

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

<b>Nama Mahasiswa:</b>	<b>NIM:</b>	<b>Nilai:</b>
<b>Maulana Azizwara</b>	<b>1301160210</b>	.....
<b>Nama Mahasiswa:</b>	<b>NIM:</b>	<b>Nilai:</b>
<b>Dianita Yustyka Sari</b>	<b>1301164349</b>	.....
<b>Nama Mahasiswa:</b>	<b>NIM:</b>	<b>Nilai:</b>
<b>Fanny Risca Wijayanti</b>	<b>1301164237</b>	.....

**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:
-------	------	--------

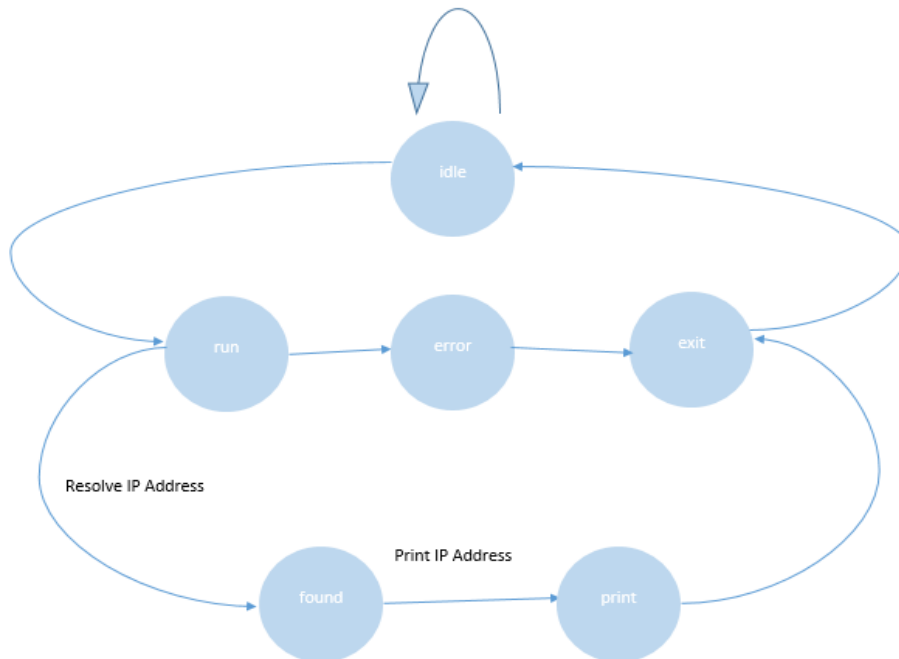
Jawaban:

```

root@localhost:/home/fannyrisca/Documents

[fannyrisca@localhost ~]$ sudo su
[sudo] password for fannyrisca:
[root@localhost fannyrisca]# cd Documents
[root@localhost Documents]# ls
no1.go no2.go no3.go tugas1
[root@localhost Documents]# go run no1.go www.google.com
Resolved address is 216.239.38.120
[root@localhost Documents]#

```



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]

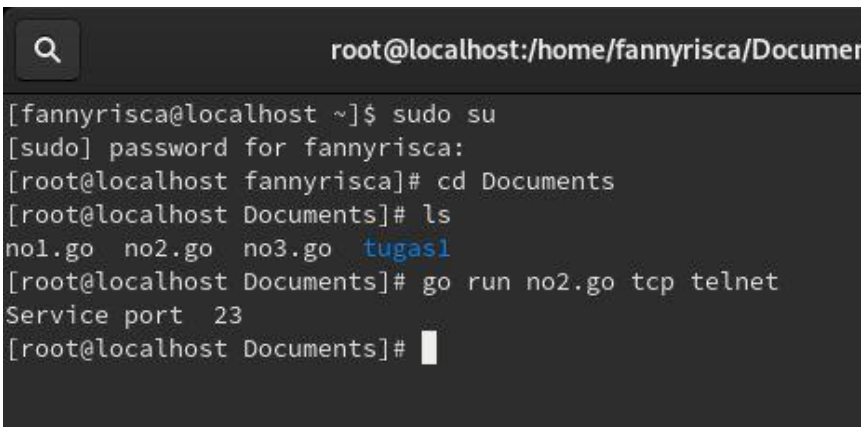
    port, err := net.LookupPort(networkType, service)
    if err != nil {
        fmt.Println("Error: ", err.Error())
        os.Exit(2)
    }

