

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:	NIM:	Nilai:
YUWANTORO MUKHLISIN	1301150042
Nama Mahasiswa:	NIM:	Nilai:
LUKMAN BUDIMAN	1301164725
Nama Mahasiswa:	NIM:	Nilai:
SALMA FAUZIA SUSAN	1301164442

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 disetiap 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:	NIM:	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!

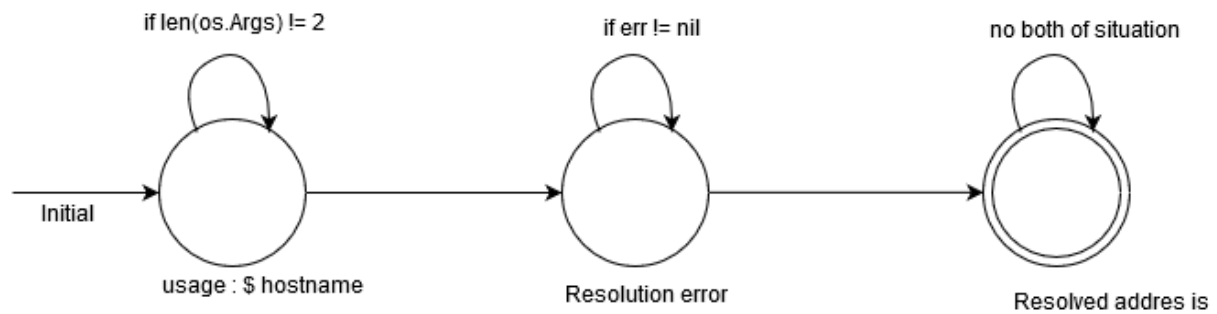
Nama:	NIM:	Nilai:
-------	------	--------

Jawaban:

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
t: bash
[luckman@localhost Tugas3]$ go run ResolveIP.go www.google.com
Resolved address is 216.239.38.120
[luckman@localhost Tugas3]$

```



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]

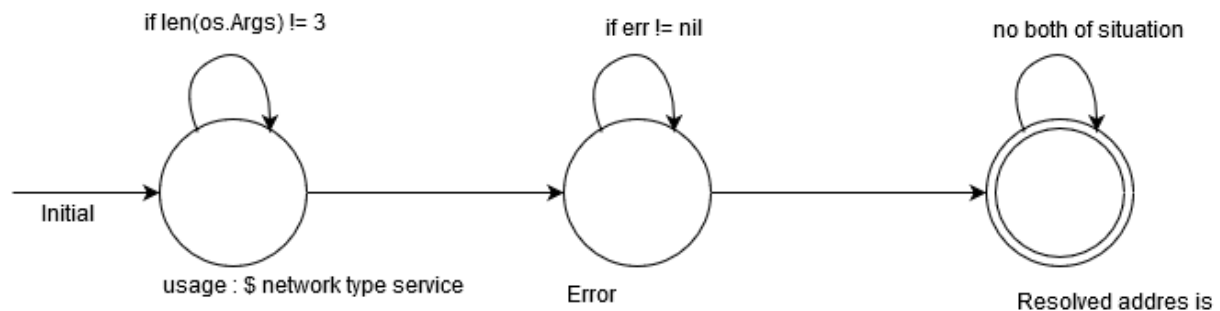
    port, err := net.LookupPort(networkType, service)
    if err != nil {
        fmt.Println("Error: ", err.Error())
        os.Exit(2)
    }

    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!

Jawaban:



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

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)

    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)
    }
}

