

Nama : Nanda Safira Ihzanti

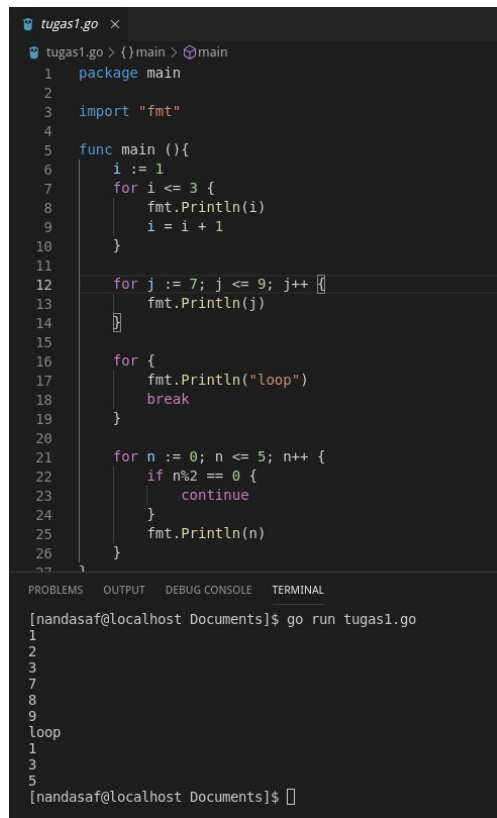
NIM : 1301164322

Pemrograman Jaringan

1. Diagram TCP Finite State Machine memperlihatkan proses terjadinya Three-way Handshake ketika client dan server melakukan pembuatan koneksi TCP.

No 2.

a. Melakukan perulangan sesuai dengan nilai yang telah ditentukan

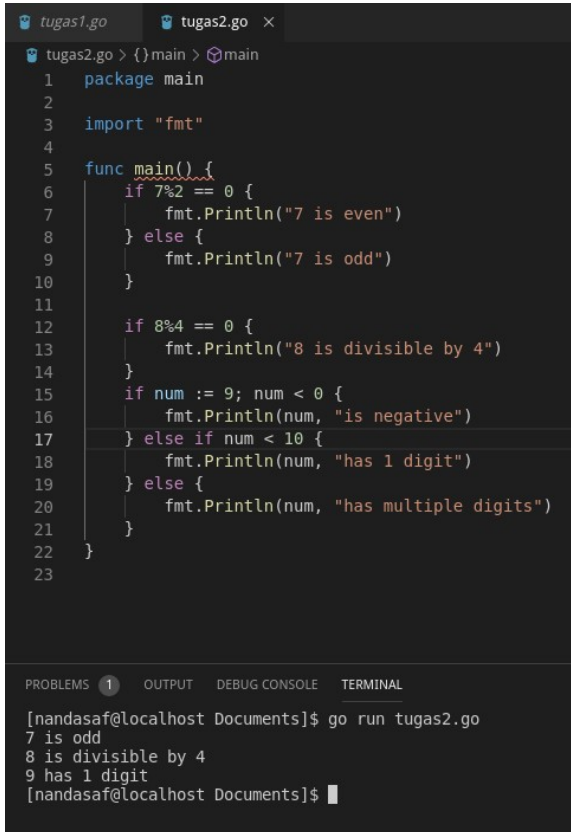


```
tugas1.go x
tugas1.go > {}main > main
1 package main
2
3 import "fmt"
4
5 func main(){
6     i := 1
7     for i <= 3 {
8         fmt.Println(i)
9         i = i + 1
10    }
11
12    for j := 7; j <= 9; j++ {
13        fmt.Println(j)
14    }
15
16    for {
17        fmt.Println("loop")
18        break
19    }
20
21    for n := 0; n <= 5; n++ {
22        if n%2 == 0 {
23            continue
24        }
25        fmt.Println(n)
26    }
27 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
[nandasaf@localhost Documents]$ go run tugas1.go
1
2
3
7
8
9
loop
1
3
5
[nandasaf@localhost Documents]$
```

