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

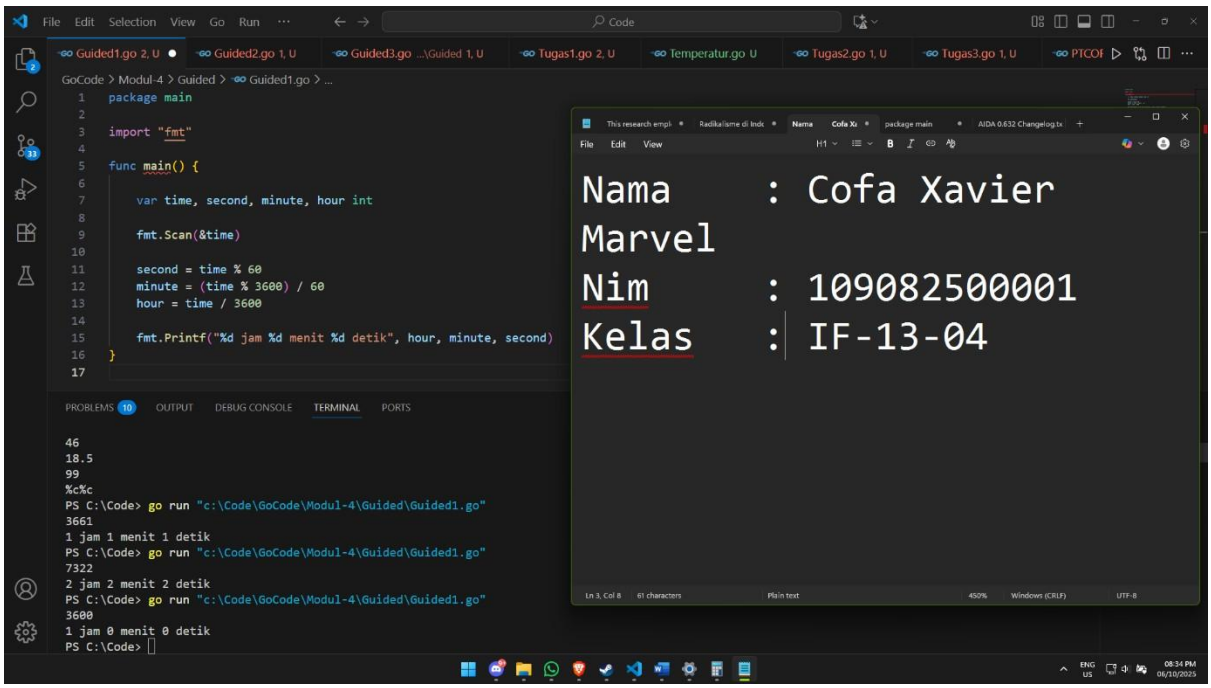
L

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) }
```

Screenshoot program



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.

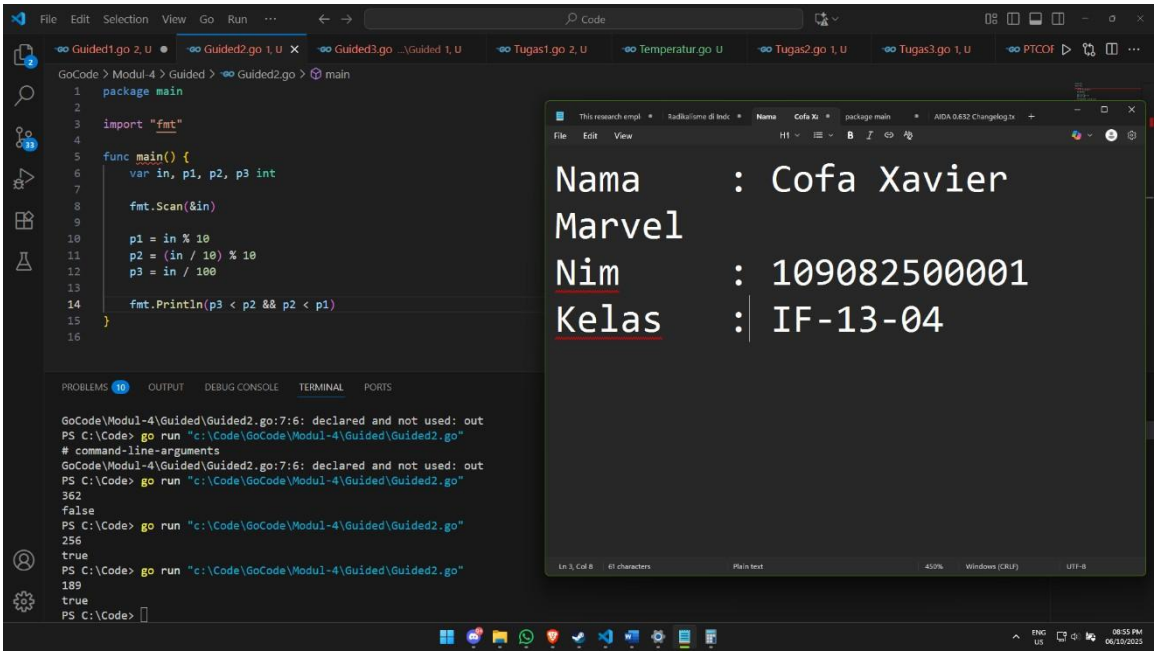
It is the outputted using fmt.Printf allowing the printing of integers in conjunction of the string.

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) }
```