    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:

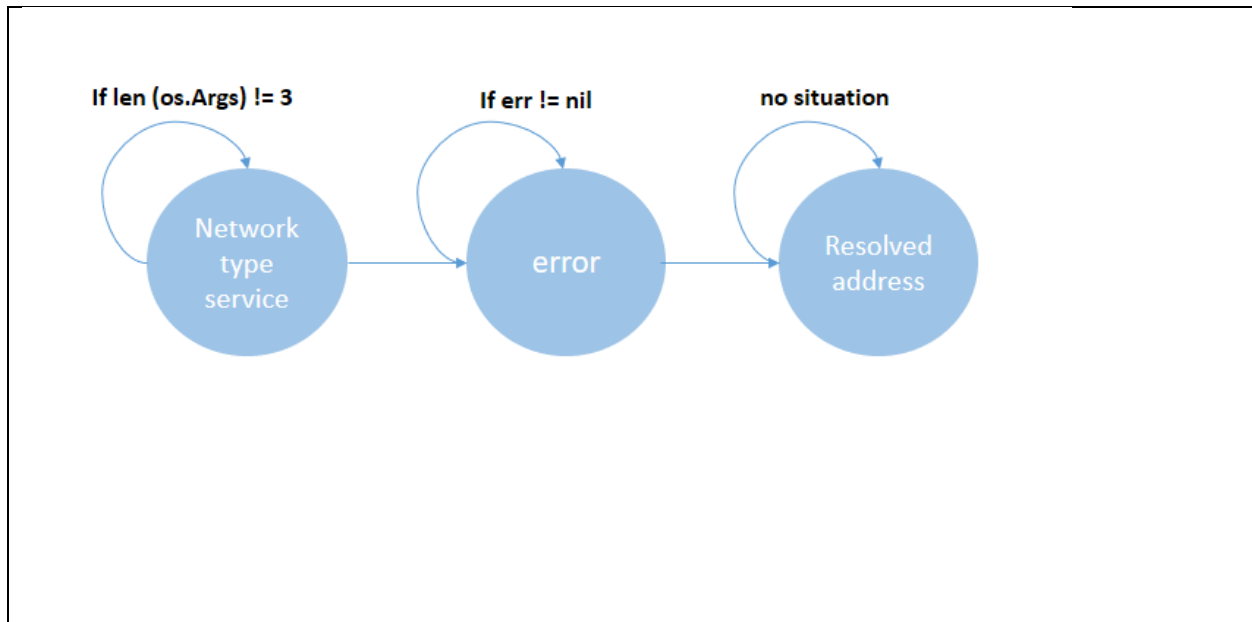


```

root@localhost:/home/fannyrisca/Documents
[fannyrisca@localhost ~]$ sudo su
[sudo] password for fannyrisca:
[root@localhost fannyrisca]# cd Documents
[root@localhost Documents]# ls
no1.go no2.go no3.go tugas1
[root@localhost Documents]# go run no2.go tcp telnet
Service port 23
[root@localhost Documents]#

```

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)
}

```

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