b. Perbandingan yang dilakukan sesuai nilai yang telah ditentukan



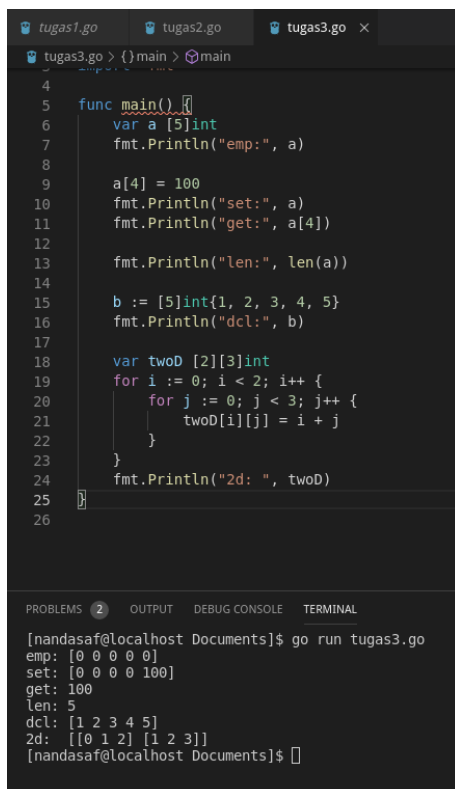
```
tugas2.go > {} main > main
1 package main
2
3 import "fmt"
4
5 func main(){
6     if 7%2 == 0 {
7         fmt.Println("7 is even")
8     } else {
9         fmt.Println("7 is odd")
10    }
11
12    if 8%4 == 0 {
13        fmt.Println("8 is divisible by 4")
14    }
15    if num := 9; num < 0 {
16        fmt.Println(num, "is negative")
17    } else if num < 10 {
18        fmt.Println(num, "has 1 digit")
19    } else {
20        fmt.Println(num, "has multiple digits")
21    }
22 }
23
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

```
[nandasaf@localhost Documents]$ go run tugas2.go
7 is odd
8 is divisible by 4
9 has 1 digit
[nandasaf@localhost Documents]$
```

No 3.

a.

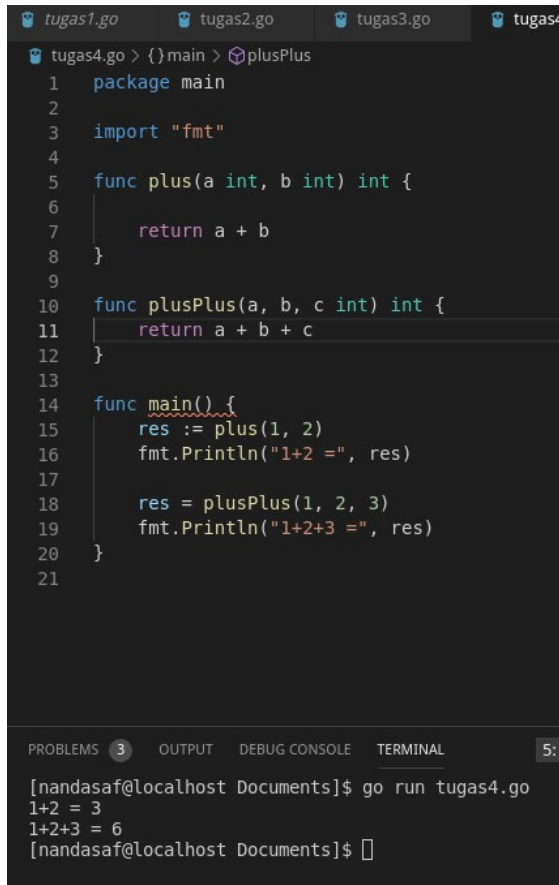