Screenshoot program



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 362 by 10 gets 25 modulo by 10 gets 6.

Modulo 362 by 10 get 2.

By comparing the separated numbers using the [</>] greater/lesser than and the [&&]and operator.

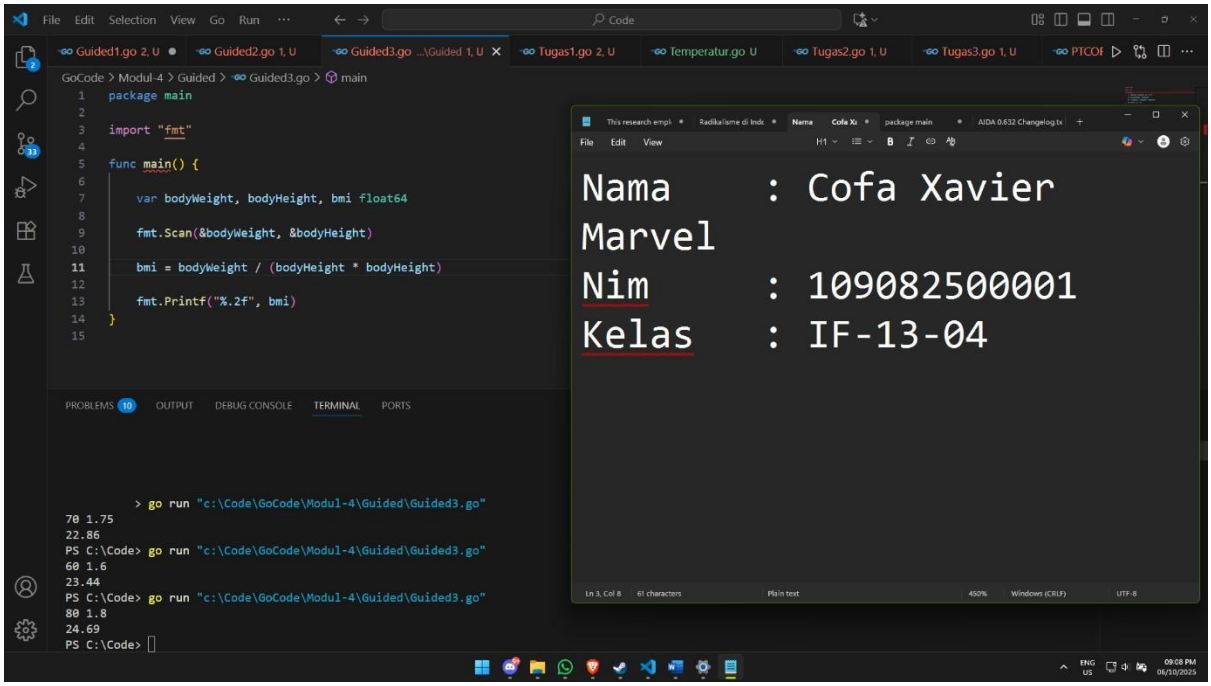
Printing the result, that being a bool.