```

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

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

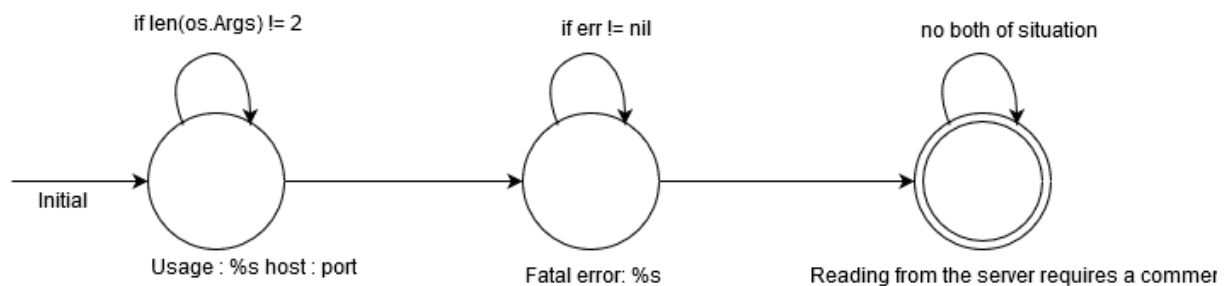
Jawaban:

```

[luckman@localhost Tugas3]$ go run GetHeadInfo.go http://www.google.com:80
Fatal error: address http://www.google.com:80: too many colons in address
[luckman@localhost Tugas3]$

[luckman@localhost Tugas3]$ go run GetHeadInfo.go www.google.com:80
HTTP/1.0 200 OK
Date: Thu, 12 Sep 2019 20:51:12 GMT
Expires: -1
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: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Set-Cookie: 1P_JAR=2019-09-12-20; expires=Sat, 12-Oct-2019 20:51:12 GMT; path=/; domain=.google.com; SameSite=none
Set-Cookie: NID=188=AQo0YVv4cdisJvEIyrIqLrg99DgtYqkl2GrGuoLL-VYlkd0FRBxI2hZuyDs0fV59xPjDQ t 2KbmfeCpmJvzf-5vvGLVTdAFyidL01fup1LrgIY0n8GefxAnI35PaCwP0vQtNgePm8fALNqhya8krERiXwsF8vl0dxDduIH8MFY; expires=Fri, 13-Mar-2020 20:51:12 GMT; path=/; domain=.google.com; HttpOnly
Accept-Ranges: none
Vary: Accept-Encoding
[luckman@localhost Tugas3]$

```



Soal No 4 (Raw Sockets and the IPConn Type)

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

```

/* Ping
*/
package main

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

// change this to my own IP address or set to 0.0.0.0
const myIPAddress = "192.168.1.2"
const ipv4HeaderSize = 20

func main() {
    if len(os.Args) != 2 {
        fmt.Println("Usage: ", os.Args[0], "host")
        os.Exit(1)
    }

    localAddr, err := net.ResolveIPAddr("ip4", myIPAddress)

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

    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)
    }
}

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
}

```

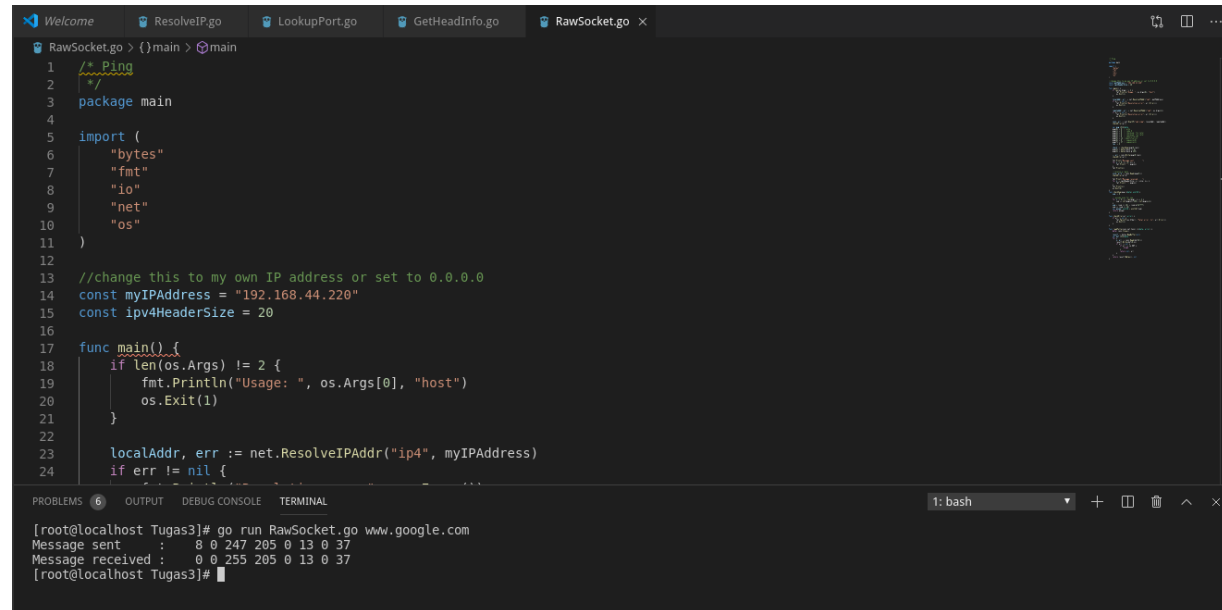
Jalankan program diatas, apakah outputnya (berikan printscreen) dan jelaskan cara kerjanya!