```

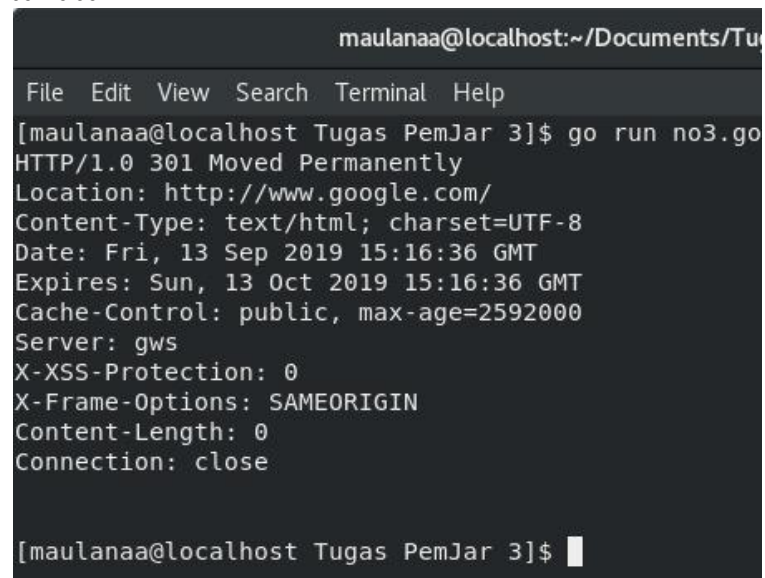
        fmt.Println(string(result))
    }
    os.Exit(0)
}

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:



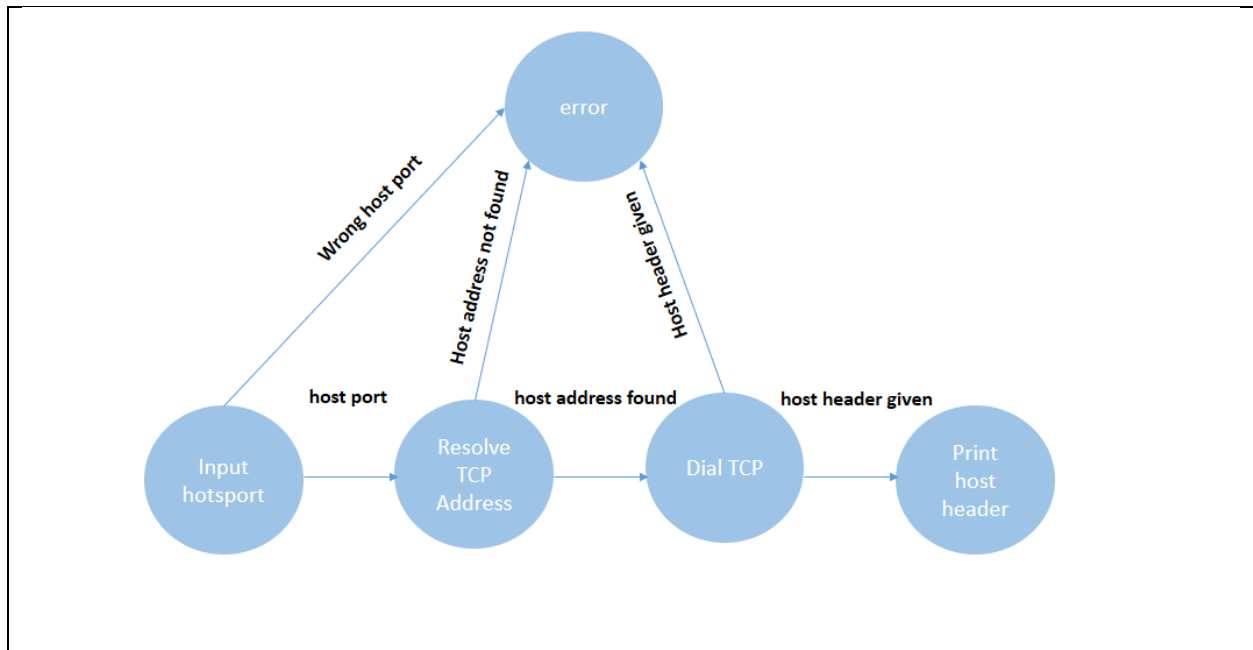
```

maulanaa@localhost:~/Documents/Tu
File Edit View Search Terminal Help
[maulanaa@localhost Tugas PemJar 3]$ go run no3.go
HTTP/1.0 301 Moved Permanently
Location: http://www.google.com/
Content-Type: text/html; charset=UTF-8
Date: Fri, 13 Sep 2019 15:16:36 GMT
Expires: Sun, 13 Oct 2019 15:16:36 GMT
Cache-Control: public, max-age=2592000
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Content-Length: 0
Connection: close

[maulanaa@localhost Tugas PemJar 3]$

