

LAPORAN PRAKTIKUM

Algoritma Pemrograman

MODUL 4

I/O, TIPE DATA & VARIABEL



Disusun Oleh:

MUHAMAD RAFI ALFIANSYAH

109082500191

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

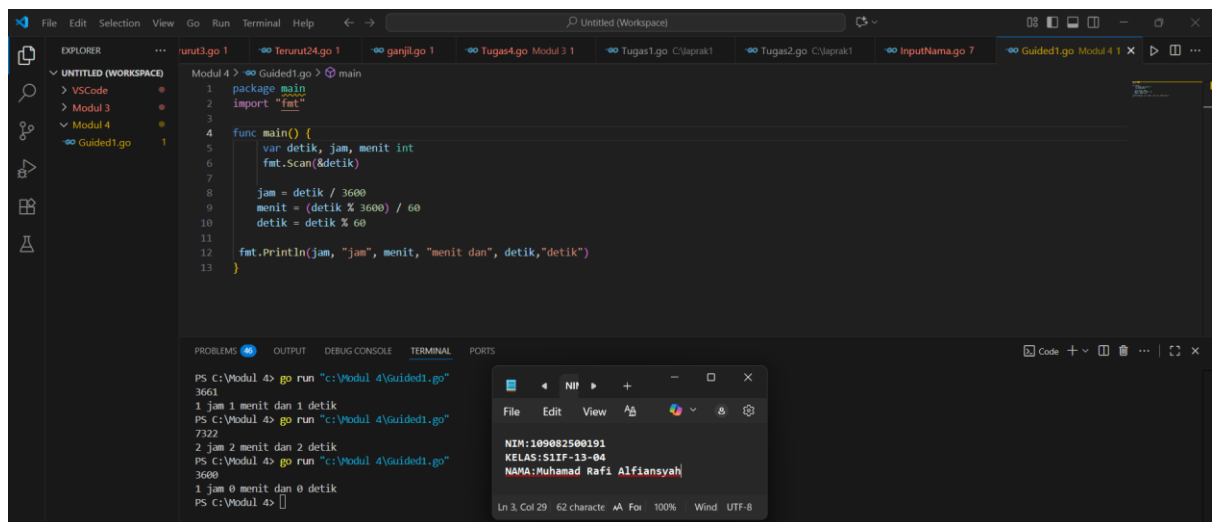
    menit = (detik % 3600) / 60

    detik = detik % 60

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

}
```

Screenshoot Program



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

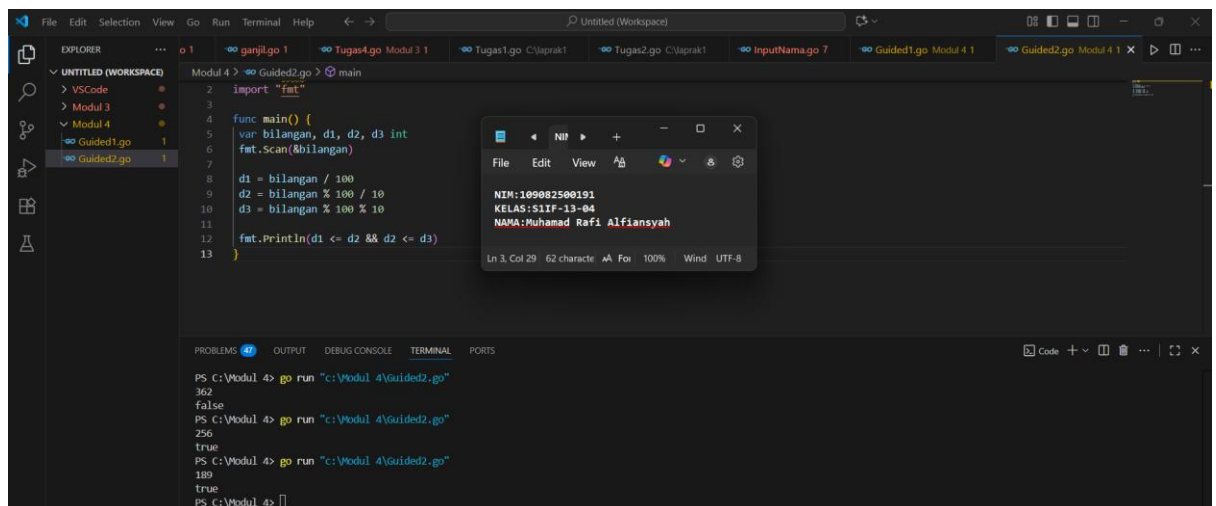
Source Code

