

LAPORAN PRAKTIKUM
Algoritma Pemrograman

MODUL 4
TIPE DATA & VARIABEL



Disusun oleh:
EWALDO ARDIANSYAH WIDYADHANA
109082500008
S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025

LATIHAN KELAS – GUIDED

1. Guided 1 Source Code

```
package main

import "fmt"

func main() {

    var detik, jam, menit int

    fmt.Scan(&detik)

    jam = detik / 3600

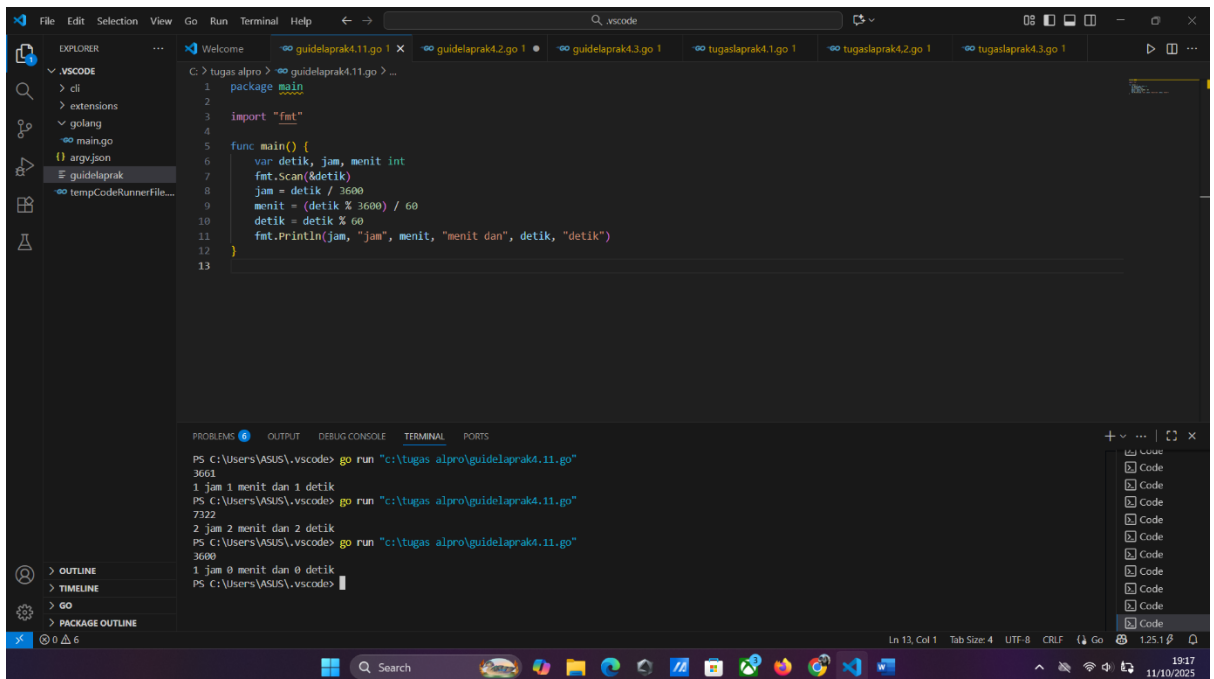
    menit = (detik % 3600) / 60

    detik = detik % 60

    fmt.Println(jam, "jam", menit, "menit dan", detik,
"detik")

}
```

Screenshoot program



The screenshot displays the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows the project structure with files like `.vscode`, `golang`, `main.go`, and `guidelaprak`. The main editor window shows a Go program that takes an input in seconds and converts it into hours, minutes, and seconds. The terminal at the bottom shows the execution of the program with the command `go run "c:\tugas alpro\guidelaprak4.11.go"`. The output shows the program running three times with different inputs: 3661, 7322, and 3600 seconds, each resulting in a formatted string of hours, minutes, and seconds.

```
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\guidelaprak4.11.go"
3661
1 jam 1 menit dan 1 detik
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\guidelaprak4.11.go"
7322
2 jam 2 menit dan 2 detik
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\guidelaprak4.11.go"
3600
1 jam 0 menit dan 0 detik
PS C:\Users\VASUS\.vscode>
```

