

LAPORAN PRAKTIKUM

Algoritma Pemrograman

MODUL 4

I/O, TIPE DATA & VARIABEL



Disusun Oleh:

ERIC SETIAWAN

109082500197

S1IF-13-04

PROGRAM STUDI S1 INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

LATIHAN KELAS – GUIDED

1. Guided 1

Source Code

```
package main

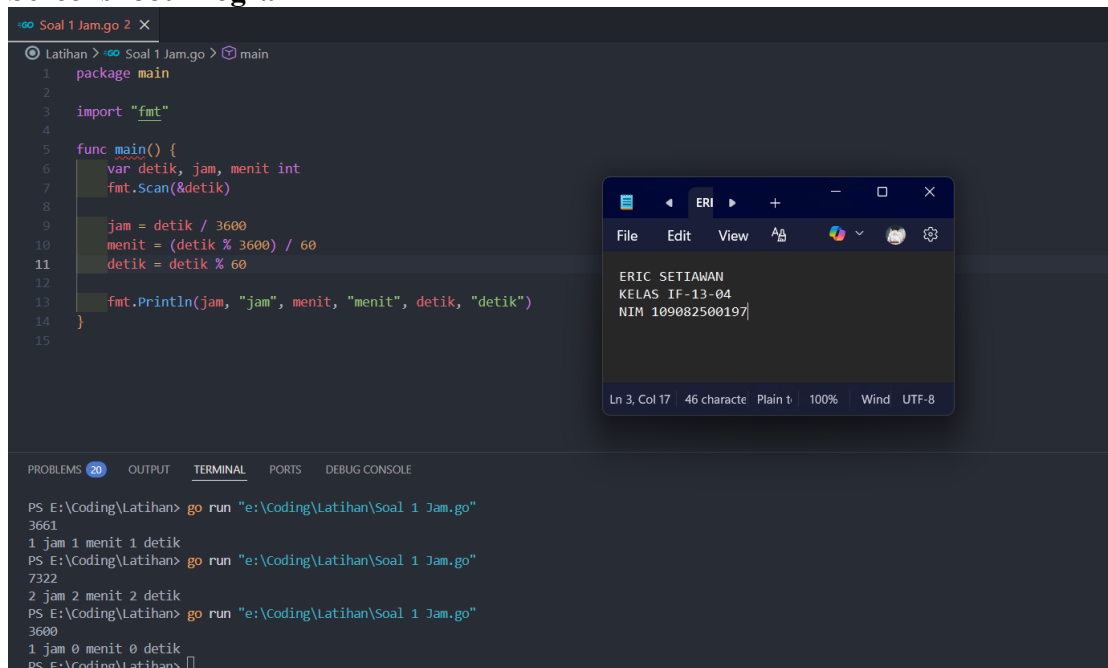
import "fmt"

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



```
Soal 1 Jam.go 2 X
Latihan > Soal 1 Jam.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var detik, jam, menit int
7     fmt.Scan(&detik)
8
9     jam = detik / 3600
10    menit = (detik % 3600) / 60
11    detik = detik % 60
12
13    fmt.Println(jam, "jam", menit, "menit", detik, "detik")
14 }
15

