checkError(err)

    conn, err := net.DialTCP("tcp", nil, tcpAddr)
    checkError(err)

    _, err = conn.Write([]byte("HEAD / HTTP/1.0\r\n\r\n"))
    checkError(err)

    result, err := ioutil.ReadAll(conn)
    checkError(err)
```

Tugas - Pemr
Halaman 6 d

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

```

        fmt.Println(string(result))

    os.Exit(0)
}

func checkError(err error) {
    if err != nil {
        fmt.Fprintf(os.Stderr, "Fatal error: %s", err.Error())
        os.Exit(1)
    }
}

```

Jalankan program diatas (go run GetHeadInfo.go http://www.google.com:80), apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya menggunakan diagram FSM!

Jawaban:

```

1  /* GetHeadInfo
2  */
3  package main
4
5  import (
6      "fmt"
7      "io/ioutil"
8      "net"
9      "os"
10 )
11
12 func main() {
13     if len(os.Args) != 2 {
14         fmt.Fprintf(os.Stderr, "Usage: %s host:port ", os.Args[0])
15         os.Exit(1)
16     }
17
18     url := os.Args[1]
19     resp, err := http.Head(url)
20     if err != nil {
21         fmt.Fprintf(os.Stderr, "Error: %s", err.Error())
22         os.Exit(1)
23     }
24     fmt.Println(string(resp.Header))
25 }

```

```

[javiar@localhost-live go]$ go run GetHeadInfo.go www.google.com:80
HTTP/1.1 200 OK
Date: Thu, 12 Sep 2019 13:27:21 GMT
Expires: 0
Cache-Control: private, max-age=0
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Server: gcs
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: IP=34W-2019-09-12-13; expires=Sat, 12-Oct-2019 13:27:21 GMT; path=/; domain=.google.com; SameSite=None
Set-Cookie: HB=183a8p2wH08z5a9jG1128v6S1ro6g2BaAM050cFbAc-IfmjF70dX0u8z2UdumH0rzCqPwYf1hw4450eV15m4P2NcluzXEp3Ewuu1WCS1R3rtu7Nau4hYegEhJynht5
B35Yx0D-90p1ym4e0v0h; expires=Fri, 13-Mar-2020 13:27:21 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding

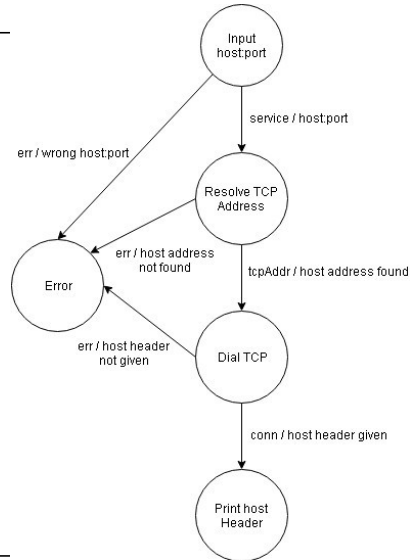
```

Program ini bekerja dengan menggunakan input argumen host serta port, ketika menjalankan programnya. Setelah itu host dari input akan dicari dan mengembalikan alamat dari host jika ditemukan. Alamat ini digunakan untuk memanggil host yang dimaksud. Lalu jika mendapat balasan maka program akan mencatat info header dari host dan mencetaknya sebagai hasil dari program

Nama:

NIM:

Nilai:



Soal No 4 (Raw Sockets and the IPConn Type)

```
/* Ping
*/
package main

import (
    "bytes"
    "fmt"
    "io"
    "net"
    "os"
)

// chan if err != nil {
const m      fmt.Println("Resolution error", err.Error())
const i      os.Exit(1)
}

func main() {
    remoteAddr, err := net.ResolveIPAddr("ip4", os.Args[1])
    if err != nil {
        fmt.Println("Resolution error", err.Error())
        os.Exit(1)
    }

    conn, err := net.DialIP("ip4:icmp", localAddr, remoteAddr)
    checkError(err)

    var msg [512]byte
    msg[0] = 8 // echo
    msg[1] = 0 // code 0
    msg[2] = 0 // checksum, fix later
    msg[3] = 0 // checksum, fix later
    msg[4] = 0 // identifier[0]
    msg[5] = 13 // identifier[1] (arbitrary)
    msg[6] = 0 // sequence[0]
    msg[7] = 37 // sequence[1] (arbitrary)
    len := 8

    // now fix checksum bytes
    check := checksum(msg[0:len])
    msg[2] = byte(check >> 8)
    msg[3] = byte(check & 255)
}
```


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

```

// send the message
_, err = conn.Write(msg[0:len])
checkError(err)

fmt.Print("Message sent:  ")
for n := 0; n < 8; n++ {
    fmt.Print(" ", msg[n])
}
fmt.Println()

// receive a reply
size, err2 := conn.Read(msg[0:])
checkError(err2)

fmt.Print("Message received:")
for n := ipv4HeaderSize; n < size; n++ {
    fmt.Print(" ", msg[n])
}
fmt.Println()
os.Exit(0)
}
func checkSum(msg []byte) uint16 {
    sum := 0