Deskripsi program

Program ini berfungsi untuk mengonversi waktu dari satuan detik menjadi jam, menit, dan detik agar lebih mudah dipahami oleh pengguna. Dalam komputer, waktu sering direpresentasikan dalam bentuk total detik, sehingga program ini membantu mengubahnya menjadi format yang lebih familiar.

Pengguna cukup memasukkan sebuah bilangan bulat positif yang menunjukkan jumlah detik. Program kemudian menghitung jumlah jam dengan membagi total detik dengan 3600, menghitung menit dari sisa detik yang tersisa, dan sisanya menjadi detik terakhir. Hasil perhitungan tersebut akan ditampilkan dalam bentuk kalimat deskriptif, misalnya "1 jam 1 menit dan 1 detik". Dengan cara ini, program memudahkan pengguna untuk mengetahui lama waktu dalam format jam, menit, dan detik secara cepat dan akurat

2. Guided 2

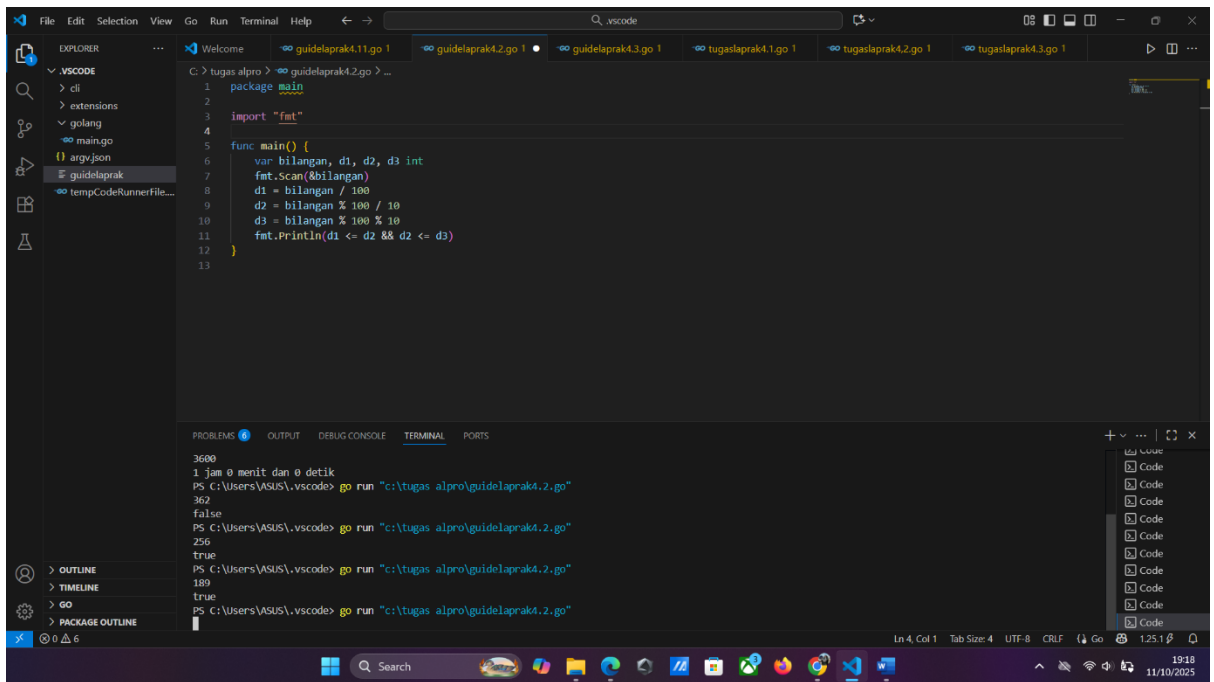
Source Code

```
package main

import "fmt"

func main() {
    var bilangan, d1, d2, d3 int
    fmt.Scan(&bilangan)
    d1 = bilangan / 100
    d2 = bilangan % 100 / 10
    d3 = bilangan % 100 % 10
    fmt.Println(d1 <= d2 && d2 <= d3)
}
```