PROBLEMS 20 OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 1 Jam.go"
3661
1 jam 1 menit 1 detik
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 1 Jam.go"
7322
2 jam 2 menit 2 detik
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 1 Jam.go"
3600
1 jam 0 menit 0 detik
PS E:\Coding\Latihan>
```

ERIC SETIAWAN
KELAS IF-13-04
NIM 109082500197

Ln 3, Col 17 46 character Plain text 100% Window UTF-8

Deskripsi Program

Program Go ini berfungsi untuk mengonversi sebuah jumlah detik yang dimasukkan oleh pengguna menjadi format waktu yang lebih mudah dipahami, yaitu dalam bentuk jam, menit, dan detik. Hasil akhirnya akan ditampilkan di layar dengan format seperti “X jam Y menit dan Z detik”. Program ini sederhana namun sangat berguna untuk memahami cara kerja operasi pembagian dan sisa hasil bagi (modulus) dalam bahasa Go.

Program diawali dengan pernyataan `package main` dan `import "fmt"`. Baris ini menunjukkan bahwa program merupakan program utama, serta menggunakan paket `fmt` yang berfungsi untuk melakukan proses input dan output data. Selanjutnya, di dalam `func main()`, terdapat tiga variabel bertipe integer yaitu detik, jam, dan menit. Variabel detik digunakan untuk menampung nilai yang dimasukkan oleh pengguna, sedangkan jam dan menit berfungsi untuk menyimpan hasil konversi waktu dari total detik yang dimasukkan tadi.

Perintah `fmt.Scan(&detik)` berfungsi membaca nilai dari input pengguna melalui keyboard dan menyimpannya ke variabel detik. Setelah itu, dilakukan proses perhitungan untuk mengonversi detik menjadi jam dengan rumus $\text{jam} = \text{detik} / 3600$. Pembagian dengan angka 3600 dilakukan karena satu jam terdiri dari 3600 detik. Hasil pembagian ini otomatis menghasilkan bilangan bulat karena variabel bertipe `int`, sehingga sisa detik tidak ikut dihitung pada tahap ini.

Langkah berikutnya adalah menghitung menit dengan rumus $\text{menit} = (\text{detik} \% 3600) / 60$. Operasi `detik % 3600` digunakan untuk mendapatkan sisa detik setelah jam dihitung, artinya detik yang tersisa kurang dari satu jam. Kemudian sisa tersebut dibagi 60 untuk mengetahui berapa banyak menit penuh yang terkandung di dalamnya. Setelah jam dan menit dihitung, sisa detik yang belum dikonversi diperoleh dengan $\text{detik} \% 60$, yang berarti menyisakan detik kurang dari satu menit.

Terakhir, program mencetak hasil konversi ke layar menggunakan `fmt.Println(jam, "jam", menit, "menit dan", detik, "detik")`. Fungsi ini menampilkan hasil dalam format kalimat yang mudah dibaca oleh pengguna. Misalnya, jika pengguna memasukkan angka 3661, maka hasil output yang ditampilkan adalah “1 jam 1 menit dan 1 detik”. Secara keseluruhan, program ini menggambarkan bagaimana sebuah nilai numerik (total detik) dapat diolah menjadi format waktu yang lebih informatif dengan memanfaatkan operasi aritmetika dasar dalam bahasa Go.

2. Guided 2

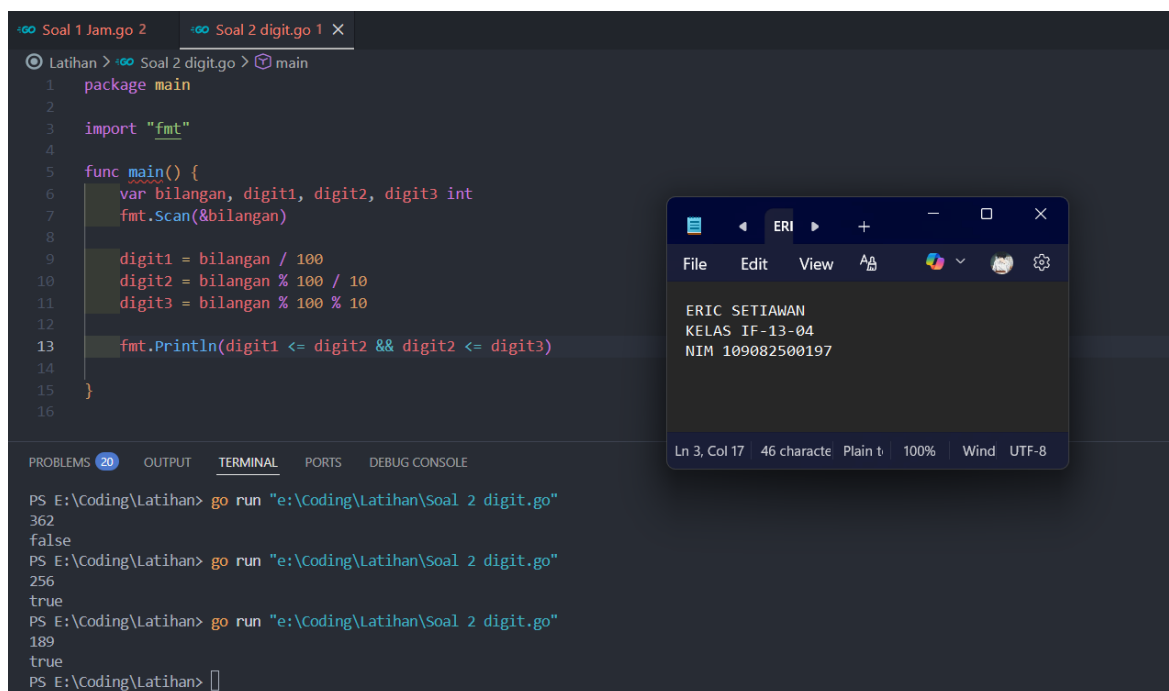
Source Code

```
package main import
"fmt" func main() {
    var bilangan, digit1, digit2, digit3 int
    fmt.Scan(&bilangan)

    digit1 = bilangan / 100
    digit2 = bilangan % 100 / 10
    digit3 = bilangan % 100 % 10

    fmt.Println(digit1 <= digit2 && digit2 <= digit3)
}
```

Screenshoot Program



```
Soal 1 Jam.go 2  Soal 2 digit.go 1 X
Latihan > Soal 2 digit.go > main
1 package main
2
3 import "fmt"
4
5 func main() {
6     var bilangan, digit1, digit2, digit3 int
7     fmt.Scan(&bilangan)
8
9     digit1 = bilangan / 100
10    digit2 = bilangan % 100 / 10
11    digit3 = bilangan % 100 % 10
12
13    fmt.Println(digit1 <= digit2 && digit2 <= digit3)
14
15 }
16

PROBLEMS 20 OUTPUT TERMINAL PORTS DEBUG CONSOLE
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 2 digit.go"
362
false
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 2 digit.go"
256
true
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 2 digit.go"
189
true
PS E:\Coding\Latihan>
```

Deskripsi Program

Program Go ini digunakan untuk memeriksa apakah susunan tiga digit dari suatu bilangan tiga angka berada dalam urutan menaik atau tidak. Artinya, program akan menentukan apakah setiap digit berikutnya bernilai sama atau lebih besar daripada digit sebelumnya. Hasil yang ditampilkan berupa nilai logika true jika urutannya menaik, dan false jika tidak.

Program diawali dengan pernyataan package main dan import "fmt", yang menunjukkan bahwa ini merupakan program utama dan menggunakan paket fmt untuk melakukan proses input dan output. Di dalam func main(), terdapat empat variabel bertipe integer, yaitu bilangan, digit1, digit2, dan digit3. Variabel bilangan berfungsi untuk menampung angka yang dimasukkan oleh pengguna, sedangkan tiga variabel lainnya digunakan untuk menyimpan masing-masing digit dari bilangan tersebut.

Ketika program dijalankan, pengguna akan diminta memasukkan satu bilangan bulat, misalnya 357. Nilai ini dibaca oleh perintah `fmt.Scan(&bilangan)` dan kemudian diolah untuk memisahkan setiap digitnya. Rumus `digit1 = bilangan / 100` digunakan untuk mengambil digit pertama (ratusan), `digit2 = bilangan % 100 / 10` untuk mendapatkan digit kedua (puluhan), dan `digit3 = bilangan % 100 % 10` untuk mengambil digit terakhir (satuan). Dengan cara ini, ketiga digit dari bilangan tersebut dapat diakses secara terpisah dan siap untuk dibandingkan.

Setelah semua digit diperoleh, program melakukan pengecekan dengan ekspresi `fmt.Println(digit1 <= digit2 && digit2 <= digit3)`. Bagian ini akan memeriksa apakah digit pertama lebih kecil atau sama dengan digit kedua, dan digit kedua lebih kecil atau sama dengan digit ketiga. Jika kedua kondisi tersebut benar, maka hasil yang ditampilkan adalah true, menandakan bahwa urutan digit bersifat menaik. Sebaliknya, jika salah satu kondisi tidak terpenuhi, maka hasilnya false. Misalnya, jika pengguna memasukkan angka 357, program akan menampilkan true, sedangkan jika yang dimasukkan adalah 531, maka hasilnya false. Program sederhana ini membantu memahami bagaimana operasi pembagian dan modulus dapat digunakan untuk memisahkan serta membandingkan digit-digit dalam sebuah bilangan.

3. Guided 3

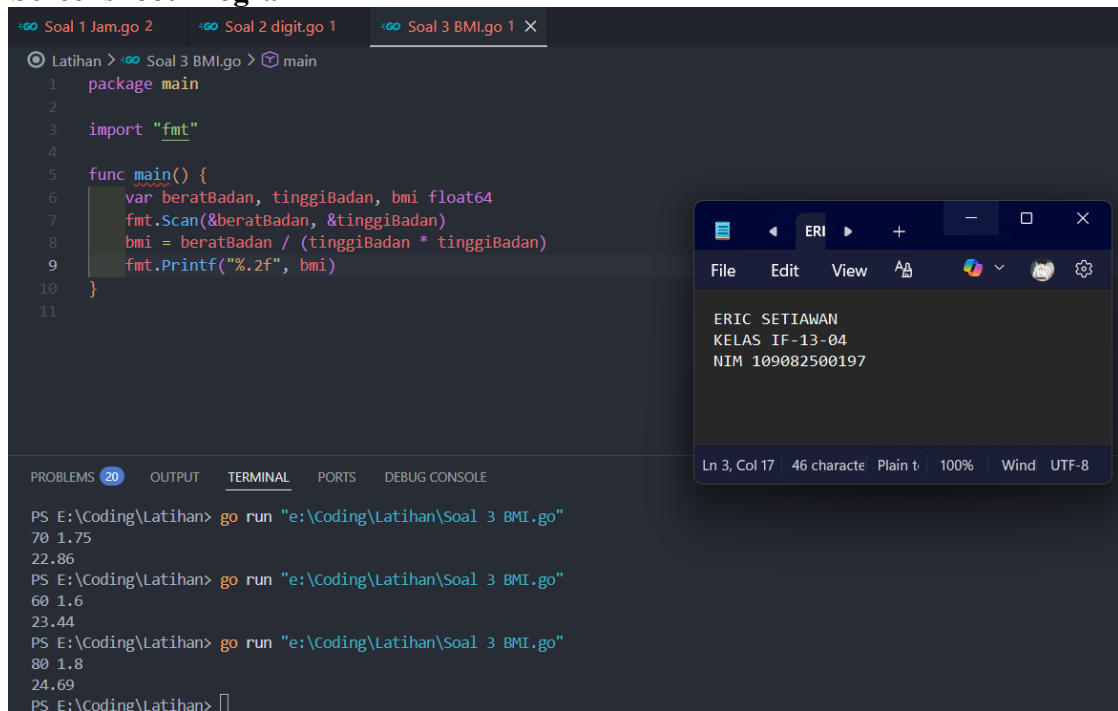
Source Code

```
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



The screenshot shows a Go IDE with three tabs: 'Soal 1 Jam.go 2', 'Soal 2 digit.go 1', and 'Soal 3 BMI.go 1'. The active tab 'Soal 3 BMI.go 1' displays the source code from the previous block. The interface includes a sidebar with 'Latihan > Soal 3 BMI.go > main', a main editor area, and a bottom panel with 'PROBLEMS (20)', 'OUTPUT', 'TERMINAL', 'PORTS', and 'DEBUG CONSOLE'. The 'TERMINAL' tab is active, showing the command 'go run "e:\Coding\Latihan\Soal 3 BMI.go"' and its output: '70 1.75', '22.86', '60 1.6', '23.44', '80 1.8', and '24.69'. A context menu is open over the code, showing options like 'File', 'Edit', 'View', and 'Aa', along with a preview of the code and a status bar indicating 'Ln 3, Col 17 | 46 character | Plain text | 100% | Window | UTF-8'.

```
package main
import "fmt"
func main() {
    var beratBadan, tinggiBadan, bmi float64
    fmt.Scan(&beratBadan, &tinggiBadan)
    bmi = beratBadan / (tinggiBadan * tinggiBadan)
    fmt.Printf("%.2f", bmi)
}
```

ERIC SETIAWAN
KELAS IF-13-04
NIM 109082500197

PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 3 BMI.go"

70 1.75
22.86
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 3 BMI.go"

60 1.6
23.44
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Soal 3 BMI.go"

80 1.8
24.69
PS E:\Coding\Latihan>

Deskripsi Program

Program Go ini digunakan untuk menghitung nilai Body Mass Index (BMI) seseorang berdasarkan berat badan dan tinggi badan yang dimasukkan oleh pengguna. Nilai BMI sendiri berfungsi untuk mengetahui apakah berat badan seseorang tergolong ideal, kurang, atau berlebih. Program ini akan menghasilkan output berupa angka desimal dengan dua digit di belakang koma yang menunjukkan hasil perhitungan BMI.