    // assume even for now
    for n := 0; n < len(msg); n += 2 {
        sum += int(msg[n])*256 + int(msg[n+1])
    }
    sum = (sum >> 16) + (sum & 0xffff)
    sum += (sum >> 16)
    var answer uint16 = uint16(^sum)
    return answer
}

func checkError(err error) {
    if err != nil {
        fmt.Fprintf(os.Stderr, "Fatal error: %s", err.Error())
        os.Exit(1)
    }
}

```

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

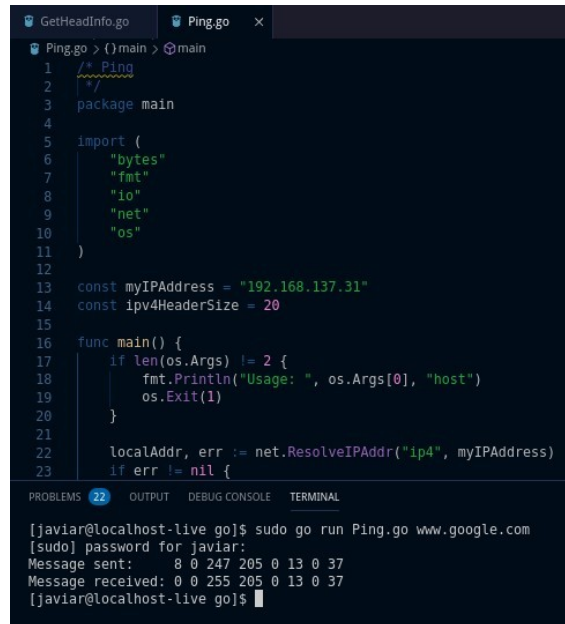
```
func readFully(conn net.Conn) ([]byte, error) {
    defer conn.Close()

    result := bytes.NewBuffer(nil)
    var buf [512]byte
    for {
        n, err := conn.Read(buf[0:])
        result.Write(buf[0:n])
        if err != nil {
            if err == io.EOF {
                break
            }
            return nil, err
        }
    }
    return result.Bytes(), nil
}
```

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

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

Jawaban:



```

1  /* Ping
2  */
3  package main
4
5  import (
6      "bytes"
7      "fmt"
8      "io"
9      "net"
10     "os"
11 )
12
13 const myIPAddress = "192.168.137.31"
14 const ipv4HeaderSize = 20
15
16 func main() {
17     if len(os.Args) != 2 {
18         fmt.Println("Usage: ", os.Args[0], "host")
19         os.Exit(1)
20     }
21
22     localAddr, err := net.ResolveIPAddr("ip4", myIPAddress)
23     if err != nil {

```

```

[javiar@localhost-live go]$ sudo go run Ping.go www.google.com
[sudo] password for javiar:
Message sent:      8 0 247 205 0 13 0 37
Message received: 0 0 255 205 0 13 0 37
[javiar@localhost-live go]$

```

pengguna mengirimkan ping kepada sebuah host berupa pesan byte delapan index array disertai dengan checksum nya pada index kedua dan ketiga. Awalnya adalah nol. Setelah dihitung check sum dari pesan, dimasukan nilai checksum ke index 2 dan 3 kemudian dikirimkan host tujuan.

Soal No 5 (Multi-Threaded Server)

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

```

package main

import (
    "bufio"
    "fmt"
    "net"
)

func check(err error, message string) {
    if err != nil {
        panic(err)
    }
    fmt.Printf("%s\n", message)
}

func main() {
    ln, err := net.Listen("tcp", ":8080")
    check(err, "Server is ready.")

    for {
        conn, err := ln.Accept()
        check(err, "Accepted connection.")

        go func() {
            buf := bufio.NewReader(conn)

            for {
                name, err := buf.ReadString('\n')

                if err != nil {
                    fmt.Printf("Client disconnected.\n")
                    break
                }

                conn.Write([]byte("Hello, " + name))
            }
        }()
    }
}

```

Jalankan program diatas di dalam visual box yang sudah anda buat, kemudian lakukan telnet ke port 8080 dalam jumlah yang banyak secara bersamaan, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

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

Jawaban:

```

package main

import (
    "bufio"
    "fmt"
    "net"
    "time"
)

func check(err error, message string) {
    if err != nil {
        panic(err)
    }
    fmt.Printf("%s\n", message)
}

type ClientJob struct {
    name string
    conn net.Conn
}

func generateResponses(clientJobs chan ClientJob) {
    for {
        // Wait for the next job to come off the queue.
        clientJob := <-clientJobs

        // Do something thats keeps the CPU busy for a whole second.
        for start := time.Now(); time.Now().Sub(start) < time.Second; {
        }

        // Send back the response.
        clientJob.conn.Write([]byte("Hello, " + clientJob.name))
    }
}

func main() {
    clientJobs := make(chan ClientJob)
    go generateResponses(clientJobs)

    ln, err := net.Listen("tcp", ":8080")
    check(err, "Server is ready.")

    for {
        conn, err := ln.Accept()
        check(err, "Accepted connection.")

        go func() {
            buf := bufio.NewReader(conn)

            for {
                name, err := buf.ReadString('\n')

                if err != nil {
                    fmt.Printf("Client disconnected.\n")
                    break
                }

                clientJobs <- ClientJob{name, conn}
            }
        }()
    }
}

```

Tug
Hal

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

| |
|--|
| <p>Jalankan program diatas di dalam virtual box yang sudah anda buat, kemudian lakukan telnet ke port 8080 dalam jumlah yang banyak secara bersamaan, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!</p> |
|--|

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

Jawaban: