

LAPORAN PRAKTIKUM

Algoritma Pemrograman

MODUL 4

MODUL 4. I/O, TIPE DATA & VARIABEL



Disusun oleh:

ALMA BONITA MIA WARDHANA

109082500015

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")

}
```

Screenshot program

```
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\contohsoal1\main.go"
3661
1 jam 1 menit dan 1 detik
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\contohsoal1\main.go"
7322
2 jam 2 menit dan 2 detik
PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\contohsoal1\main.go"
3600
1 jam 0 menit dan 0 detik
PS D:\Modul4_AlproAlma>
```

Deskripsi program

Program di atas ditulis menggunakan bahasa Go dan berfungsi untuk mengubah jumlah detik yang dimasukkan pengguna menjadi satuan jam, menit, dan detik. Pertama, program mengimpor paket "fmt" yang digunakan untuk menampilkan dan

menerima input dari pengguna. Di dalam fungsi main(), terdapat tiga variabel bertipe int, yaitu detik, jam, dan menit.

Ketika program dijalankan, pengguna diminta memasukkan jumlah detik. Nilai tersebut kemudian disimpan dalam variabel detik. Setelah itu, program menghitung jumlah jam dengan membagi total detik dengan 3600, karena dalam satu jam terdapat 3600 detik.

Selanjutnya, untuk menghitung menit, program menggunakan sisa pembagian detik setelah dikurangi jam, yaitu $(\text{detik \% } 3600) / 60$. Operasi ini mengambil sisa detik yang tidak cukup membentuk satu jam penuh, lalu mengubahnya menjadi menit dengan membaginya 60. Setelah itu, variabel detik diubah lagi menjadi sisa dari pembagian 60 ($\text{detik} = \text{detik \% } 60$) agar hanya menyisakan detik terakhir yang tidak cukup membentuk satu menit.

Terakhir, program menampilkan hasil konversi dalam format “x jam y menit dan z detik” menggunakan perintah `fmt.Println()`. Dengan begitu, pengguna dapat melihat berapa jam, menit, dan detik dari jumlah detik yang mereka masukkan.

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

```

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)
}

```

Deskripsi program

Program di atas merupakan program sederhana dalam bahasa Go yang berfungsi untuk memeriksa apakah tiga digit dari sebuah bilangan berurutan dari kecil ke besar atau tidak. Di awal program, terdapat deklarasi tiga variabel yaitu d1, d2, dan d3, yang masing-masing digunakan untuk menyimpan digit ratusan, puluhan, dan satuan dari bilangan yang dimasukkan oleh pengguna. Program kemudian meminta pengguna untuk mengetikkan sebuah bilangan, lalu membaca nilai tersebut menggunakan perintah `fmt.Scan(&bilangan)`.

Setelah nilai dimasukkan, program memecah bilangan tersebut menjadi tiga bagian. Variabel d1 diisi dengan hasil pembagian bilangan dengan 100, sehingga nilainya mewakili digit ratusan. Variabel d2 dihitung dari sisa pembagian bilangan dengan 100, kemudian hasilnya dibagi 10, yang berarti mengambil digit puluhan. Sedangkan d3 dihitung dari sisa pembagian bilangan dengan 100, lalu diambil sisa 10-nya, sehingga nilainya mewakili digit satuan.

Setelah ketiga digit diperoleh, program melakukan perbandingan dengan ekspresi `d1 <= d2 && d2 <= d3`. Artinya, program akan mengecek apakah digit pertama lebih kecil atau sama dengan digit kedua, dan digit kedua lebih kecil atau sama dengan digit ketiga. Jika kedua kondisi ini benar, maka hasilnya true; sebaliknya, jika salah satu tidak terpenuhi, hasilnya false. Hasil ini kemudian ditampilkan di layar menggunakan `fmt.Println`. Misalnya, jika pengguna memasukkan angka 345, program akan menampilkan true, sedangkan jika memasukkan 531, hasilnya akan false.

3. Guided 3

Source Code

```
package main
```

```

import "fmt"

func main() {

    var beratBadan, tinggiBadan, bmi float64

    fmt.Scan(&beratBadan, &tinggiBadan)

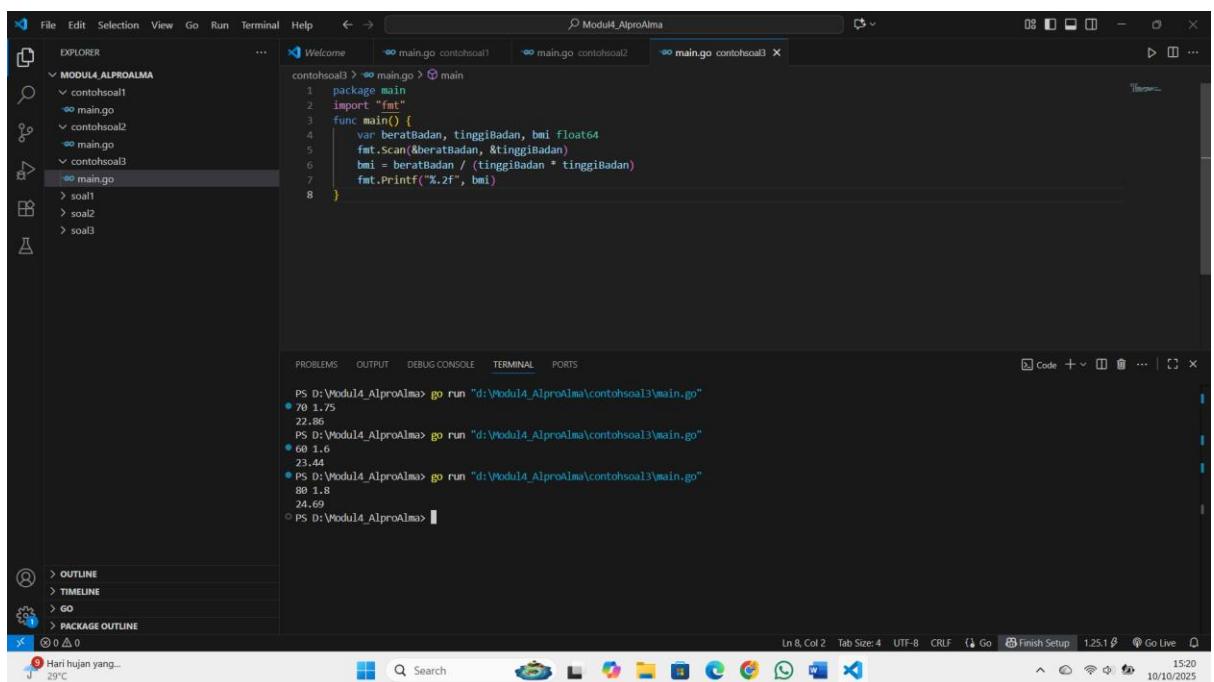
    bmi = beratBadan / (tinggiBadan * tinggiBadan)

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

}

```

Screenshot program



Deskripsi program

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk menghitung BMI (Body Mass Index) atau Indeks Massa Tubuh seseorang berdasarkan berat badan dan tinggi badan yang dimasukkan. Di dalam program, terdapat tiga variabel bertipe float64, yaitu beratBadan, tinggiBadan, dan bmi. Tipe data ini dipilih karena nilai berat dan tinggi badan biasanya mengandung angka desimal, misalnya 60.5 kilogram atau 1.70 meter.

Ketika program dijalankan, pengguna diminta untuk memasukkan dua nilai, yaitu berat badan dan tinggi badan, yang dibaca menggunakan perintah `fmt.Scan(&beratBadan, &tinggiBadan)`. Setelah kedua nilai tersebut dimasukkan,

program akan menghitung nilai BMI menggunakan rumus standar, yaitu berat badan dibagi dengan kuadrat dari tinggi badan. Rumus ini ditulis dalam kode sebagai `bmi = beratBadan / (tinggiBadan * tinggiBadan)`.

Hasil perhitungan kemudian ditampilkan di layar menggunakan perintah `fmt.Printf("%2f", bmi)`, yang berarti hasil BMI akan dicetak dengan dua angka di belakang koma agar tampak lebih rapi dan mudah dibaca. Misalnya, jika pengguna memasukkan berat badan 60 dan tinggi badan 1.65, maka program akan menghitung BMI-nya menjadi sekitar 22.04. Program ini membantu pengguna mengetahui nilai BMI mereka, yang bisa digunakan untuk menilai apakah berat badan tergolong ideal, kurang, atau berlebih.

TUGAS

1. Tugas 1

Source code

```
package main

import "fmt"

func main() {

    var beli, diskon int

    var hasil int


    fmt.Println("Masukkan harga beli:")

    fmt.Scan(&beli)


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

    fmt.Scan(&diskon)


    hasil = beli * diskon / 100

    hasil2 := beli - hasil


    fmt.Println("Harga setelah diskon adalah:", hasil2)

}
```

Screenshoot program

The screenshot shows a Go development environment with the following details:

- File Explorer:** Shows a project structure under "MODUL4_ALPROALMA".
 - contohsoal1: main.go
 - contohsoal2: main.go
 - contohsoal3: main.go
 - soalt:
 - main.go
 - soal1:
 - main.go
 - soal2
 - soal3
- Code Editor:** Displays the "main.go" file for the "soalt" package.

```
soalt > main.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var beli, diskon int
7     var hasil int
8
9     fmt.Println("Masukkan harga beli:")
10    fmt.Scan(&beli)
11
12    fmt.Print("Masukkan diskon (%): ")
13    fmt.Scan(&diskon)
14
15    hasil = beli * diskon / 100
16    hasil2 := beli - hasil
17
18    fmt.Println("Harga setelah diskon adalah:", hasil2)
19 }
```
- Terminal:** Shows the execution of the program multiple times.
 - PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\soalt\main.go"
 - Masukkan harga beli:
100000
Masukkan diskon (%): 10
Harga setelah diskon adalah: 90000
 - PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\soalt\main.go"
 - Masukkan harga beli:
200000
Masukkan diskon (%): 20
Harga setelah diskon adalah: 160000
 - PS D:\Modul4_AlproAlma> go run "d:\Modul4_AlproAlma\soalt\main.go"
 - Masukkan harga beli:
150000
Masukkan diskon (%): 15
Harga setelah diskon adalah: 127500
 - PS D:\Modul4_AlproAlma>

Deskripsi program

Program di atas adalah program sederhana dalam bahasa Go yang berfungsi untuk menghitung harga akhir suatu barang setelah diberi diskon. Di awal program, terdapat dua variabel bertipe int bernama beli dan diskon, yang masing-masing digunakan untuk menyimpan harga awal barang dan besar persentase diskon yang diberikan. Selain itu, ada juga variabel hasil yang digunakan untuk menyimpan nilai potongan harga dari hasil perhitungan diskon tersebut.

Saat program dijalankan, pengguna akan diminta untuk memasukkan harga beli barang melalui perintah `fmt.Println("Masukkan harga beli:")` dan nilainya disimpan ke dalam variabel beli menggunakan `fmt.Scan(&beli)`. Setelah itu, program meminta pengguna memasukkan besar diskon dalam persen, misalnya 10 untuk 10%, yang kemudian disimpan dalam variabel diskon.

Setelah kedua nilai tersebut diperoleh, program menghitung besarnya potongan harga dengan rumus $hasil = beli * diskon / 100$. Artinya, harga beli dikalikan dengan persentase diskon, lalu dibagi 100 untuk mendapatkan jumlah potongan dalam satuan rupiah. Nilai potongan ini kemudian digunakan untuk menghitung harga akhir setelah diskon, yaitu $hasil2 := beli - hasil$. Jadi, harga awal dikurangi dengan jumlah potongan diskon menghasilkan harga yang harus dibayar.

Terakhir, program menampilkan hasilnya ke layar dengan perintah `fmt.Println("Harga setelah diskon adalah:", hasil2)`. Misalnya, jika pengguna memasukkan harga beli 200000 dan diskon 10, maka program akan menampilkan hasil 180000 sebagai harga setelah diskon. Dengan kata lain, program ini membantu pengguna mengetahui berapa harga yang perlu dibayar setelah potongan harga diterapkan.

2. Tugas 2

Source code

```

package main

import "fmt"

func main() {

    var bmi, tinggi float64

    var berat float64


    fmt.Print("Masukkan BMI: ")

    fmt.Scan(&bmi)


    fmt.Print("Masukkan tinggi (meter): ")

    fmt.Scan(&tinggi)


    berat = bmi * (tinggi * tinggi)