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)
}
```

Screenshoot program



Deskripsi program

Declare bodyweight, bodyheight and bmi as floats.

Scan and assign them.

Then calculate the bmi by using weight / (height * height).

using Printf to print the float.

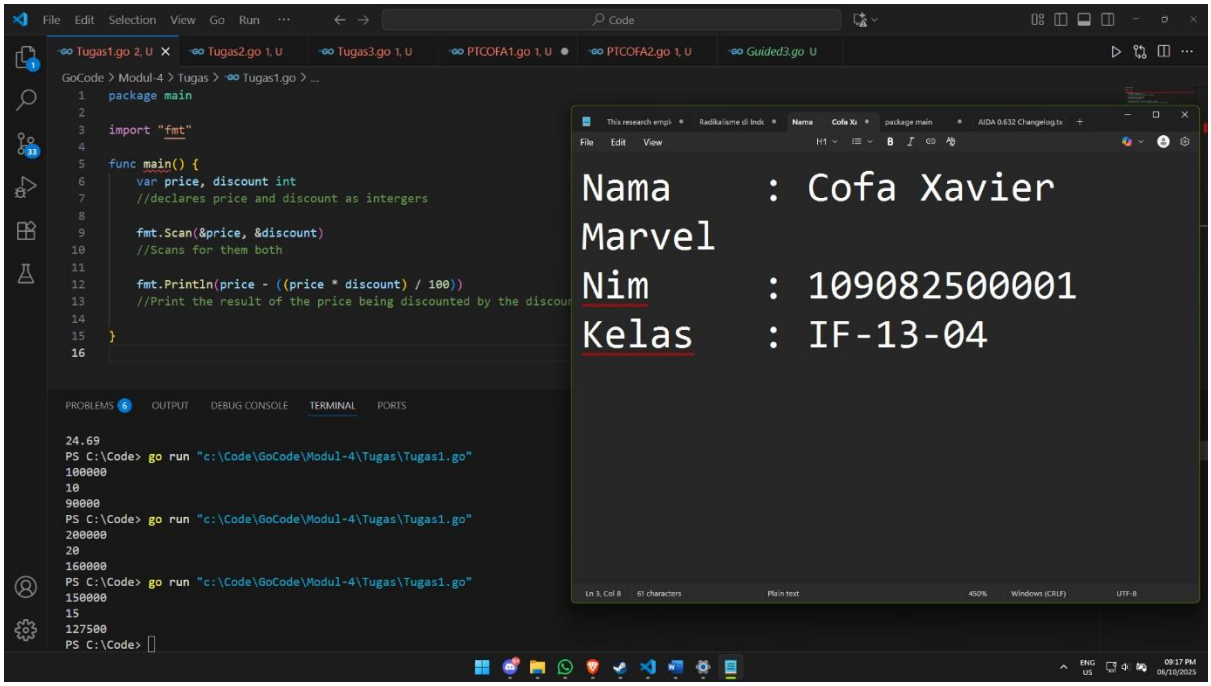
TUGAS

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

}
```

Screenshoot program



Deskripsi program

This program discounts the price variable using the equation.

(Price – ((Price * discount) / 100)))