Screenshoot program



Deskripsi program

untuk menentukan apakah setiap digit pada suatu bilangan tiga angka tersusun secara terurut membesar atau tidak. Pengguna akan memasukkan sebuah bilangan bulat positif antara 100 hingga 999. Program kemudian memisahkan setiap digit dari bilangan tersebut, lalu membandingkan apakah digit pertama lebih kecil dari digit kedua dan digit kedua lebih kecil dari digit ketiga. Jika kondisi tersebut terpenuhi, maka hasil yang ditampilkan adalah "true", menandakan bilangan tersebut terurut membesar. Sebaliknya, jika tidak memenuhi syarat, program akan menampilkan "false". Program ini membantu pengguna memahami pola kenaikan pada setiap digit dalam sebuah bilangan secara sederhana dan cepat.

3. Guided 3

Source Code

```
package main

import "fmt"

func main() {

    var beratbadan, tinggibadan, bmi float64

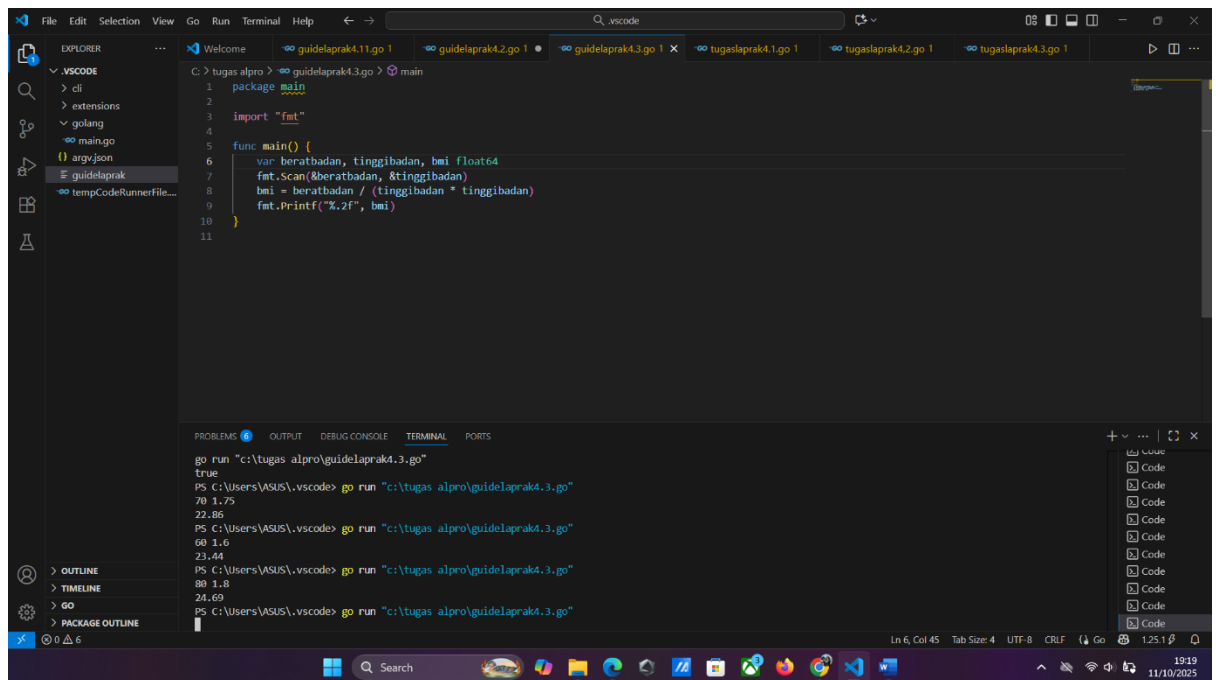
    fmt.Scan(&beratbadan, &tinggibadan)

    bmi = beratbadan / (tinggibadan * tinggibadan)

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

}
```