Program dimulai dengan deklarasi package main dan import "fmt", yang menandakan bahwa ini adalah program utama dan menggunakan paket fmt untuk membaca input serta menampilkan output ke layar. Di dalam func main(), terdapat tiga variabel bertipe float64, yaitu beratBadan, tinggiBadan, dan bmi. Tipe float64 digunakan karena hasil perhitungan BMI umumnya berupa bilangan desimal, bukan bilangan bulat.

Baris `fmt.Scan(&beratBadan, &tinggiBadan)` berfungsi untuk membaca dua input dari pengguna, yakni berat badan (dalam kilogram) dan tinggi badan (dalam meter). Setelah nilai kedua variabel tersebut dimasukkan, program menghitung nilai BMI dengan rumus $bmi = \text{beratBadan} / (\text{tinggiBadan} * \text{tinggiBadan})$. Rumus ini sesuai dengan rumus umum BMI, yaitu berat badan dibagi kuadrat tinggi badan.

Terakhir, hasil perhitungan ditampilkan dengan perintah `fmt.Printf("%.2f", bmi)`. Tanda `%.2f` menunjukkan bahwa nilai akan dicetak dalam format desimal dengan dua angka di belakang koma agar lebih rapi dan mudah dibaca. Sebagai contoh, jika pengguna memasukkan berat badan 70 dan tinggi badan 1.75, maka hasil yang ditampilkan adalah 22.86, yang merupakan nilai BMI orang tersebut. Program ini sederhana, namun sangat berguna untuk memahami penggunaan tipe data desimal, operasi aritmetika, dan format output dalam bahasa Go.

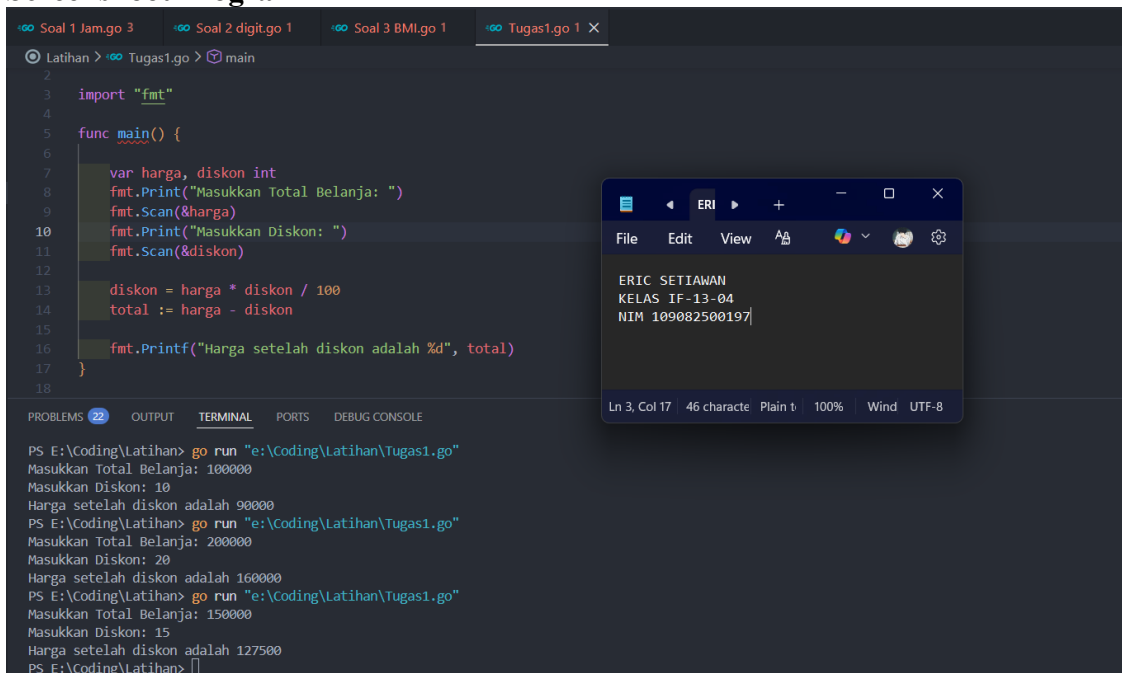
TUGAS

1. Tugas 1

Source Code

```
package main
import "fmt"
func main() {
    var harga, diskon int
    fmt.Print("Masukkan Total Belanja: ")
    fmt.Scan(&harga)
    fmt.Print("Masukkan Diskon: ")
    fmt.Scan(&diskon)
    diskon = harga * diskon / 100
    total := harga - diskon
    fmt.Printf("Harga setelah diskon adalah %d", total)
}
```

Screenshoot Program



Deskripsi Program

Program Go ini berfungsi untuk menghitung harga akhir suatu barang setelah diberikan diskon dalam bentuk persentase. Program dimulai dengan perintah `package main` dan `import "fmt"` untuk menggunakan paket input-output. Di dalam fungsi `main()`, terdapat dua variabel bertipe integer, yaitu `harga` untuk menampung total harga belanja dan `diskon` untuk menampung nilai diskon dalam persen.

Pengguna akan diminta memasukkan dua input melalui `fmt.Scan(&harga)` dan `fmt.Scan(&diskon)`. Setelah itu, program menghitung besar potongan harga dengan rumus $\text{diskon} = \text{harga} * \text{diskon} / 100$, yang berarti diskon dihitung berdasarkan persentase dari total harga. Nilai potongan tersebut kemudian dikurangkan dari harga awal dengan perintah `total := harga - diskon`, sehingga diperoleh harga akhir setelah diskon.