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

```

Screenshot program

The screenshot shows a Go development environment with the following details:

- File Explorer:** Shows a project structure named "MODUL4_ALPROALMA" containing several files: contohsoal1, contohsoal2, contohsoal3, main.go, soal1, soal2, and soal3.
- Code Editor:** Displays the content of the "main.go" file, which contains the provided Go code for calculating BMI.
- Terminal:** Shows the command-line interface output of running the program three times with different inputs. The first run uses BMI 22.85 and height 1.75m. The second run uses BMI 23.43 and height 1.6m. The third run uses BMI 24.69 and height 1.8m.
- Status Bar:** Provides information about the current file (ln 18, Col 1), tab size (Tab Size: 4), and system status (12.51 GHz, 1528, 10/10/2025).

Deskripsi program

Program di atas adalah program sederhana dalam bahasa Go yang digunakan untuk menghitung perkiraan berat badan seseorang berdasarkan nilai BMI (Body Mass Index) dan tinggi badannya. Di awal program, terdapat tiga variabel bertipe float64

yaitu bmi, tinggi, dan berat. Tipe data float64 dipilih karena nilai-nilai seperti tinggi badan dan BMI biasanya mengandung angka desimal, misalnya 1.75 meter atau 22.5. Ketika program dijalankan, pengguna akan diminta untuk memasukkan nilai BMI terlebih dahulu melalui perintah `fmt.Print("Masukkan BMI: ")`, kemudian nilainya disimpan dalam variabel `bmi` menggunakan `fmt.Scan(&bmi)`. Setelah itu, program meminta pengguna untuk memasukkan tinggi badan dalam satuan meter melalui `fmt.Print("Masukkan tinggi (meter): ")`, dan nilainya disimpan ke dalam variabel `tinggi`. Setelah kedua nilai tersebut diperoleh, program menghitung perkiraan berat badan dengan menggunakan rumus $\text{berat} = \text{BMI} \times (\text{tinggi} \times \text{tinggi})$. Rumus ini berasal dari rumus dasar perhitungan BMI, yaitu $\text{BMI} = \text{berat} / (\text{tinggi} \times \text{tinggi})$. Dalam program ini, rumus tersebut dibalik untuk mencari berat badan berdasarkan nilai BMI yang sudah diketahui.

Hasil perhitungan kemudian ditampilkan ke layar menggunakan perintah `fmt.Printf("Perkiraan berat badan: %.0f kg\n", berat)`. Format `%.0f` digunakan agar hasil ditampilkan tanpa angka di belakang koma, sehingga hasilnya lebih sederhana dibaca. Misalnya, jika pengguna memasukkan BMI 22 dan tinggi 1.7, maka program akan menghitung berat badan sebagai $22 \times (1.7 \times 1.7)$, yaitu sekitar 64 kg. Dengan demikian, program ini membantu memperkirakan berat badan ideal seseorang berdasarkan nilai BMI dan tinggi badan yang dimasukkan.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

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

    fmt.Print("Masukkan nilai x1 dan y1: ")
    fmt.Scan(&x1, &y1)

    fmt.Print("Masukkan nilai x2 dan y2: ")
    fmt.Scan(&x2, &y2)

    fmt.Print("Masukkan nilai x3 dan y3: ")
    fmt.Scan(&x3, &y3)

    // Menghitung panjang sisi-sisi segitiga
    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))

    // Menentukan sisi terpanjang
    max := AB
    nama := "AB"

    if BC > max {
        max = BC
    }

    if CA > max {
        max = CA
    }

    if max == AB {
        nama = "AB"
    } else if max == BC {
        nama = "BC"
    } else if max == CA {
        nama = "CA"
    }
}
```