Screenshoot program



Deskripsi program

menghitung nilai BMI (Body Mass Index) seseorang berdasarkan berat badan dalam kilogram dan tinggi badan dalam meter. Pengguna akan memasukkan dua nilai, yaitu berat dan tinggi badan. Program kemudian menghitung nilai BMI dengan menggunakan

rumus:

$BMI = \text{berat} / (\text{tinggi} \times \text{tinggi})$.

Hasil perhitungan tersebut akan menampilkan nilai BMI dalam bentuk bilangan desimal. Program ini membantu pengguna mengetahui indeks massa tubuhnya dengan cepat dan akurat berdasarkan data berat serta tinggi badan yang dimasukkan.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {
    var totalAwal, diskon int
```

```

    fmt.Print("Masukkan total belanja awal: ")

    fmt.Scan(&totalAwal)

    fmt.Print("Masukkan besarnya diskon (%): ")

    fmt.Scan(&diskon)

    totalAkhir := totalAwal - (totalAwal * diskon / 100)

    fmt.Println("Total belanja akhir setelah diskon adalah:",
totalAkhir)
}

```

Screenshoot program

The screenshot shows a Go program in VS Code. The source code is as follows:

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var totalAwal, diskon int
7     fmt.Print("Masukkan total belanja awal: ")
8     fmt.Scan(&totalAwal)
9     fmt.Print("Masukkan besarnya diskon (%): ")
10    fmt.Scan(&diskon)
11
12    totalAkhir := totalAwal - (totalAwal * diskon / 100)
13
14    fmt.Println("Total belanja akhir setelah diskon adalah:", totalAkhir)
15 }
16

```

The terminal output shows the program being run three times with different inputs:

```

PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\tugaslaprak4.1.go"
Masukkan total belanja awal: 100000 10
Masukkan besarnya diskon (%): Total belanja akhir setelah diskon adalah: 90000
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\tugaslaprak4.1.go"
Masukkan total belanja awal: 200000 20
Masukkan besarnya diskon (%): Total belanja akhir setelah diskon adalah: 160000
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\tugaslaprak4.1.go"
Masukkan total belanja awal: 1500000 15
Masukkan besarnya diskon (%): Total belanja akhir setelah diskon adalah: 1275000
PS C:\Users\VASUS\.vscode>

```

Deskripsi program

Program ini berfungsi untuk menghitung total harga belanja setelah mendapatkan potongan diskon dengan besaran tertentu. Pengguna memasukkan dua data, yaitu total belanja awal dan persentase diskon. Program kemudian menghitung total harga akhir dengan mengurangi total belanja awal berdasarkan persentase diskon yang diberikan. Hasil akhirnya berupa total belanja setelah diskon yang ditampilkan dalam bentuk angka. Program ini membantu pengguna mengetahui jumlah yang harus dibayar setelah memperoleh potongan harga secara cepat dan akurat