```
package main

import "fmt"

func main() {
    var bilangan, 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

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, d1, d2, dan d3. 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 $d1 = \text{bilangan} / 100$ digunakan untuk mengambil digit pertama (ratusan), $d2 = \text{bilangan} \% 100 / 10$ untuk mendapatkan digit kedua (puluhan), dan $d3 = \text{bilangan} \% 100 \% 10$ 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(d1 <= d2 && d2 <= d3)`. 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"

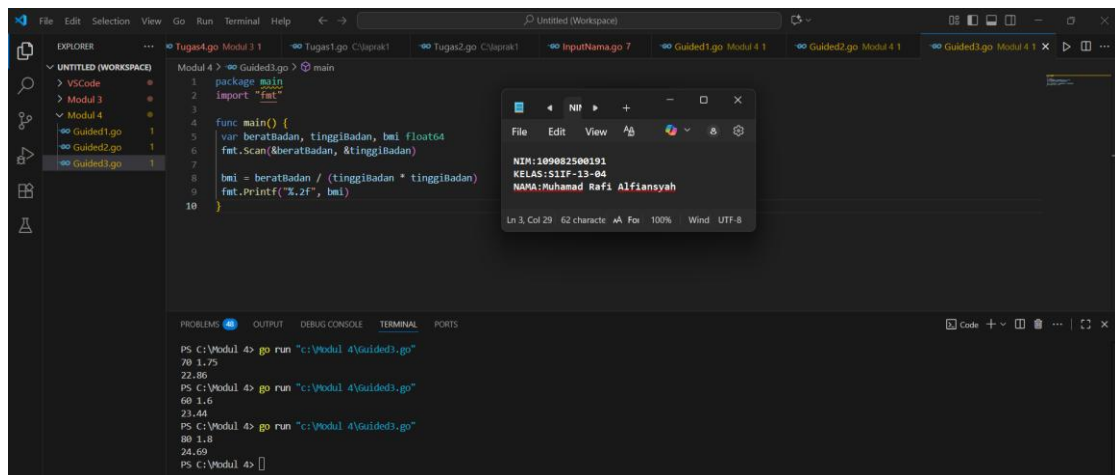
func main() {
    var beratBadan, tinggiBadan, bmi float64

    fmt.Scan(&beratBadan, &tinggiBadan)

    bmi = beratBadan / (tinggiBadan * tinggiBadan)

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

Screenshoot Program



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 h,d int

    fmt.Print("Masukkan Total Belanja: ")

    fmt.Scan(&h)

    fmt.Print("Masukkan Diskon(dalam persen): ")

    fmt.Scan(&d)

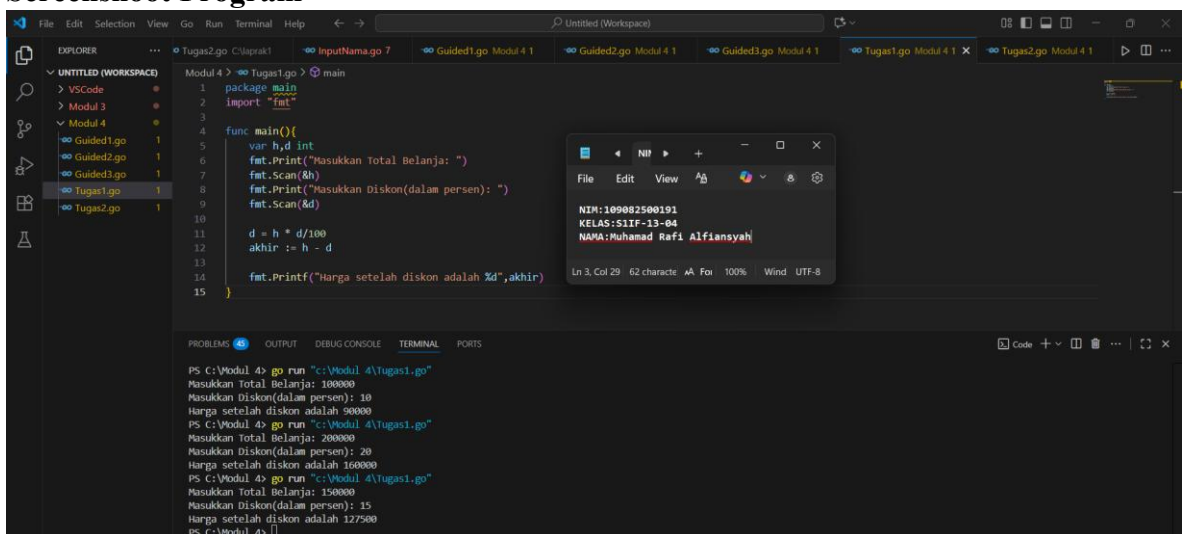

    d = h * d/100

    akhir := h - d


    fmt.Printf("Harga setelah diskon adalah %d",akhir)

