

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 4
I/O, DATA TYPES & VARIABLES



Disusun oleh:

Cofa Xavier Marvel

109082500001

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2025

L

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var time, second, minute, hour int

    fmt.Scan(&time)

    second = time % 60
    minute = (time % 3600) / 60
    hour = time / 3600

    fmt.Printf("%d jam %d menit %d detik", hour, minute, second)
}
```

Screenshot program

The screenshot shows a Windows desktop environment. On the left, a code editor window (VS Code) displays the Go code for 'Guided1.go'. The code calculates the hours, minutes, and seconds from a given total number of seconds. On the right, a text editor window (Notepad) shows student information:

| | |
|--------|----------------|
| Nama | : Cofa Xavier |
| Marvel | |
| Nim | : 109082500001 |
| Kelas | : IF-13-04 |

At the bottom left, a terminal window shows the command-line output of running the Go program:

```
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided1.go"
3661
1 jam 1 menit 1 detik
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided1.go"
7322
2 jam 2 menit 2 detik
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided1.go"
3600
1 jam 0 menit 0 detik
PS C:\Code> [ ]
```

Deskripsi program

This program calculates the amount of time in hours, minutes, and seconds.

This program finds the number of hours by dividing the amount of second by 3600 because there are 3600 seconds in a single hour.

The program finds the second count by the modulo of the seconds and 3600, finding the leftover seconds from calculating the hours, then divided by 60 as there are 60 seconds in a minute.

Then the seconds are found by the modulo of the number of seconds and 60,finding the left overs of the left overs out putting the seconds that did not make to the minutes or the hours.

It is the outputted using `fmt.Printf` allowing the printing of integers in conjunction of the string.

2. Guided 2 Source Code

```
package main

import "fmt"

func main() {
    var in, p1, p2, p3 int
    var out bool

    fmt.Scan(&in)

    p1 = in % 10
    p2 = (in / 10) % 10
    p3 = in / 100

    fmt.Println(p3 < p2 && p2 < p1)
}
```

Screenshot program

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var in, p1, p2, p3 int
7
8     fmt.Scan(&in)
9
10    p1 = in % 10
11    p2 = (in / 10) % 10
12    p3 = in / 100
13
14    fmt.Println(p3 < p2 && p2 < p1)
15 }
16

```

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

GoCode\Modul-4\Guided\Guided2.go:7:6: declared and not used: out
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided2.go"
# command-line-arguments
GoCode\Modul-4\Guided\Guided2.go:7:6: declared and not used: out
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided2.go"
362
false
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided2.go"
256
true
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided2.go"
189
true
PS C:\Code>

```

This research empirically indicates that the student's understanding of Go programming concepts is strong, particularly in handling integers and basic control structures like loops and conditionals.

Deskripsi program

This program first separates the number in a variable using modulo and division

All variables being integers allow the use of modulo and division.

Dividing 362 by 100 gets 3.

Dividing 256 by 10 gets 25 modulo by 10 gets 5.

Modulo 189 by 10 get 9.

By comparing the separated numbers using the greater/lesser than and the and operator.

Printing the result, that being a bool.

3. Guided 3

Source Code

```

package main

import "fmt"

func main() {

    var bodyWeight, bodyHeight, bmi float64

    fmt.Scan(&bodyWeight, &bodyHeight)

    bmi = (bodyHeight * bodyHeight) / bodyWeight

    fmt.Printf("%.2f", bmi)
}

```

Screenshot program

The screenshot shows a GoCode IDE interface. On the left, the code editor displays a file named Guided3.go with the following content:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var bodyWeight, bodyHeight, bmi float64
7
8     fmt.Scan(&bodyWeight, &bodyHeight)
9
10    bmi = bodyWeight / (bodyHeight * bodyHeight)
11
12    fmt.Printf("%.2f", bmi)
13
14 }
15

```

Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab shows the command `go run "c:\Code\GoCode\Modul-4\Guided\Guided3.go"` and its output:

```

> go run "c:\Code\GoCode\Modul-4\Guided\Guided3.go"
70 1.75
22.86
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided3.go"
60 1.6
23.44
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Guided\Guided3.go"
88 1.8
24.69
PS C:\Code>

```

To the right of the code editor is a separate window titled "package main" containing student information:

| | |
|---------------|----------------|
| <u>Nama</u> | : Cofa Xavier |
| <u>Marvel</u> | |
| <u>Nim</u> | : 109082500001 |
| <u>Kelas</u> | : IF-13-04 |

Deskripsi program

This program uses the standard fare.

Declare bodyweight, bodyheight and bmi as floats

Scan and assign them.

Then calculate the bmi by bodyweight divide by bodyweight times bodyweight.

Print using Printf to print the float.