2. Tugas 2

Source code

```
package main

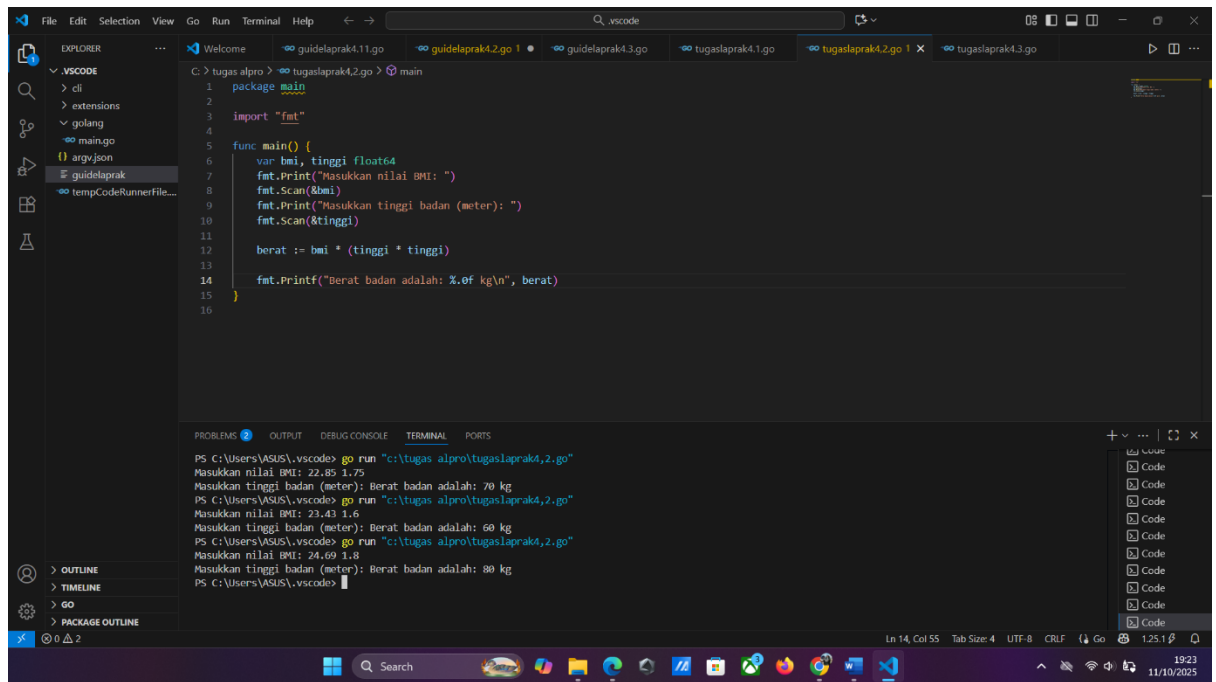
import "fmt"

func main() {
    var bmi, tinggi float64
    fmt.Print("Masukkan nilai BMI: ")
    fmt.Scan(&bmi)
    fmt.Print("Masukkan tinggi badan (meter): ")
    fmt.Scan(&tinggi)

    berat := bmi * (tinggi * tinggi)

    fmt.Printf("Berat badan adalah: %.0f kg\n", berat)
}
```

Screenshoot program



Deskripsi program

Program ini digunakan untuk menentukan berat badan seseorang berdasarkan nilai BMI dan tinggi badan. Pengguna memasukkan dua nilai, yaitu BMI (Body Mass Index) dan tinggi badan dalam meter. Program kemudian menghitung berat badan dengan menggunakan rumus: $\text{berat} = \text{BMI} \times (\text{tinggi} \times \text{tinggi})$. Hasil perhitungan akan menampilkan berat badan seseorang dalam satuan kilogram. Program ini membantu pengguna mengetahui berat badan yang sesuai dengan nilai BMI dan tinggi badan yang dimasukkan secara cepat dan akurat

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

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

    // Input titik A
    fmt.Println("Masukkan koordinat titik A (x y): ")
    fmt.Scan(&x1, &y1)

    // Input titik B
    fmt.Println("Masukkan koordinat titik B (x y): ")
```



```

fmt.Scan(&x2, &y2)

// Input titik C
fmt.Print("Masukkan koordinat titik C (x y): ")
fmt.Scan(&x3, &y3)

// Hitung panjang sisi-sisi segitiga
sisiAB := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
sisiBC := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
sisiCA := math.Sqrt(math.Pow(x3-x1, 2) + math.Pow(y3-y1, 2))

// Tentukan sisi terpanjang
sisiTerpanjang := sisiAB
if sisiBC > sisiTerpanjang {
    sisiTerpanjang = sisiBC
}
if sisiCA > sisiTerpanjang {
    sisiTerpanjang = sisiCA
}