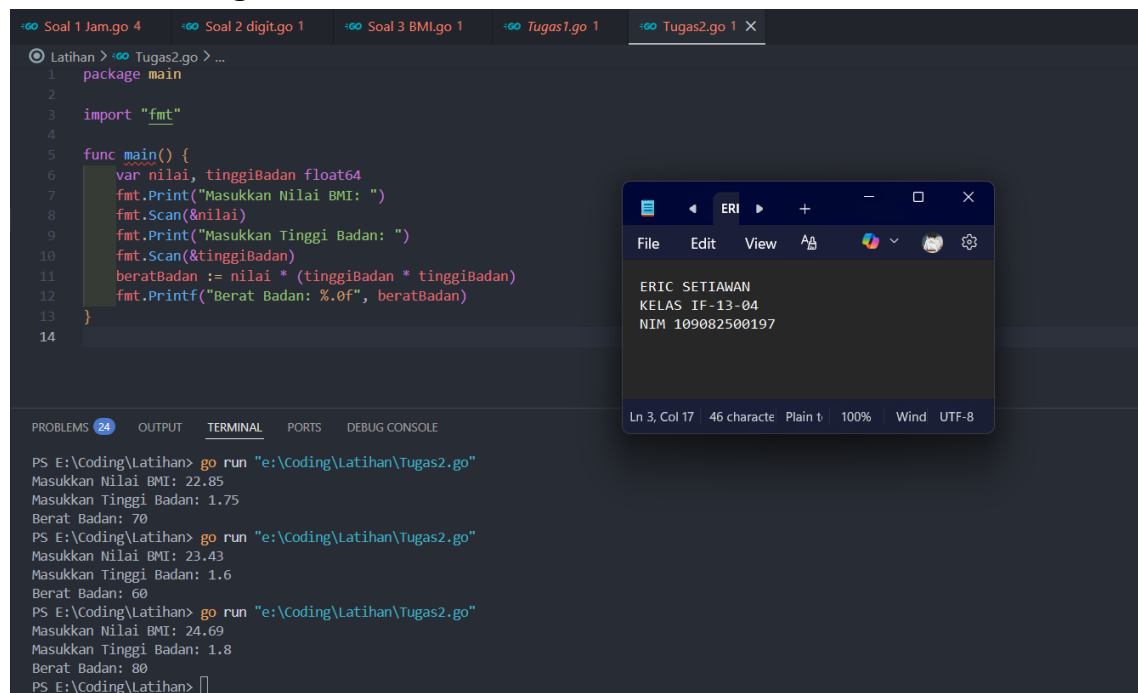
Hasil akhirnya ditampilkan ke layar menggunakan `fmt.Printf("Harga setelah diskon adalah %d", total)`. Sebagai contoh, jika total belanja adalah 200000 dan diskon 10, maka program akan menampilkan “Harga setelah diskon adalah 180000”. Program ini sederhana namun efektif untuk memahami penggunaan operasi aritmetika dan format output di Go.

2. Tugas 2

Source Code

```
package main
import "fmt"
func main() {
    var nilai, tinggiBadan float64
    fmt.Print("Masukkan Nilai BMI: ")
    fmt.Scan(&nilai)
    fmt.Print("Masukkan Tinggi Badan: ")
    fmt.Scan(&tinggiBadan)
    beratBadan := nilai * (tinggiBadan * tinggiBadan)
    fmt.Printf("Berat Badan: %.0f", beratBadan)
}
```

Screenshoot Program



```
Soal 1 Jam.go 4  Soal 2 digit.go 1  Soal 3 BMI.go 1  Tugas1.go 1  Tugas2.go 1 X
Latihan > Tugas2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var nilai, tinggiBadan float64
7     fmt.Print("Masukkan Nilai BMI: ")
8     fmt.Scan(&nilai)
9     fmt.Print("Masukkan Tinggi Badan: ")
10    fmt.Scan(&tinggiBadan)
11    beratBadan := nilai * (tinggiBadan * tinggiBadan)
12    fmt.Printf("Berat Badan: %.0f", beratBadan)
13 }
14
```

```
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Tugas2.go"
Masukkan Nilai BMI: 22.85
Masukkan Tinggi Badan: 1.75
Berat Badan: 70
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Tugas2.go"
Masukkan Nilai BMI: 23.43
Masukkan Tinggi Badan: 1.6
Berat Badan: 60
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Tugas2.go"
Masukkan Nilai BMI: 24.69
Masukkan Tinggi Badan: 1.8
Berat Badan: 80
PS E:\Coding\Latihan>
```

```
ERIC SETIAWAN
KELAS IF-13-04
NIM 109082500197
```

Deskripsi Program

Program Go ini digunakan untuk menghitung berat badan seseorang jika diketahui nilai BMI dan tinggi badan. Program ini merupakan kebalikan dari perhitungan BMI pada umumnya, di mana rumus yang digunakan adalah berat badan = BMI × (tinggi badan × tinggi badan).

Program diawali dengan deklarasi package main dan import "fmt" agar dapat menggunakan fitur input dan output. Di dalam fungsi main(), terdapat dua variabel bertipe float64, yaitu nilai untuk menampung nilai BMI dan tinggiBadan untuk menampung tinggi badan dalam satuan meter. Penggunaan tipe float64 diperlukan karena hasil perhitungan kemungkinan besar berupa angka desimal.

Melalui perintah fmt.Print, pengguna diminta untuk memasukkan nilai BMI dan tinggi badan. Nilai tersebut kemudian dibaca dengan fmt.Scan(&nilai) dan fmt.Scan(&tinggiBadan). Setelah input diperoleh, program menghitung berat badan menggunakan rumus beratBadan := nilai * (tinggiBadan * tinggiBadan). Nilai beratBadan kemudian menampung hasil akhir dari perhitungan tersebut.