```
tugas3.go > {} main > main
4
5 func main(){
6     var a [5]int
7     fmt.Println("emp:", a)
8
9     a[4] = 100
10    fmt.Println("set:", a)
11    fmt.Println("get:", a[4])
12
13    fmt.Println("len:", len(a))
14
15    b := [5]int{1, 2, 3, 4, 5}
16    fmt.Println("dcl:", b)
17
18    var twoD [2][3]int
19    for i := 0; i < 2; i++ {
20        for j := 0; j < 3; j++ {
21            twoD[i][j] = i + j
22        }
23    }
24    fmt.Println("2d: ", twoD)
25 }
26
```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL

```
[nandasaf@localhost Documents]$ go run tugas3.go
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]]
[nandasaf@localhost Documents]$
```

b.



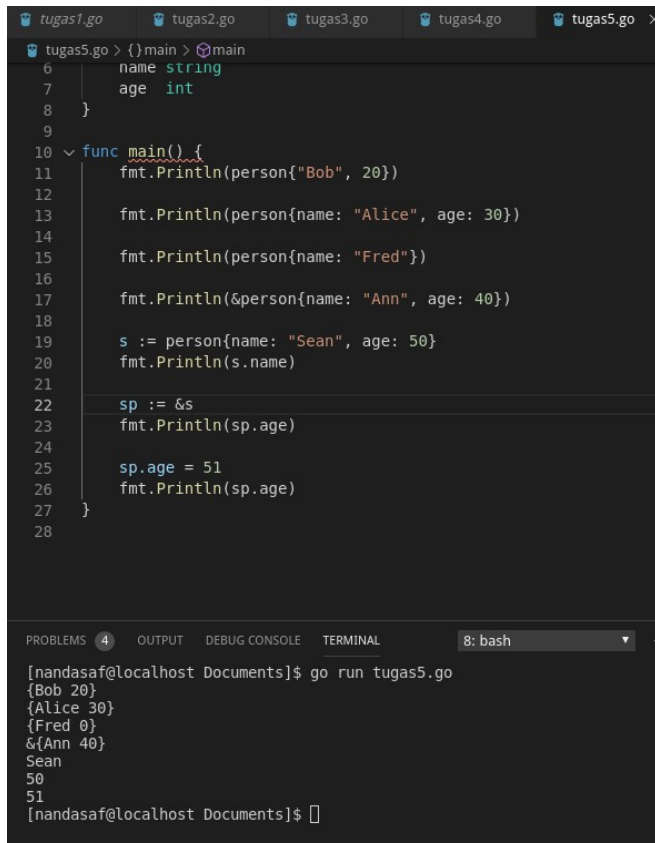
```
tugas4.go > {}main > plusPlus
1 package main
2
3 import "fmt"
4
5 func plus(a int, b int) int {
6     return a + b
7 }
8
9
10 func plusPlus(a, b, c int) int {
11     return a + b + c
12 }
13
14 func main() {
15     res := plus(1, 2)
16     fmt.Println("1+2 =", res)
17
18     res = plusPlus(1, 2, 3)
19     fmt.Println("1+2+3 =", res)
20 }
21
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL 5:

```
[nandasaf@localhost Documents]$ go run tugas4.go
1+2 = 3
1+2+3 = 6
[nandasaf@localhost Documents]$
```

No. 4

a. Terdapat tipe data name dan age yang di assign untuk memberikan nilai parameter tipe data tersebut