Nama:

NIM:

Nilai:

Jawaban:



```
RawSocket.go > {}main > @main
1  /*_Ping
2  */
3  package main
4
5  import (
6      "bytes"
7      "fmt"
8      "io"
9      "net"
10     "os"
11 )
12
13 //change this to my own IP address or set to 0.0.0.0
14 const myIPAddress = "192.168.44.220"
15 const ipv4HeaderSize = 20
16
17 func main() {
18     if len(os.Args) != 2 {
19         fmt.Println("Usage: ", os.Args[0], "host")
20         os.Exit(1)
21     }
22
23     localAddr, err := net.ResolveIPAddr("ip4", myIPAddress)
24     if err != nil {
25         // ...
26     }
27 }
28
29 [root@localhost Tugas3]# go run RawSocket.go www.google.com
Message sent      :  8 0 247 205 0 13 0 37
Message received  :  0 0 255 205 0 13 0 37
[root@localhost Tugas3]#
```

Cara kerja :

cara mengirim pesan ping ke host. Ping menggunakan perintah "echo" dari protokol ICMP. Ini adalah protokol yang berorientasi byte, di mana klien mengirimkan aliran byte ke host lain, dan host membalas. formatnya adalah:

- Byte pertama adalah 8, untuk pesan echo
- Byte kedua adalah nol
- Byte ketiga dan keempat adalah checksum pada seluruh pesan
- Byte kelima dan keenam adalah pengidentifikasi arbitrer
- Ketujuh dan delapan byte adalah nomor urut sembarang
- Sisa paket adalah data pengguna

Program diatas akan menyiapkan koneksi IP, mengirim permintaan ping ke host dan mendapatkan balasan. Untuk mengaksesnya perlu memiliki akses root untuk menjalankannya dengan sukses

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
                }

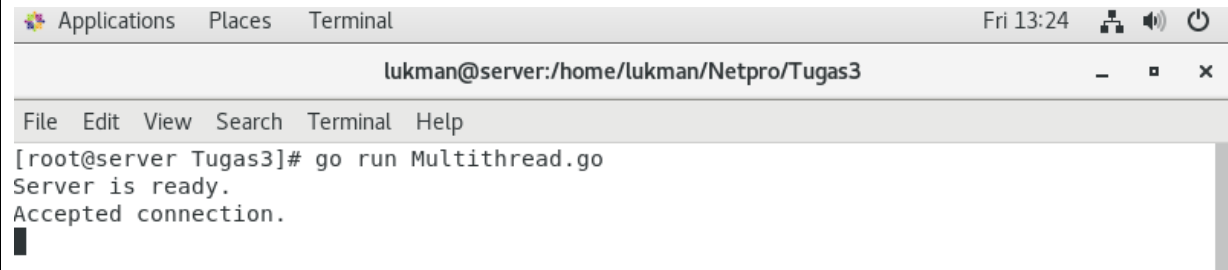
                conn.Write([]byte("Hello, " + name))
            }
        }()
    }
}

```

Jalankan program diatas di dalam 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!

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

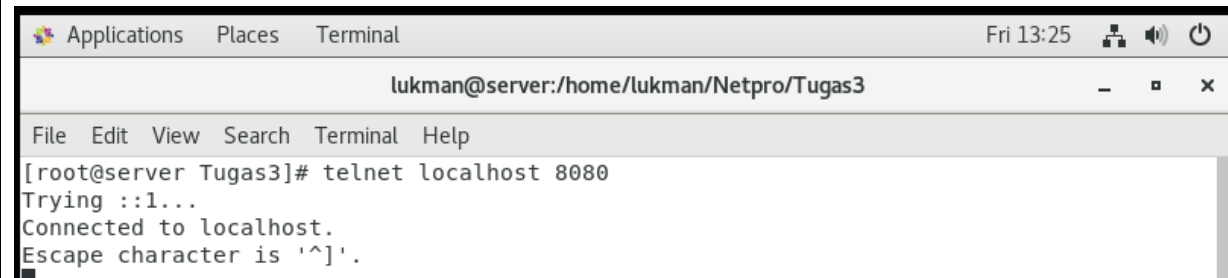
Jawaban:



```

Applications  Places  Terminal  Fri 13:24
lukman@server:/home/lukman/Netpro/Tugas3
File Edit View Search Terminal Help
[root@server Tugas3]# go run Multithread.go
Server is ready.
Accepted connection.

```



```

Applications  Places  Terminal  Fri 13:25
lukman@server:/home/lukman/Netpro/Tugas3
File Edit View Search Terminal Help
[root@server Tugas3]# telnet localhost 8080
Trying ::1...
Connected to localhost.
Escape character is '^]'.

```

Cara kerjanya yaitu server menunggu client yang terhubung , ketika client terhubung maka server mereply dengan accepted connection.

Soal No 6 (Multi-Threaded Server)

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

```

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}
            }
        }()
    }
}

```

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!

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

Jawaban:

Cara kerjanya yaitu server menunggu client yang terhubung , ketika client terhubung maka server mereply dengan accepted connection. Multi Threaded juga adalah proses dengan thread yang banyak dan mengerjakan lebih dari satu tugas dalam satu waktu.