Terakhir, hasilnya ditampilkan menggunakan fmt.Printf("Berat Badan: %.0f", beratBadan). Format %.0f digunakan agar angka ditampilkan tanpa angka di belakang koma, karena berat badan umumnya ditulis sebagai bilangan bulat. Sebagai contoh, jika nilai BMI yang dimasukkan adalah 22 dan tinggi badan 1.7, maka program akan menampilkan "Berat Badan: 64". Program ini sederhana, ringkas, dan bermanfaat untuk memahami penggunaan rumus matematika dasar serta format output pada bahasa Go.

3. Tugas 3

Source code

```
package main

import (
    "fmt"
    "math"
)

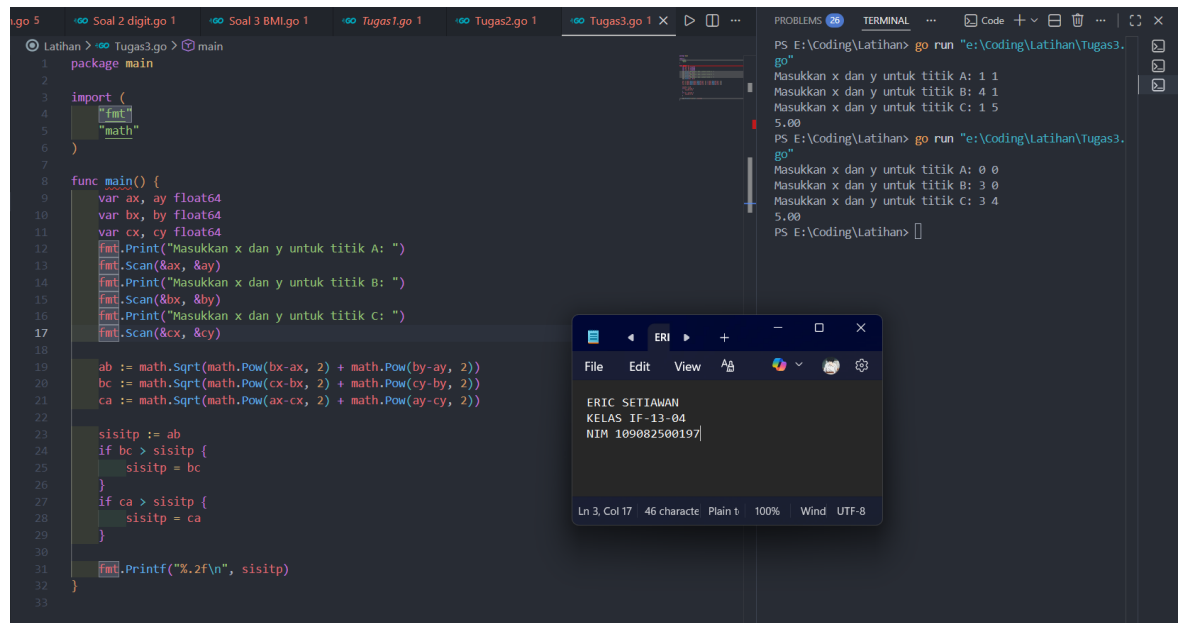
func main() {
    var ax, ay float64
    var bx, by float64
    var cx, cy float64
    fmt.Print("Masukkan x dan y untuk titik A: ")
    fmt.Scan(&ax, &ay)
    fmt.Print("Masukkan x dan y untuk titik B: ")
    fmt.Scan(&bx, &by)
    fmt.Print("Masukkan x dan y untuk titik C: ")
    fmt.Scan(&cx, &cy)

    ab := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))
    bc := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))
    ca := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))

    sisitp := ab
    if bc > sisitp {
        sisitp = bc
    }
    if ca > sisitp {
        sisitp = ca
    }

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

Screenshoot program



The screenshot shows a Go program in VS Code and its execution output in the terminal. The program calculates the longest side of a triangle given three points A, B, and C. The terminal output shows the program running successfully and printing the longest side as 5.00.

```
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var ax, ay float64
10    var bx, by float64
11    var cx, cy float64
12    fmt.Print("Masukkan x dan y untuk titik A: ")
13    fmt.Scan(&ax, &ay)
14    fmt.Print("Masukkan x dan y untuk titik B: ")
15    fmt.Scan(&bx, &by)
16    fmt.Print("Masukkan x dan y untuk titik C: ")
17    fmt.Scan(&cx, &cy)
18
19    ab := math.Sqrt(math.Pow(bx-ax, 2) + math.Pow(by-ay, 2))
20    bc := math.Sqrt(math.Pow(cx-bx, 2) + math.Pow(cy-by, 2))
21    ca := math.Sqrt(math.Pow(ax-cx, 2) + math.Pow(ay-cy, 2))
22
23    sisipt := ab
24    if bc > sisipt {
25        sisipt = bc
26    }
27    if ca > sisipt {
28        sisipt = ca
29    }
30
31    fmt.Printf("%.2f\n", sisipt)
32 }
33
```