// Tampilkan hasil dengan dua angka di belakang koma
fmt.Printf("Sisi terpanjang adalah: %.2f\n", sisiTerpanjang)
}

```

Screenshoot program

The screenshot shows the Visual Studio Code (VS Code) interface. The Explorer panel on the left shows the project structure with files like `main.go` and `main.json`. The main editor displays a Go program that calculates the longest side of a triangle given three points A, B, and C. The program uses the `math` package for square root calculations. The terminal at the bottom shows the command `go run "c:\tugas alpro\tugaslaprak4.3.go"` being executed, and the output shows the coordinates for points A, B, and C, followed by the calculated longest side, which is 5.00.

```

C:\tugas alpro> go run "c:\tugas alpro\tugaslaprak4.3.go"
package main
import (
    "fmt"
    "math"
)
func main() {
    var x1, y1, x2, y2, x3, y3 float64
    // Input titik A
    fmt.Print("Masukkan koordinat titik A (x y): ")
    fmt.Scan(&x1, &y1)
    // Input titik B
    fmt.Print("Masukkan koordinat titik B (x y): ")
    fmt.Scan(&x2, &y2)
    // Input titik C
    fmt.Print("Masukkan koordinat titik C (x y): ")
    fmt.Scan(&x3, &y3)
    // Hitung panjang sisi-sisi segitiga
    sisiAB := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))
    sisiBC := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))
    sisiCA := math.Sqrt(math.Pow(x3-x1, 2) + math.Pow(y3-y1, 2))
    // Tentukan sisi terpanjang
    sisiTerpanjang := sisiAB
    if sisiBC > sisiTerpanjang {
        sisiTerpanjang = sisiBC
    }
    if sisiCA > sisiTerpanjang {
        sisiTerpanjang = sisiCA
    }
    // Tampilkan hasil dengan dua angka di belakang koma
    fmt.Printf("Sisi terpanjang adalah: %.2f\n", sisiTerpanjang)
}

```

PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\tugaslaprak4.3.go"

Masukkan koordinat titik A (x y): 0.0 0.0 3.0 0.0 3.0 4.0

Masukkan koordinat titik B (x y): 0.0 0.0 3.0 0.0 3.0 4.0

Masukkan koordinat titik C (x y): 0.0 0.0 3.0 0.0 3.0 4.0

Sisi terpanjang adalah: 5.00

Deskripsi program

Program ini dibuat untuk menghitung panjang sisi-sisi segitiga serta menentukan sisi terpanjangnya berdasarkan tiga titik koordinat yang diberikan dalam sistem kartesius dua dimensi. Pengguna akan memasukkan tiga pasang bilangan riil yang merepresentasikan koordinat titik A, B, dan C. Selanjutnya, program menggunakan teorema Pythagoras untuk menghitung jarak antar setiap pasangan titik dengan rumus $\sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$. Setelah semua panjang sisi diketahui, program akan membandingkan hasil perhitungan tersebut untuk menentukan sisi terpanjang dari segitiga yang terbentuk. Hasil akhir ditampilkan dalam bentuk bilangan riil dengan dua angka di belakang koma agar lebih presisi dan mudah dibaca.

TUGAS PENDAHULUAN

Question Sheet

1. Float to int to ASCII

Create a program that converts float to int to ASCII

The input consists of four floats *c1,c2,c3,c4*

The output a four letter word

No.	Input	Output
1	51 55 55 99	func
2	41.5 46 48.5 110	Scan
3	18.5 46 18.5 99	%c%c

2. Booleans

Create a program that finds the truth value of D

The input consists of three Booleans *A, B, C*

The output is the truth value of D

No.	Input	Output
1	T T T	with A = true, B = true, and C = true the resulting truth value of D = false
2	T F T	with A = true, B = false, and C = true the resulting truth value of D = false
3	F F F	with A = false, B = false, and C = false the resulting truth value of D = true

Soal no 1