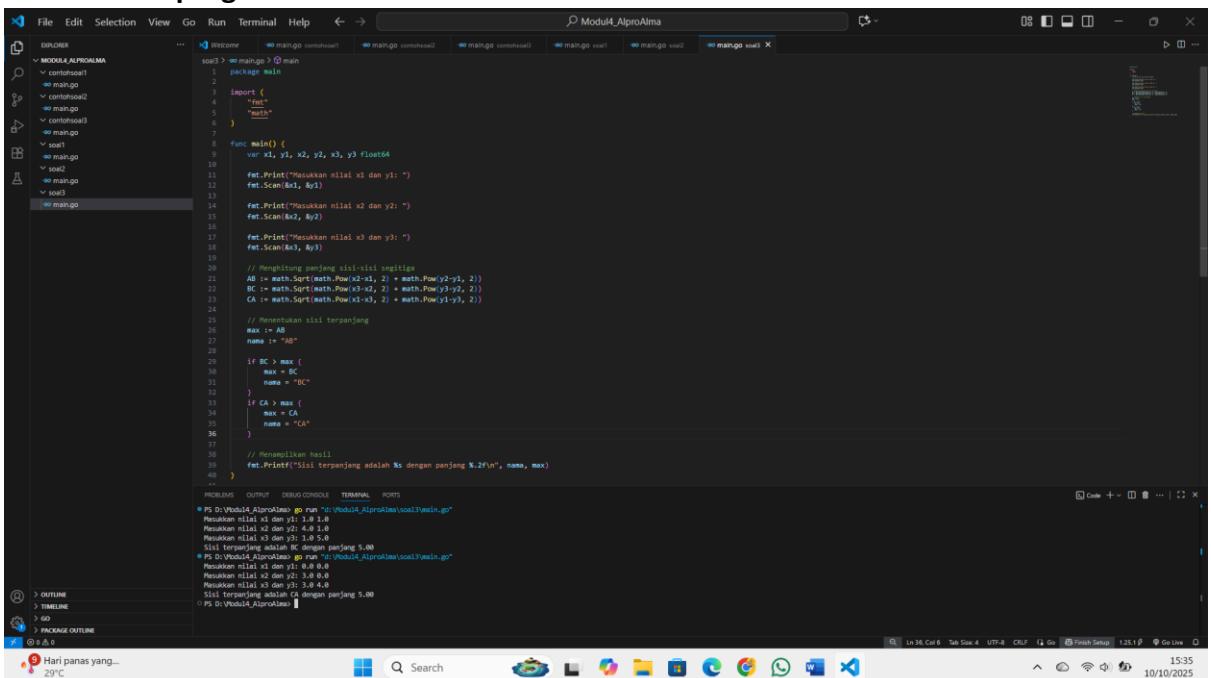
```

        nama = "BC"
    }
    if CA > max {
        max = CA
        nama = "CA"
    }

    // Menampilkan hasil
    fmt.Printf("Sisi terpanjang adalah %s dengan panjang
%.2f\n", nama, max)
}

```

Screenshoot program



Deskripsi program

Program di atas adalah program dalam bahasa Go yang digunakan untuk menentukan sisi terpanjang dari sebuah segitiga berdasarkan tiga titik koordinat yang dimasukkan pengguna. Di awal program, terdapat enam variabel bertipe float64, yaitu x_1 , y_1 , x_2 , y_2 , x_3 , dan y_3 , yang masing-masing digunakan untuk menyimpan koordinat titik A, B, dan C. Tipe float64 digunakan karena nilai koordinat bisa berupa angka desimal.

Ketika program dijalankan, pengguna diminta untuk memasukkan nilai koordinat titik pertama (x_1, y_1), kemudian titik kedua (x_2, y_2), dan terakhir titik ketiga (x_3, y_3). Nilai-nilai ini dibaca menggunakan perintah `fmt.Scan` dan disimpan ke dalam variabel yang sesuai. Setelah semua nilai koordinat dimasukkan, program mulai menghitung panjang ketiga sisi segitiga, yaitu AB, BC, dan CA.

Perhitungan panjang sisi dilakukan dengan menggunakan rumus jarak antar dua titik pada bidang koordinat, yaitu $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Dalam program, rumus ini ditulis menggunakan fungsi `math.Sqrt` untuk menghitung akar kuadrat dan `math.Pow` untuk

menghitung pangkat dua. Dengan cara ini, program memperoleh nilai panjang dari setiap sisi segitiga.