}
```

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 `h` untuk menampung total harga belanja dan `d` untuk menampung nilai diskon dalam persen.

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

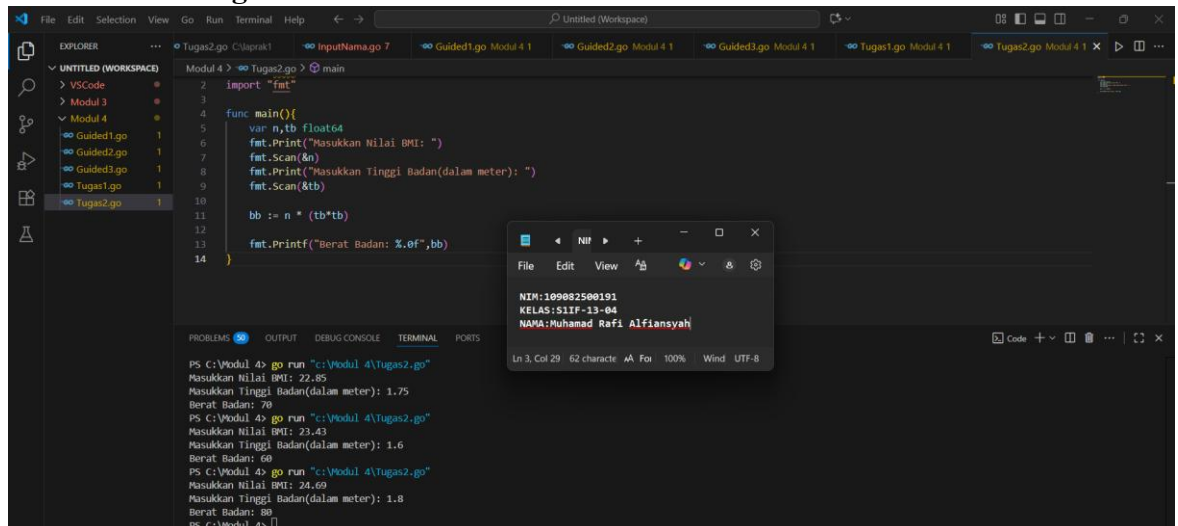
Hasil akhirnya ditampilkan ke layar menggunakan `fmt.Printf("Harga setelah diskon adalah %d", akhir)`. Sebagai contoh, jika total belanja adalah 200000 dan diskon 10, maka program akan menampilkan “Harga setelah diskon adalah 180000”. Program ini sederhana namun efektif untuk memahami penggunaan operasi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

2. Tugas 2

Source Code

```
package main
import "fmt"
func main(){
    var n,tb float64
    fmt.Print("Masukkan Nilai BMI: ")
    fmt.Scan(&n)
    fmt.Print("Masukkan Tinggi Badan(dalam meter): ")
    fmt.Scan(&tb)
    bb := n * (tb*tb)
    fmt.Printf("Berat Badan: %.0f",bb)
}
```

Screenshoot Program



```
Modul4 > Tugast2.go > main
2  import "fmt"
3
4  func main(){
5      var n,tb float64
6      fmt.Print("Masukkan Nilai BMI: ")
7      fmt.Scan(&n)
8      fmt.Print("Masukkan Tinggi Badan(dalam meter): ")
9      fmt.Scan(&tb)
10
11      bb := n * (tb*tb)
12
13      fmt.Printf("Berat Badan: %.0f",bb)
14  }
```

```
PS C:\Modul 4> go run "c:\Modul 4\tugast2.go"
Masukkan Nilai BMI: 22.85
Masukkan Tinggi Badan(dalam meter): 1.75
Berat Badan: 70
PS C:\Modul 4> go run "c:\Modul 4\tugast2.go"
Masukkan Nilai BMI: 23.43
Masukkan Tinggi Badan(dalam meter): 1.6
Berat Badan: 60
PS C:\Modul 4> go run "c:\Modul 4\tugast2.go"
Masukkan Nilai BMI: 24.69
Masukkan Tinggi Badan(dalam meter): 1.8
Berat Badan: 80
PS C:\Modul 4> []
```

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 n untuk menampung nilai BMI dan tb 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(&n) dan fmt.Scan(&tb). Setelah input diperoleh, program menghitung berat badan menggunakan rumus $bb := n * (tb * tb)$. Nilai bb kemudian menampung hasil akhir dari perhitungan tersebut.

Terakhir, hasilnya ditampilkan menggunakan fmt.Printf("Berat Badan: %.0f", bb). 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