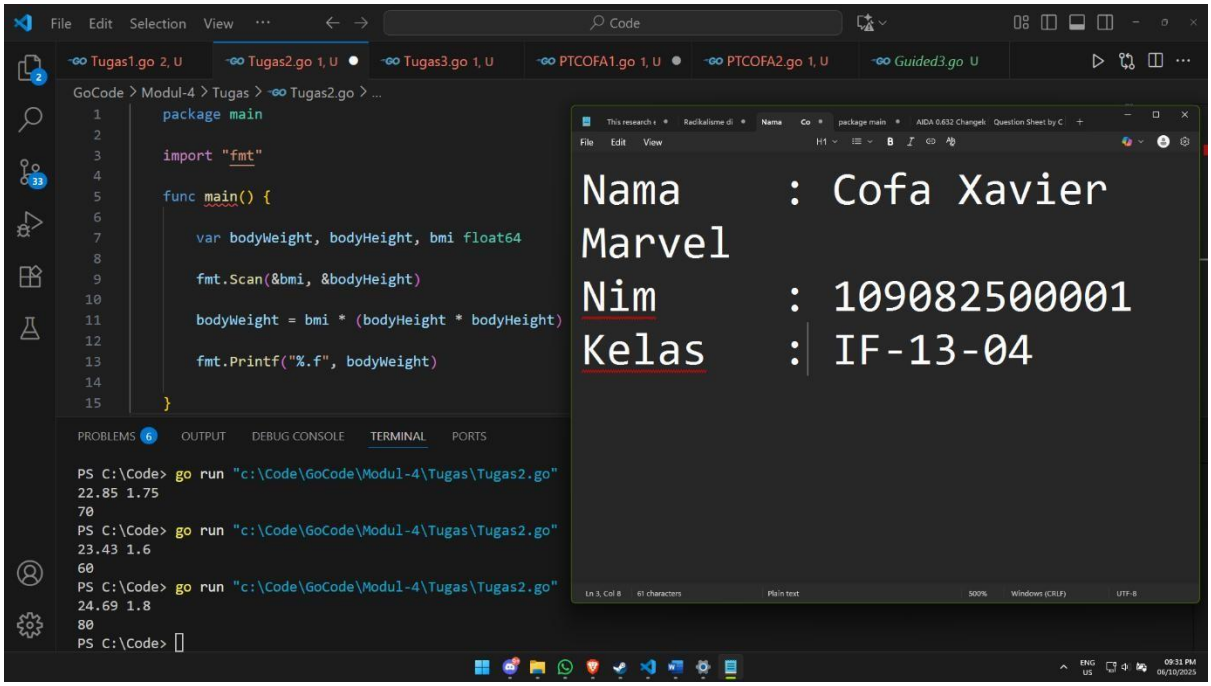
The equation is contained in the Printline command.

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)