```

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



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

#### Soal No 4 (Raw Sockets and the IPConn Type)

```

/* 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()

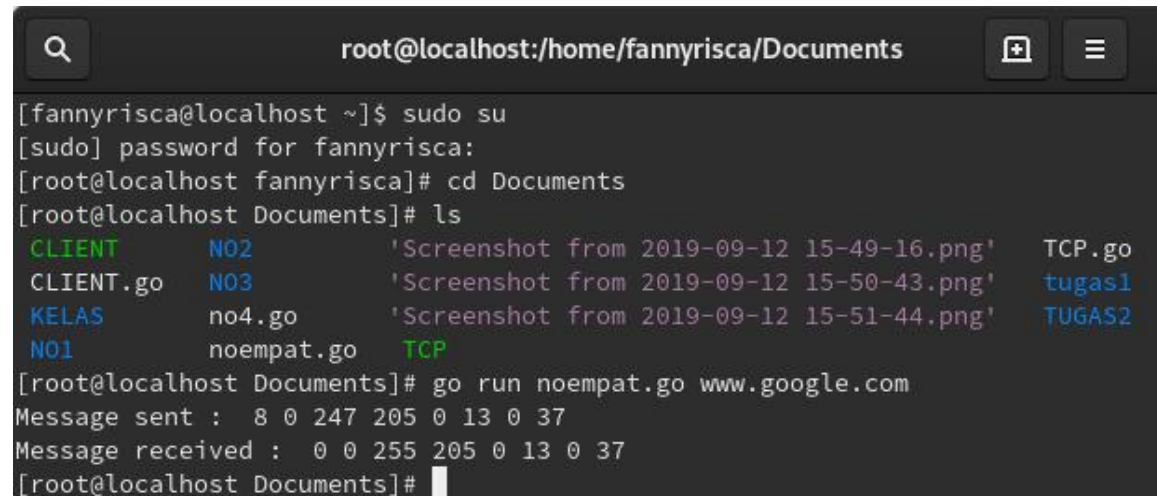
        result := bytes.NewBuffer(nil)
        var buf [512]byte
        for {
            n, err := conn.Read(buf[0:])
            result.Write(buf[0:n])
            if err != nil {
                if err == io.EOF {
                    break
                }
            }
            return nil, err
    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

```

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

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

Jawaban:



```

root@localhost:/home/fannyrisca/Documents

[fannyrisca@localhost ~]$ sudo su
[sudo] password for fannyrisca:
[root@localhost fannyrisca]# cd Documents
[root@localhost Documents]# ls
CLIENT      NO2          'Screenshot from 2019-09-12 15-49-16.png'  TCP.go
CLIENT.go   NO3          'Screenshot from 2019-09-12 15-50-43.png'  tugas1
KELAS       no4.go       'Screenshot from 2019-09-12 15-51-44.png'  TUGAS2
NO1         noempat.go   TCP

[root@localhost Documents]# go run noempat.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 Documents]#

```

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

### Soal No 5 (Multi-Threaded Server)

```

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:

CLIENT

```

maulanaa@localhost:~
File Edit View Search Terminal Tabs Help

root@192:~/NetPro3 x maulanaa@localhost:~ x

[maulanaa@localhost ~]$ telnet 192.168.1.19 8080
Trying 192.168.1.19...
Connected to 192.168.1.19.
Escape character is '^]'.
Maulana Azizwara
Hello, Maulana Azizwara

```

SERVER