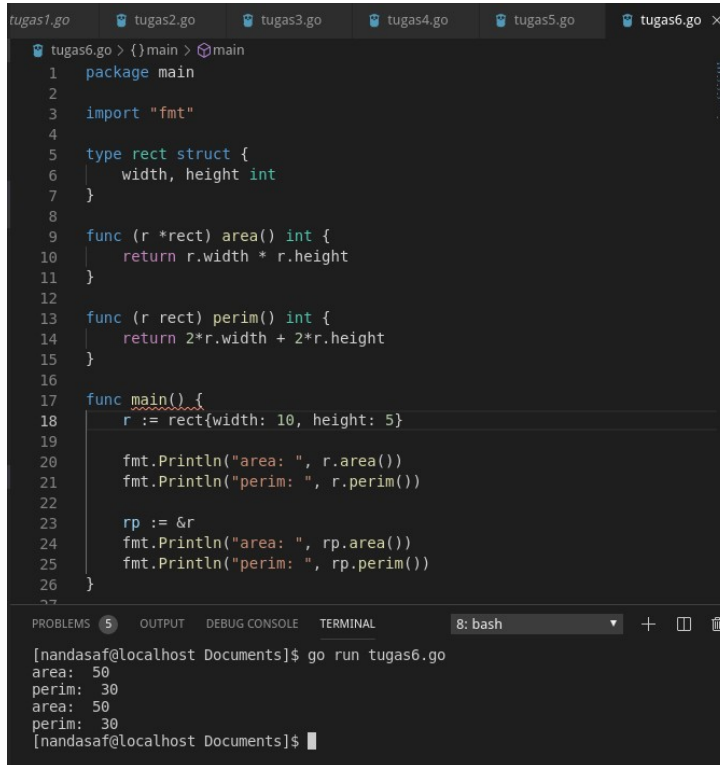


```
tugas5.go > {}main > main
6     name string
7     age  int
8 }
9
10 func main() {
11     fmt.Println(person{"Bob", 20})
12
13     fmt.Println(person{name: "Alice", age: 30})
14
15     fmt.Println(person{name: "Fred"})
16
17     fmt.Println(&person{name: "Ann", age: 40})
18
19     s := person{name: "Sean", age: 50}
20     fmt.Println(s.name)
21
22     sp := &s
23     fmt.Println(sp.age)
24
25     sp.age = 51
26     fmt.Println(sp.age)
27 }
28
```

PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL 8: bash

```
[nandasaf@localhost Documents]$ go run tugas5.go
{Bob 20}
{Alice 30}
{Fred 0}
&{Ann 40}
Sean
50
51
[nandasaf@localhost Documents]$
```

b. Metode yang digunakan ketika struct sudah diinisialisasi pada variabel



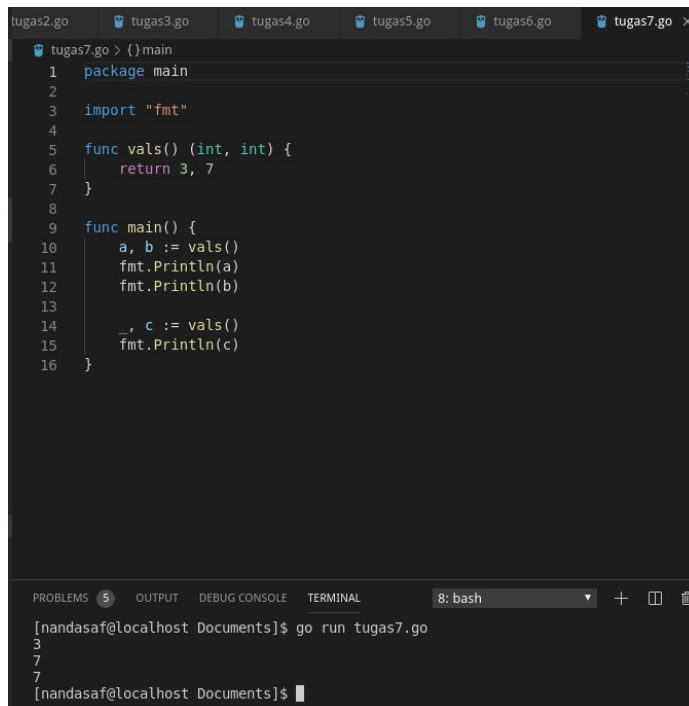
```
tugas6.go > {}main > main
1 package main
2
3 import "fmt"
4
5 type rect struct {
6     width, height int
7 }
8
9 func (r *rect) area() int {
10     return r.width * r.height
11 }
12
13 func (r rect) perim() int {
14     return 2*r.width + 2*r.height
15 }
16
17 func main() {
18     r := rect{width: 10, height: 5}
19
20     fmt.Println("area: ", r.area())
21     fmt.Println("perim: ", r.perim())
22
23     rp := &r
24     fmt.Println("area: ", rp.area())
25     fmt.Println("perim: ", rp.perim())
26 }
27
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL 8: bash