```
Modul 4 > Tugas3.go Modul 4.1
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)
}
```

NIM: 109082500191
KELAS: S1IP-13-04
NAMA: Muhamad Rafi Alfiansyah

PS C:\Modul 4> go run "c:\Modul 4\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 C:\Modul 4> go run "c:\Modul 4\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 C:\Modul 4>

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:

$$\text{Jarak} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

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 sisitp. 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

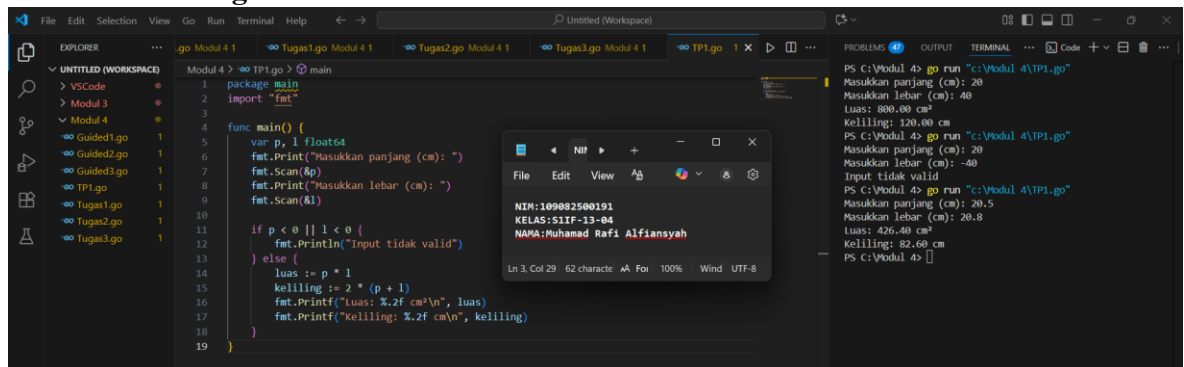
1. Soal 1

Buatlah program Go yang meminta input panjang dan lebar persegi panjang dalam satuan cm. Hitung Luas dan Keliling. Tampilkan hasil Luas dan Keliling dengan format desimal 2 digit setelah koma. Gunakan paket "fmt" untuk input/output. Jika input negatif, tampilkan pesan "Input tidak valid".

Source Code

```
package main
import "fmt"
func main() {
    var p, l float64
    fmt.Print("Masukkan panjang (cm): ")
    fmt.Scan(&p)
    fmt.Print("Masukkan lebar (cm): ")
    fmt.Scan(&l)
    if p < 0 || l < 0 {
        fmt.Println("Input tidak valid")
    } else {
        luas := p * l
        keliling := 2 * (p + l)
        fmt.Printf("Luas: %.2f cm²\n", luas)
        fmt.Printf("Keliling: %.2f cm\n", keliling)
    }
}
```

Screenshoot Program



Deskripsi Program

Program Go ini digunakan untuk menghitung luas dan keliling persegi panjang berdasarkan input panjang dan lebar dari pengguna. Program diawali dengan deklarasi package main dan import "fmt", yang menunjukkan bahwa ini adalah program utama dan menggunakan paket fmt untuk melakukan input dan output data di layar.

Di dalam fungsi main(), terdapat dua variabel p dan l bertipe float64, masing-masing untuk menampung nilai panjang dan lebar dalam satuan sentimeter. Program kemudian meminta pengguna untuk memasukkan kedua nilai tersebut dengan perintah fmt.Print, lalu membaca input menggunakan fmt.Scan(&p) dan fmt.Scan(&l).

Selanjutnya terdapat struktur percabangan if yang memeriksa apakah nilai panjang (p) atau lebar (l) bernilai negatif. Jika salah satu bernilai negatif, program akan menampilkan pesan "Input tidak valid", karena panjang atau lebar tidak mungkin bernilai negatif dalam konteks ukuran benda fisik.

Namun, jika input valid (tidak ada yang negatif), program akan melanjutkan perhitungan Luas yang dihitung dengan rumus $p * l$ dan Keliling yang dihitung dengan rumus $2 * (p + l)$.

Hasil kedua perhitungan tersebut ditampilkan ke layar menggunakan fmt.Printf dengan format desimal dua angka di belakang koma (%.2f) agar hasilnya terlihat rapi dan presisi. Satuan cm^2 digunakan untuk luas, sedangkan cm untuk keliling.

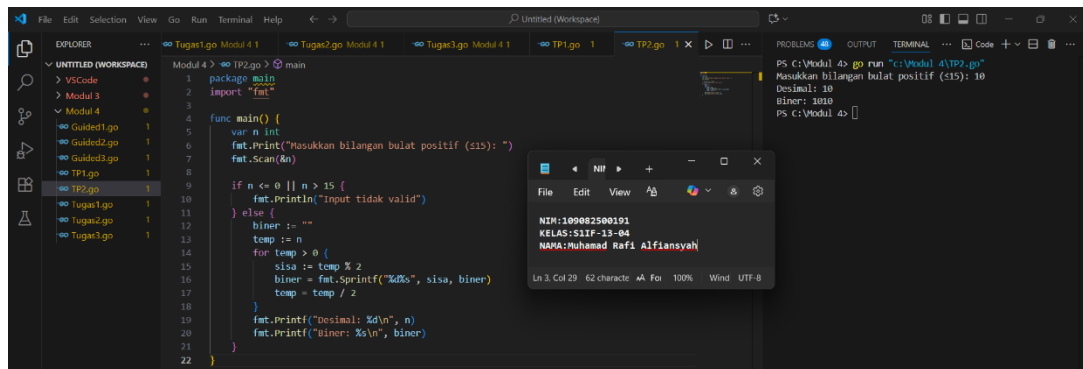
2. Soal 1

Buatlah program Go yang meminta input bilangan bulat positif. Konversikan n ke bentuk biner sederhana. Tampilkan bilangan desimal asli dan hasil biner. Jika $n \leq 0$ atau $n > 15$, tampilkan "Input tidak valid". Gunakan paket "fmt" dan loop sederhana untuk konversi.

Source Code