Setelah mendapatkan panjang ketiga sisi, program kemudian membandingkan ketiganya untuk menentukan sisi mana yang paling panjang. Awalnya, variabel max diisi dengan nilai panjang sisi AB, dan variabel nama diberi nilai "AB". Program kemudian memeriksa apakah panjang sisi BC atau CA lebih besar dari nilai max. Jika iya, maka nilai max dan nama diperbarui sesuai sisi yang lebih panjang.

Terakhir, program menampilkan hasil ke layar dengan perintah `fmt.Println`, yang menuliskan sisi terpanjang beserta panjangnya dengan dua angka di belakang koma. Misalnya, jika pengguna memasukkan tiga titik yang membentuk segitiga dengan sisi AB sepanjang 5.00, BC sepanjang 6.32, dan CA sepanjang 4.47, maka program akan menampilkan pesan "Sisi terpanjang adalah BC dengan panjang 6.32". Dengan demikian, program ini membantu pengguna mengetahui sisi mana yang paling panjang dari segitiga yang dibentuk oleh tiga titik koordinat yang diberikan.

TUGAS PENDAHULUAN

Soal 1

Pemberi Soal: Johanson Leeroy - 109082500017

Deskripsi Soal:

Buatlah sebuah program dalam bahasa Go yang dapat menghitung rata-rata dari 5 bilangan bulat. Program harus menerima 5 masukan berupa bilangan bulat positif.

Contoh Input/Output:

Input	Output
10 20 30 40 50	30
79 89 67 90 97	84.4
34 34 34 34 34	34

Source code

```
package main

import "fmt"

func main() {
    var a, b, c, d, e int
    var rata float64

    fmt.Print("Masukkan 5 bilangan bulat: ")
    fmt.Scan(&a, &b, &c, &d, &e)
```

```

rata = float64 (a+b+c+d+e) / 5

fmt.Printf("Rata-rata = %.1f\n", rata)

}

```

Screenshot program

The screenshot shows a Go development environment with the following details:

- File Explorer:** Shows a project structure under "MODUL4_ALPROALMA" with files like "contohsoal1.go", "main.go", "contohsoal2.go", "soal1.go", "soal2.go", "soal3.go", "TP1_AJ/main.go", and "TP2_AJ/main.go".
- Code Editor:** Displays the content of "TP1_AJ/main.go" which contains a Go program to calculate the average of five integers.
- Terminal:** Shows command-line output for three runs of the program, each prompting for five integers and printing the average (30.0, 90.97, and 34.0 respectively).
- Status Bar:** Includes information like "Ln 16, Col 1", "Tab Size: 4", "UTF-8", "CRLF", "Go", "Finish Setup", "1.25.1", "Go Live", and the date "10/10/2025".

Soal 2

Pemberi Soal: Johanson Leeroy - 109082500017

Deskripsi Soal:

Buatlah sebuah program dalam bahasa Go yang dapat mengkonversi satuan jarak: meter menjadi kilometer. Program harus menerima 1 masukan berupa bilangan bulat positif.

Contoh Input/Output:

Input	Output
6000	6 km
8600	8.6 km
450	0.45 km

Source code

```
package main
```

```

import "fmt"

func main() {
    var meter float64

    var kilometer float64

    fmt.Print("Masukkan jarak (meter): ")

    fmt.Scan(&meter)

    kilometer = meter / 1000

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

```

Screenshot program

The screenshot shows a Go development environment with the following components:

- File Explorer:** Shows a project structure under "MODUL4_ALPROALMA" containing several "main.go" files and other Go source files.
- Code Editor:** Displays the content of "main.go" from the previous code block.
- Terminal:** Shows command-line interactions:
 - "go run "d:\Modul4_AlproAlma\TP2_AJ\main.go"" followed by "Masukkan jarak (meter): 6800" and "6,00 km".
 - "go run "d:\Modul4_AlproAlma\TP2_AJ\main.go"" followed by "Masukkan jarak (meter): 8600" and "8,60 km".
 - "go run "d:\Modul4_AlproAlma\TP2_AJ\main.go"" followed by "Masukkan jarak (meter): 450" and "0,45 km".
 - "PS D:\Modul4_AlproAlma>" prompt.
- Status Bar:** Shows system information like temperature (29°C), battery level (Sebagian cerah), and system time (10/10/2025).