```

root@192:~/NetPro3
File Edit View Search Terminal Tabs Help

root@192:~/NetPro3 x maulanaa@localhost:~ x

[root@192 NetPro3]# ls
no1.go no2.go no3.go no4.go no5.go no6.go README.md
[root@192 NetPro3]# go run no5.go
Server is ready
Accepted connection

```

1. Program membuat server pada port 8080 dengan menggunakan protokol TCP. Lalu, program dicek apakah terjadi error atau tidak. Jika tidak error, program menampilkan 'Server is ready' yang mana artinya server sudah berjalan dan sedang menunggu client.
2. Jika client yang melakukan telnet, program akan mengecek apakah ada error atau tidak. Jika tidak ada error, program akan menampilkan 'Accepted connection' pada server yang berarti telnet yang dilakukan client ke server berhasil.
3. Client yang berhasil melakukan telnet ke server dapat mengirim pesan ke server. Jika client mengirim pesan, server akan mengeceknya terlebih dahulu apakah ada error, jika tidak ada error pesan tersebut akan ditampilkan di client.

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

### Soal No 6 (Multi-Threaded Server)

```

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}
            }
        }()
    }
}

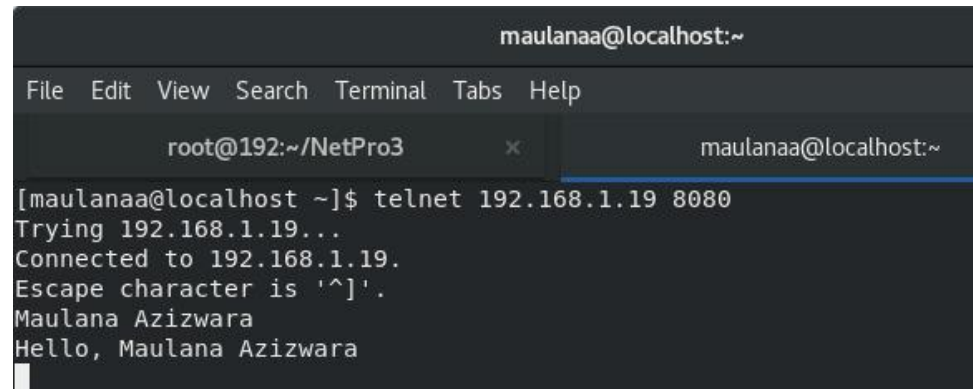
```

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

an 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!

Jawaban:

CLIENT

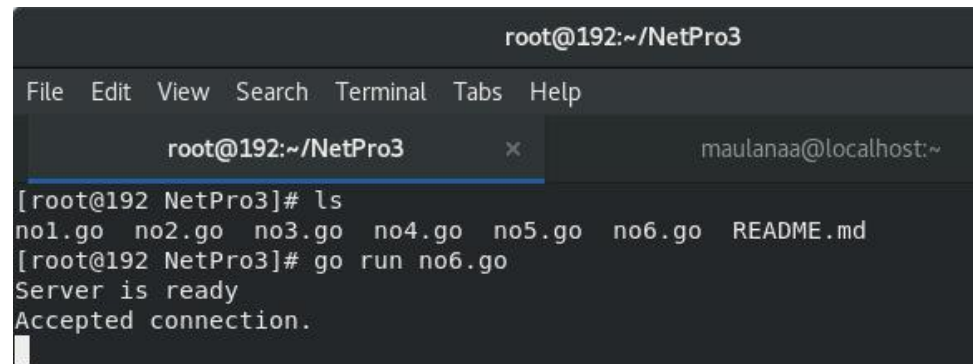


```

maulanaa@localhost:~
File Edit View Search Terminal Tabs Help
root@192:~/NetPro3 x maulanaa@localhost:~
[maulanaa@localhost ~]$ telnet 192.168.1.19 8080
Trying 192.168.1.19...
Connected to 192.168.1.19.
Escape character is '^]'.
Maulana Azizwara
Hello, Maulana Azizwara

```

SERVER



```

root@192:~/NetPro3
File Edit View Search Terminal Tabs Help
root@192:~/NetPro3 x maulanaa@localhost:~
[root@192 NetPro3]# ls
no1.go no2.go no3.go no4.go no5.go no6.go README.md
[root@192 NetPro3]# go run no6.go
Server is ready
Accepted connection.

```

Penjelasan untuk no soal 6 akan sama seperti no 5, namun pada nomor 6 menggunakan channel untuk generate response yang mana pengiriman dan penerimaan pesan terjadi secara synchronous.