```
package main

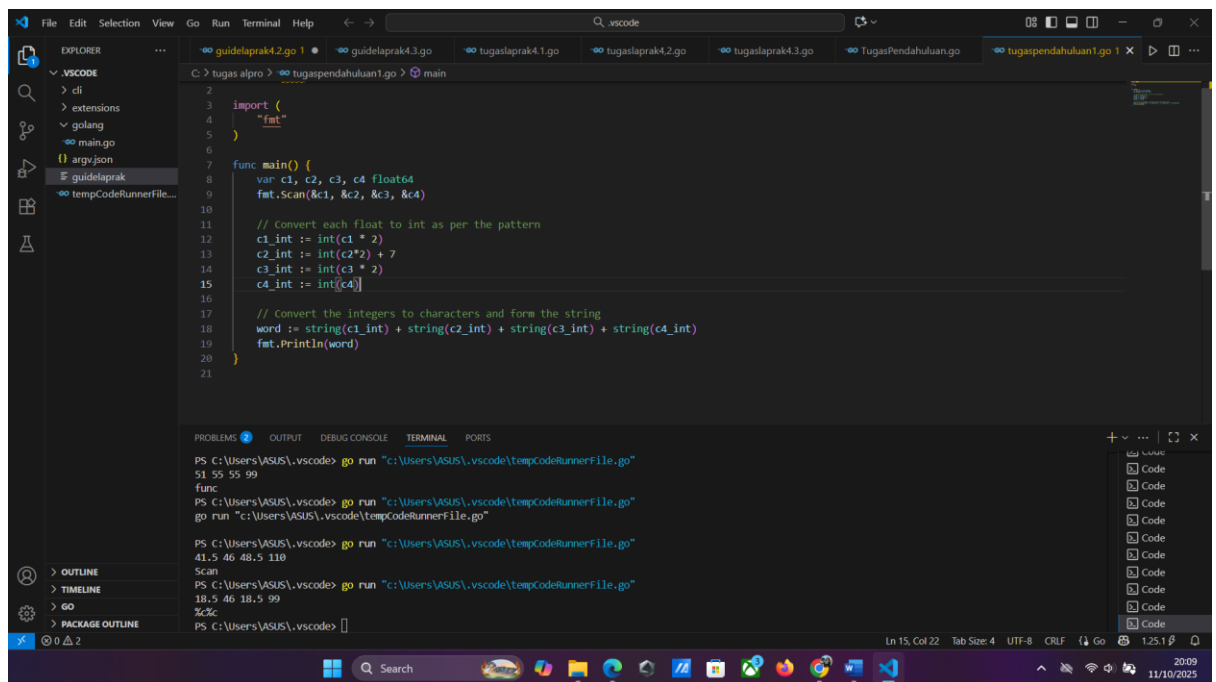
import (
    "fmt"
)

func main() {
    var c1, c2, c3, c4 float64
    fmt.Scan(&c1, &c2, &c3, &c4)

    // Convert each float to int as per the pattern
    c1_int := int(c1 * 2)
    c2_int := int(c2*2) + 7
    c3_int := int(c3 * 2)
    c4_int := int(c4)

    // Convert the integers to characters and form the string
    word := string(c1_int) + string(c2_int) + string(c3_int) + string(c4_int)
    fmt.Println(word)
}
```

Screenshoot program



Soal no 2

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var a, b, c bool
```