```
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Tugas3.go"
Masukkan x dan y untuk titik A: 1 1
Masukkan x dan y untuk titik B: 4 1
Masukkan x dan y untuk titik C: 1 5
5.00
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\Tugas3.go"
Masukkan x dan y untuk titik A: 0 0
Masukkan x dan y untuk titik B: 3 0
Masukkan x dan y untuk titik C: 3 4
5.00
PS E:\Coding\Latihan>
```

Deskripsi Program

Program Go di atas digunakan untuk menentukan panjang sisi terpanjang dari sebuah segitiga yang dibentuk oleh tiga titik koordinat A, B, dan C di bidang kartesius. Program ini menggunakan paket math untuk melakukan perhitungan akar kuadrat dan pangkat dua.

Pertama, program mendeklarasikan enam variabel bertipe float64: ax, ay, bx, by, cx, dan cy, yang masing-masing menyimpan koordinat x dan y dari titik A, B, dan C. Pengguna diminta untuk memasukkan nilai-nilai tersebut melalui fmt.Scan.

Setelah semua koordinat diperoleh, program menghitung panjang masing-masing sisi segitiga menggunakan rumus jarak antar dua titik: $Jarak = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

Perhitungannya dilakukan tiga kali untuk sisi AB, BC, dan CA dengan fungsi math.Sqrt dan math.Pow. Selanjutnya, program menentukan sisi mana yang paling panjang dengan membandingkan nilai ab, bc, dan ca. Nilai terbesar disimpan dalam variabel sisipt. Terakhir, hasilnya dicetak dengan format dua angka di belakang koma (%.2f) menggunakan fmt.Printf. Sebagai contoh, jika A(0,0), B(3,0), dan C(0,4), maka sisi terpanjangnya adalah 5.00, yang merupakan sisi miring dari segitiga siku-siku tersebut. Program ini ringkas namun menunjukkan penerapan konsep jarak antar titik, fungsi matematika, dan logika perbandingan dalam Go.

TUGAS PENDAHULUAN

Pemberi Soal: Muhamad Rafi Alfiansyah - 109082500191

1. Soal 1

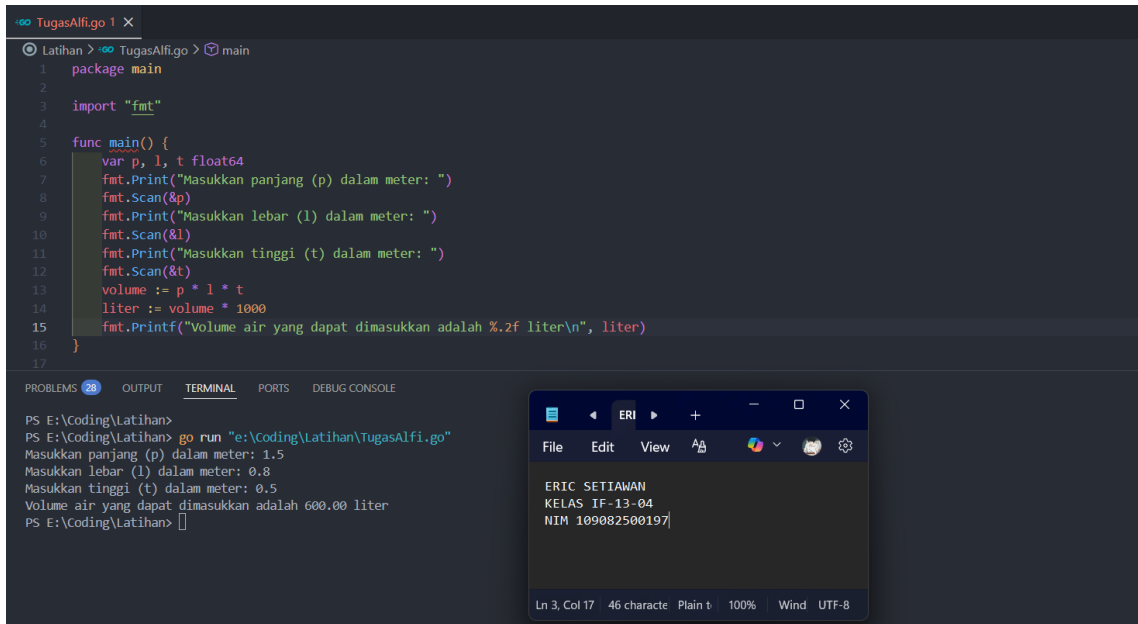
Buatlah program golang untuk menentukan berapa liter air yang dapat dimasukkan pada sebuah bak mandi berbentuk balok dengan meter (Masukkan input dari p,l,t balok lalu outputnya liter) (hasil rumus volume di ubah ke liter)

INPUT	OUTPUT
Panjang (p): 1.5 Lebar (l): 0.8 Tinggi (t): 0.5	Volume air yang dapat dimasukkan adalah 600.00 liter

Source Code

```
package main
import "fmt"
func main() {
    var p, l, t float64
    fmt.Print("Masukkan panjang (p) dalam meter: ")
    fmt.Scan(&p)
    fmt.Print("Masukkan lebar (l) dalam meter: ")
    fmt.Scan(&l)
    fmt.Print("Masukkan tinggi (t) dalam meter: ")
    fmt.Scan(&t)
    volume := p * l * t
    liter := volume * 1000
    fmt.Printf("Volume air yang dapat dimasukkan adalah %.2f liter\n", liter)
}
```

Screenshoot Program



The screenshot shows a Go program in a VS Code editor. The code defines a `main` function that prompts the user for length (`p`), width (`l`), and height (`t`) in meters. It calculates the volume in cubic meters (`m³`) and converts it to liters by multiplying by 1000. The output is formatted to two decimal places. The terminal shows the execution of the program with the inputs 1.5, 0.8, and 0.5, resulting in a volume of 600.00 liters. An inset window shows a student's ID card information.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var p, l, t float64
7     fmt.Print("Masukkan panjang (p) dalam meter: ")
8     fmt.Scan(&p)
9     fmt.Print("Masukkan lebar (l) dalam meter: ")
10    fmt.Scan(&l)
11    fmt.Print("Masukkan tinggi (t) dalam meter: ")
12    fmt.Scan(&t)
13    volume := p * l * t
14    liter := volume * 1000
15    fmt.Printf("Volume air yang dapat dimasukkan adalah %.2f liter\n", liter)
16 }
17
```

PROBLEMS 26 OUTPUT TERMINAL PORTS DEBUG CONSOLE

```
PS E:\Coding\Latihan>
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\TugasAlfi.go"
Masukkan panjang (p) dalam meter: 1.5
Masukkan lebar (l) dalam meter: 0.8
Masukkan tinggi (t) dalam meter: 0.5
Volume air yang dapat dimasukkan adalah 600.00 liter
PS E:\Coding\Latihan>
```

ERIC SETIAWAN
KELAS IF-13-04
NIM 109082500197

Ln 3, Col 17 46 character Plain text 100% Window UTF-8

Deskripsi Program

Program ini menggunakan paket **fmt** untuk input dan output.