```
[nandasaf@localhost Documents]$ go run tugas6.go
area: 50
perim: 30
area: 50
perim: 30
[nandasaf@localhost Documents]$
```

No 5.

a.



```
tugas2.go tugas3.go tugas4.go tugas5.go tugas6.go tugas7.go
1 package main
2
3 import "fmt"
4
5 func vals() (int, int) {
6     return 3, 7
7 }
8
9 func main() {
10     a, b := vals()
11     fmt.Println(a)
12     fmt.Println(b)
13
14     _, c := vals()
15     fmt.Println(c)
16 }
```

PROBLEMS 5 OUTPUT DEBUG CONSOLE TERMINAL 8: bash

```
[nandasaf@localhost Documents]$ go run tugas7.go
3
7
7
[nandasaf@localhost Documents]$
```

b.

```
tugas3.go  tugas4.go  tugas5.go  tugas6.go  tugas7.go  tugas8.go x
tugas8.go > {}main > main
12  numbPtr := flag.Int("numb", 42, "an int")
13  boolPtr := flag.Bool("fork", false, "a bool")
14
15  var svar string
16  flag.StringVar(&svar, "svar", "bar", "a string var")
17
18  flag.Parse()
19
20  fmt.Println("word:", *wordPtr)
21  fmt.Println("numb:", *numbPtr)
22  fmt.Println("fork:", *boolPtr)
23  fmt.Println("svar:", svar)
24  fmt.Println("tail:", flag.Args())
25
26
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL 8: bash +
[nandasaf@localhost Documents]$ go run tugas8.go
word: foo
numb: 42
fork: false
svar: bar
tail: []
[nandasaf@localhost Documents]$
```

```
tugas4.go  tugas5.go  tugas6.go  tugas7.go  tugas8.go x  tugas9.go
tugas8.go > {}main > main
4   "flag"
5   "fmt"
6
7
8  func main(){
9
10  wordPtr := flag.String("word", "foo", "a string")
11
12  numbPtr := flag.Int("numb", 42, "an int")
13  boolPtr := flag.Bool("fork", false, "a bool")
14
15  var svar string
16  flag.StringVar(&svar, "svar", "bar", "a string var")
17
18  flag.Parse()
19
20  fmt.Println("word:", *wordPtr)
21  fmt.Println("numb:", *numbPtr)
22  fmt.Println("fork:", *boolPtr)
23  fmt.Println("svar:", svar)
24  fmt.Println("tail:", flag.Args())
25
26
PROBLEMS 7 OUTPUT DEBUG CONSOLE TERMINAL 8: bash +
[nandasaf@localhost Documents]$ go run tugas8.go -word=nanda -numb=10 -svar=a b c d
word: nanda
numb: 10
fork: false
svar: a
tail: [b c d]
[nandasaf@localhost Documents]$
```

No 6.

Server menjalankan program sementara client menunggu request pesan dari server

```
tugas1.go  tugas2.go  tugas3.go  tugas4.go  tugas5.go  tugas6.go  tugas7.go  tugas8.go  tugas9.go x
tugas9.go > {}main > main
1  package main
2
3  import (
4      "fmt"
5      "net/http"
6  )
7
8  func main(){
9      http.HandleFunc("/", func(w http.ResponseWriter, r *http.Request) {
10         fmt.Fprintf(w, "Hello, you've requested: %s\n", r.URL.Path)
11     })
12
13     http.ListenAndServe(":8000", nil)
14 }
15
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL
[nandasaf@localhost Documents]$ go run tugas9.go
```