}
```

Screenshoot program



Deskripsi program

All variables are floats.

This program calculates bodyweight using the formula

Weight = bmi * (height * height)

Then it prints the Weight using printf.

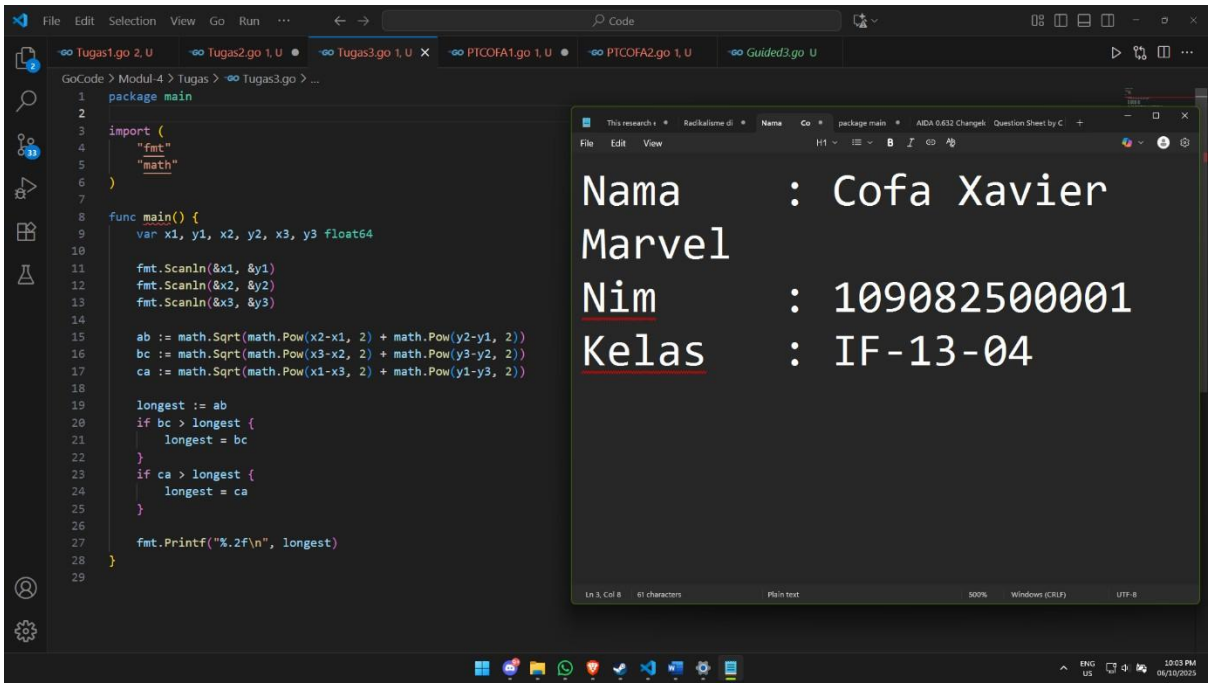
Tugas3

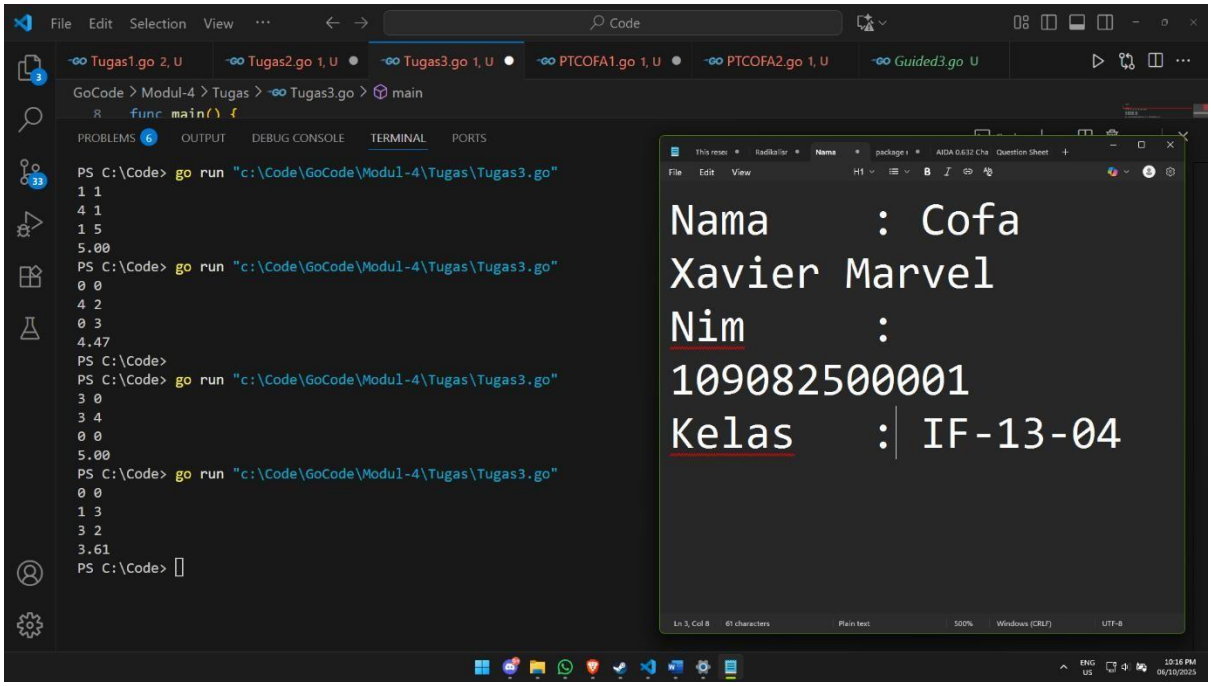
Source code

```go
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)
}
```

Screenshoot program





Deskripsi program

This program exists to calculate the lengths of the sides of the triangle formed from three 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 1,2 and 3 in x-y format.

Ex: x1 y1

x2 y2

x3 y3

The program calculates the length using the formula.

$d=\sqrt{((x2-x1)^2+(y2-y1)^2)}$

d = length

Using a brand new, never seen, if statement to compare and choose the longest side. The chosen longest is then printed using fmt.Printf.

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