Variabel **p**, **l**, dan **t** dideklarasikan sebagai **float64** untuk menangani nilai desimal.

Input diminta satu per satu untuk panjang, lebar, dan tinggi dalam meter.

Volume dihitung dengan rumus **p * l * t** (dalam m³).

Volume dikonversi ke liter dengan mengalikan 1000.

Output ditampilkan dengan 2 desimal

2. Soal 2

Buatlah program pada golang untuk mengetahui berapa waktu yang di butuhkan untuk menempuh suatu jarak dengan kecepatan tertentu

(Input kecepatan dan jarak,pake rumus jkw, outputnya waktu yang dibutuhkan)

INPUT	OUTPUT
Kecepatan = 60 km/jam, Jarak = 120 km	Waktu yang dibutuhkan adalah 2.00 jam

Source Code

```
package main

import "fmt"

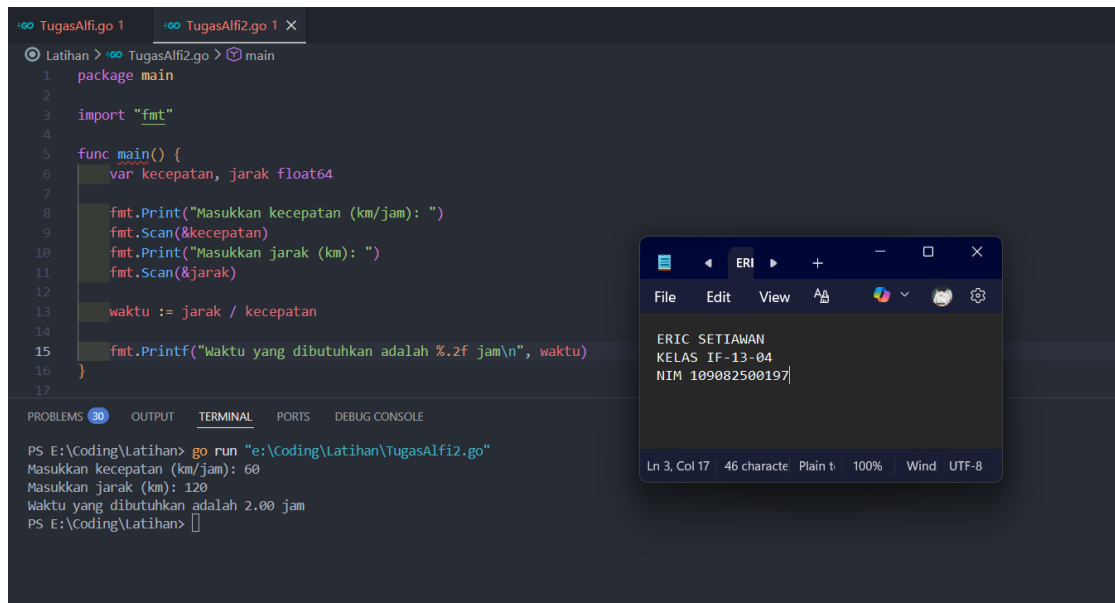
func main() {
    var kecepatan, jarak float64

    fmt.Print("Masukkan kecepatan (km/jam): ")
    fmt.Scan(&kecepatan)
    fmt.Print("Masukkan jarak (km): ")
    fmt.Scan(&jarak)

    waktu := jarak / kecepatan

    fmt.Printf("Waktu yang dibutuhkan adalah %.2f jam\n",
waktu)
}
```

Screenshoot Program



The screenshot shows a Go program in a VS Code editor and its execution output in a terminal window.

Go Program Code:

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var kecepatan, jarak float64
7
8     fmt.Print("Masukkan kecepatan (km/jam): ")
9     fmt.Scan(&kecepatan)
10    fmt.Print("Masukkan jarak (km): ")
11    fmt.Scan(&jarak)
12
13    waktu := jarak / kecepatan
14
15    fmt.Printf("Waktu yang dibutuhkan adalah %.2f jam\n", waktu)
16 }
17
```

Terminal Output:

```
PS E:\Coding\Latihan> go run "e:\Coding\Latihan\TugasAlfi2.go"
Masukkan kecepatan (km/jam): 60
Masukkan jarak (km): 120
Waktu yang dibutuhkan adalah 2.00 jam
PS E:\Coding\Latihan>
```

The terminal window also displays the user's name and ID:

```
ERIC SETIAWAN
KELAS IF-13-04
NIM 109082500197
```

Deskripsi Program

Program ini menggunakan paket **fmt** untuk input dan output.

Variabel **kecepatan** dan **jarak** dideklarasikan sebagai **float64** untuk menangani nilai desimal.

Input diminta untuk kecepatan (dalam km/jam) dan jarak (dalam km).

Waktu dihitung dengan rumus **waktu = jarak / kecepatan** (dalam jam).

Output ditampilkan dengan 2 desimal untuk presisi.