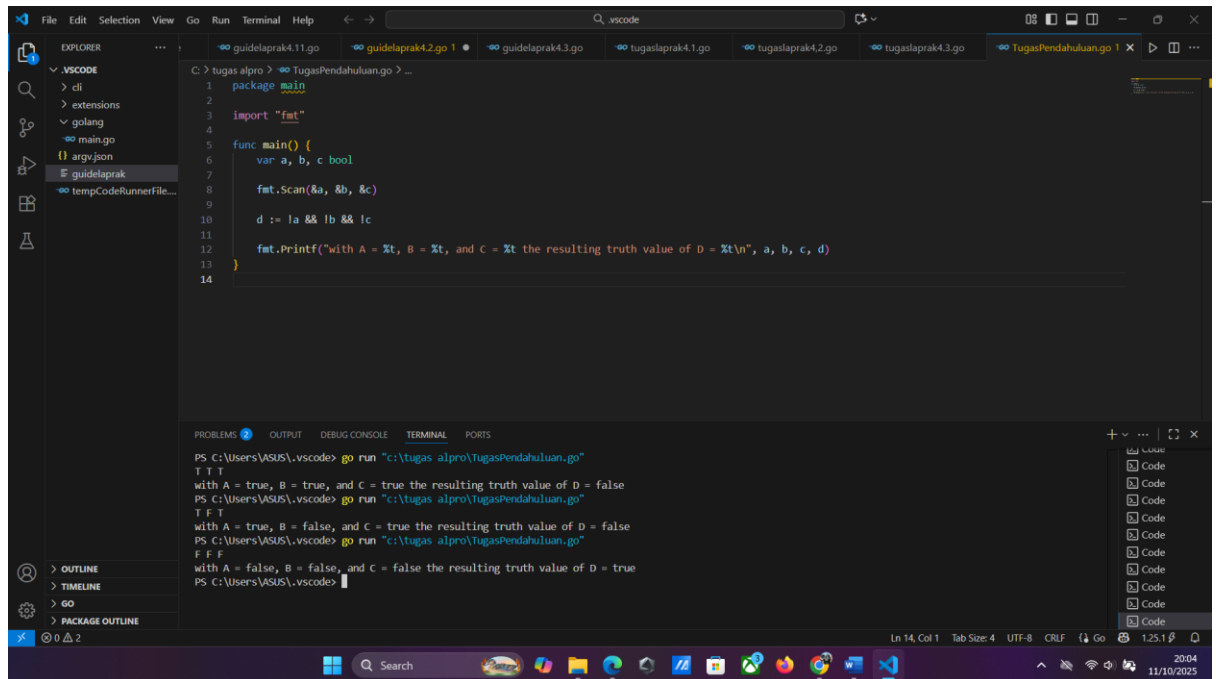
```
    fmt.Scan(&a, &b, &c)
```

```
    d := !a && !b && !c
```

```
    fmt.Printf("with A = %t, B = %t, and C = %t the resulting truth value of D = %t\n", a, b, c, d)
```

```
}
```

Screenshoot program



The screenshot displays the Visual Studio Code (VS Code) interface. The Explorer sidebar on the left shows a project structure with folders for .VSCode, cli, extensions, golang, main.go, argu.json, guidelaprak, and tempCodeRunnerFile... The main editor area shows a Go file named Tugaspendahuluan.go with the following code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c bool
7
8     fmt.Scan(&a, &b, &c)
9
10    d := !a && !b && !c
11
12    fmt.Printf("with A = %t, B = %t, and C = %t the resulting truth value of D = %t\n", a, b, c, d)
13 }
14
```

The Output panel at the bottom shows the execution results of the program, demonstrating the truth value of D for different combinations of A, B, and C:

```
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\tugaspendahuluan.go"
T T T
with A = true, B = true, and C = true the resulting truth value of D = false
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\tugaspendahuluan.go"
T F T
with A = true, B = false, and C = true the resulting truth value of D = false
PS C:\Users\VASUS\.vscode> go run "c:\tugas alpro\tugaspendahuluan.go"
F F F
with A = false, B = false, and C = false the resulting truth value of D = true
PS C:\Users\VASUS\.vscode>
```

The status bar at the bottom indicates the current file is Un 14, Col 1, Tab Size: 4, UTF-8, CRLF, and the version is 1.25.1. The date and time shown are 2024 11/10/2025.