TUGAS

Tugas 1

Source code

```

package main

import "fmt"

func main() {
    var price, discount int
    //declares price and discount as intergers

    fmt.Scan(&price, &discount)
    //Scans for them both

    fmt.Println(price - ((price * discount) / 100))
    //Print the result of the price being discounted by the discount
}

```

Screenshot program

The screenshot shows a Go development environment with several windows open:

- Code Editor:** Displays the file `Tugas1.go` with the following content:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var price, discount int
7     //declares price and discount as intergers
8
9     fmt.Scan(&price, &discount)
//Scans for them both
10
11     fmt.Println(price - ((price * discount) / 100))
//Print the result of the price being discounted by the discount
12 }
13
14 }
```
- Terminal:** Shows the command `go run "c:\Code\GoCode\Modules-4\Tugas\Tugas1.go"` being run multiple times with different inputs (100000, 150000, 127500) and their corresponding outputs (100000, 150000, 127500).
- Note:** A note window titled "package main" contains student information:

| | |
|--------|----------------|
| Nama | : Cofa Xavier |
| Marvel | |
| Nim | : 109082500001 |
| Kelas | : IF-13-04 |

Deskripsi program

This program discounts the price variable using the equation.

(Price – ((Price * discount) / 100))

The equation is contained in the Printline command.

Tugas 2

Source code

```
package main

import "fmt"

func main() {

    var bodyWeight, bodyHeight, bmi float64

    fmt.Scan(&bmi, &bodyHeight)

    bodyWeight = bmi * (bodyHeight * bodyHeight)

    fmt.Printf("%.f", bodyWeight)

}
```

Screenshot program

The screenshot shows a Go code editor interface with several tabs open. The active tab contains the following Go code:

```

package main

import "fmt"

func main() {
    var bodyWeight, bodyHeight, bmi float64
    fmt.Scan(&bmi, &bodyHeight)
    bodyWeight = bmi * (bodyHeight * bodyHeight)
    fmt.Printf("%.f", bodyWeight)
}

```

Below the code editor is a terminal window showing the output of running the program three times:

```

PS C:\Code> go run "c:\Code\GoCode\Modul-4\Tugas\Tugas2.go"
22.85 1.75
70
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Tugas\Tugas2.go"
23.43 1.6
60
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Tugas\Tugas2.go"
24.69 1.8
80
PS C:\Code>

```

Deskripsi program

All variables are floats.

This program calculates bodyweight by multiplying the bmi by the bodyheight times bodyheight of a person.

Then it prints the float using printf.

Tugas 3

Source code

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var x1, y1, x2, y2, x3, y3 float64

    fmt.Scanln(&x1, &y1)
    fmt.Scanln(&x2, &y2)
    fmt.Scanln(&x3, &y3)

    ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
    bc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
    ca := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3, 2))

    longest := ab
}

```

```

        if bc > longest {
            longest = bc
        }
        if ca > longest {
            longest = ca
        }

        fmt.Printf("%.2f\n", longest)
    }
}

```

Screenshot program

The screenshot displays two windows side-by-side. The left window is a code editor showing Go code. The right window is a text editor showing student information.

Code Editor (Left):

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var x1, y1, x2, y2, x3, y3 float64
10
11     fmt.Scanln(&x1, &y1)
12     fmt.Scanln(&x2, &y2)
13     fmt.Scanln(&x3, &y3)
14
15     ab := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
16     bc := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
17     ca := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3, 2))
18
19     longest := ab
20     if bc > longest {
21         longest = bc
22     }
23     if ca > longest {
24         longest = ca
25     }
26
27     fmt.Printf("%.2f\n", longest)
28 }

```

Text Editor (Right):

| | | |
|---------------|---|--------------|
| <u>Nama</u> | : | Cofa Xavier |
| <u>Marvel</u> | : | |
| <u>Nim</u> | : | 109082500001 |
| <u>Kelas</u> | : | IF-13-04 |

The screenshot shows a GoCode IDE interface. On the left is a code editor with tabs for Tugas1.go, Tugas2.go, Tugas3.go (the active tab), PTCOFA1.go, PTCOFA2.go, and Guided3.go. The code in Tugas3.go is:

```
func main() {
    var x1, y1, x2, y2, x3, y3 float64
    fmt.Scan(&x1, &y1)
    fmt.Scan(&x2, &y2)
    fmt.Scan(&x3, &y3)

    side1 := math.Sqrt((x2 - x1)*(x2 - x1) + (y2 - y1)*(y2 - y1))
    side2 := math.Sqrt((x3 - x1)*(x3 - x1) + (y3 - y1)*(y3 - y1))
    side3 := math.Sqrt((x3 - x2)*(x3 - x2) + (y3 - y2)*(y3 - y2))

    if side1 > side2 && side1 > side3 {
        fmt.Println(side1)
    } else if side2 > side1 && side2 > side3 {
        fmt.Println(side2)
    } else {
        fmt.Println(side3)
    }
}
```

The terminal window shows the output of running the program with various inputs:

```
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Tugas\Tugas3.go"
1 1
4 1
1 5
5.00
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Tugas\Tugas3.go"
0 0
4 2
0 3
4.47
PS C:\Code>
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Tugas\Tugas3.go"
3 0
3 4
0 0
5.00
PS C:\Code> go run "c:\Code\GoCode\Modul-4\Tugas\Tugas3.go"
0 0
1 3
3 2
3.61
PS C:\Code>
```

On the right is a text editor window titled 'Nama' containing student information:

| | | |
|--------------|--------|----------|
| Nama | : | Cofa |
| Xavier | Marvel | |
| Nim | : | |
| 109082500001 | | |
| Kelas | : | IF-13-04 |

Below the text editor are status bars for file, edit, view, and other system information.

Deskripsi program

This program exists to calculate the lengths of the sides of the triangle formed by these points and determine the longest side of the triangle.

The input is formatted as three lines, each of which contains two real numbers representing the coordinates of points A, B, and C in x-y format.

The output is a float stating the length of the longest side of the triangle formed by these points.

Using a brand new, never seen, if statement to compare and choose the longest side.