```
package main
import "fmt"
func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif ( $\leq 15$ ): ")
    fmt.Scan(&n)
    if n <= 0 || n > 15 {
        fmt.Println("Input tidak valid")
    } else {
        biner := ""
        temp := n
        for temp > 0 {
            sisa := temp % 2
            biner = fmt.Sprintf("%d%s", sisa, biner)
            temp = temp / 2
        }
        fmt.Printf("Desimal: %d\n", n)
        fmt.Printf("Biner: %s\n", biner)
    }
}
```

Screenshoot Program



```
1 package main
2 import "fmt"
3
4 func main() {
5     var n int
6     fmt.Print("Masukkan bilangan bulat positif (1-15): ")
7     fmt.Scan(&n)
8
9     if n <= 0 || n > 15 {
10        fmt.Println("Input tidak valid")
11    } else {
12        biner := ""
13        temp := n
14        for temp > 0 {
15            sisa := temp % 2
16            biner = fmt.Sprintf("%02s", sisa, biner)
17            temp = temp / 2
18        }
19        fmt.Printf("Desimal: %d\n", n)
20        fmt.Printf("Biner: %s\n", biner)
21    }
22 }
```

Terminal Output:

```
PS C:\Modul 4> go run "c:\Modul 4\TP2.go"
Masukkan bilangan bulat positif (1-15): 10
Desimal: 10
Biner: 1010
PS C:\Modul 4>
```

Deskripsi Program

Program Go ini berfungsi untuk mengonversi bilangan desimal kecil menjadi bentuk biner dengan cara pembagian berulang. Program dimulai dengan package main dan import "fmt" untuk memanfaatkan fungsi input-output. Di dalam main(), variabel n bertipe int digunakan untuk menampung nilai yang dimasukkan pengguna. Setelah pengguna memasukkan bilangan bulat positif maksimal 15, program memeriksa validitas input menggunakan kondisi if. Jika n kurang dari atau sama dengan nol, atau lebih dari 15, maka program menampilkan pesan "Input tidak valid".

Jika input valid, proses konversi dimulai. Variabel temp menyalin nilai n, dan variabel biner menyimpan hasil konversi dalam bentuk string. Melalui perulangan for temp > 0, program menghitung sisa pembagian temp % 2 sebagai digit biner, lalu menambahkannya di depan string biner. Nilai temp kemudian dibagi dua hingga menjadi nol. Setelah perulangan selesai, hasil konversi ditampilkan dengan mencetak nilai desimal aslinya serta bentuk binernya. Misalnya, untuk input 10, program menampilkan "Desimal: 10" dan "Biner: 1010". Program ini sederhana namun efektif untuk memperlihatkan prinsip dasar konversi bilangan dari desimal ke biner.