

**LAPORAN PRAKTIKUM**  
**Algoritma Pemrograman**

**EVALUASI**



**Disusun oleh:**

**Ismail Marasabessy**

**109082500041**

**S1IF-13-07**

**PROGRAM STUDI S1 INFORMATIKA**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2025**

## SOAL

### 1. SOAL 1 Source

#### Code

```
package main

import

"fmt"

func main()

{
    var n

    int

    for {           fmt.Println("Input ketinggian piramida
(N < 8): ")           fmt.Scan(&n)

        if n < 8

    {

        break

    }

    fmt.Println("Ketinggian harus kurang dari 8, silakan input
ulang.")

}           for i := 1; i <= n; i++

{           for j := 0; j < n-i; j++

{

        fmt.Print(" ")

    }           for k := 0; k < (2*i -
1); k++ {           fmt.Print("*")

    }
```

```
fmt.Println()  
  
}      for i := 0; i < 2; i++ {  
  
for j := 0; j < n-1; j++ {  
  
    fmt.Print(" ")  
  
}  
  
fmt.Println("|")  
  
}  
  
}
```

## Screenshot program

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under "ASSEMEN2TIPEB". The "soal1" folder contains "soal1.go", which is currently selected. The "soal2" folder contains "soal2.go".
- Editor:** Displays the content of "soal1.go". The code prints a diamond shape based on user input N. If N is greater than or equal to 8, it prints a warning message. The terminal output shows the diamond pattern for N=7.
- Terminal:** Shows the command "PS C:\Users\acer\Desktop\assesmen2tipeB>" followed by the output of the program for N=7.
- Status Bar:** Shows the file path "C:\Users\acer\Desktop\assesmen2tipeB\soal1.go", line 28, column 10, and the tab size as 8.

### **Deskripsi program**

Program diawali dengan meminta pengguna memasukkan ketinggian piramida dengan syarat nilai n harus kurang dari 8. Proses input ini dilakukan di dalam perulangan for tanpa kondisi agar pengguna dapat mengulang input jika nilai yang dimasukkan tidak sesuai. Jika nilai n kurang dari 8, perulangan akan dihentikan menggunakan break. Namun, jika nilai yang dimasukkan sama dengan atau lebih dari 8, program akan menampilkan pesan kesalahan dan meminta pengguna untuk menginput ulang. Setelah mendapatkan nilai ketinggian yang valid, program mencetak piramida menggunakan perulangan bersarang. Perulangan pertama digunakan untuk mengatur baris piramida. Di dalamnya terdapat perulangan untuk mencetak spasi sebanyak  $n - i$  agar piramida berada di tengah, serta perulangan lainnya untuk mencetak simbol bintang (\*) sebanyak  $2*i - 1$  sehingga membentuk pola piramida. Setelah bagian piramida selesai dicetak, program mencetak dua baris simbol | yang berfungsi sebagai batang piramida. Jumlah spasi sebelum simbol tersebut diatur agar posisi batang sejajar dengan bagian tengah piramida.

## **2. SOAL 2 Source**

### **Code**

```
package main

import
"fmt"

func main() {
    var
    pilihan, jumlah int
    var
    harga, total int
    var
    namaProduk string

    fmt.Println(" --- DAFTAR PRODUK TOKO BUDI -
--")
    fmt.Println("1. Little Trees      -"
Rp35.000)
    fmt.Println("2. Lap Microfiber   -"
Rp25.000)
    fmt.Println("3. Cover Steer        -"
Rp150.000)
    fmt.Println("4. Sponge Cuci Mobil  -"
Rp10.000)
}
```

```
        fmt.Println("Pilih produk (1-4):  
")      fmt.Scan(&pilihan)  
  
        fmt.Print("Masukkan jumlah beli: ")  
  
        fmt.Scan(&jumlah)  
  
        switch  
  
        pilihan {      case  
  
        1:  
  
            namaProduk = "Little Trees"  
  
            harga = 35000      case 2:  
  
            namaProduk = "Lap Microfiber"  
  
            harga = 25000      case 3:  
  
            namaProduk = "Cover Steer"  
  
            harga = 150000     case 4:  
  
            namaProduk = "Sponge Cuci Mobil"  
  
            harga = 10000      default:  
  
                fmt.Println("\nMaaf, pilihan produk tidak ada!")  
  
            return  
  
        }      total = harga *  
  
jumlah
```

```
        fmt.Println("\n--- STRUK PEMBAYARAN ---")  
        fmt.Printf("Produk : %s\n", namaProduk)
```

```
fmt.Printf("Harga : Rp %d\n", harga)

fmt.Printf("Jumlah : %d\n", jumlah)      fmt.Printf("Total

: Rp %d\n", total)

}
```

## Screenshot program

The screenshot shows a Visual Studio Code interface with several windows open. On the left is the Explorer sidebar showing a project structure under 'ASSESMENT2TPEB' with files 'soal1.go' and 'soal2.go'. The main editor window displays a Go program named 'soal2.go' with code that prints a menu of products and their prices, asks for a selection, and then asks for a quantity. The terminal window on the right shows the execution of the program, starting with the user's input 'ismail marasabessy 109082500113'. The program then lists four products: Little Trees (Rp.35.000), Lap microfiber (Rp.25.000), Cover Steer (Rp.150.000), and Sponge cuci mobil (Rp.10.000). It prompts the user to 'Pilih menu (1-4):' and receives the input '1'. It then asks for the quantity with 'Masukkan jumlah porsinya:' and receives the input '5'. Finally, it prints a receipt titled 'STRUK PEMBELIAN' with the details: Produk : Little Trees, Jumlah : 1, Total : 175000.

```
File Edit Selection View Go Run ... < > assesmen2tipeB

EXPLORER
ASSESMEN2TPEB
  soal1
    soal1.go
  soal2
    soal2.go

File Edit View AA & ismail marasabessy
109082500113
Ln 2, Col 13 | 31 character Plain t 100% Wind UTF-8

package main
import "fmt"
func main() {
    var a, b int
    fmt.Println("== Daftar Produk Toko Budi ==")
    fmt.Println("1. Little Trees - Rp.35.000")
    fmt.Println("2. Lap microfiber - Rp.25.000")
    fmt.Println("3. Cover Steer - Rp.150.000")
    fmt.Println("4. Sponge cuci mobil - Rp.10.000")
    fmt.Print("Pilih menu (1-4): ")
    fmt.Scan(&a)
    fmt.Print("Masukkan jumlah porsinya: ")
    fmt.Scan(&b)
}
Ln 2, Col 13 | 31 character Plain t 100% Wind UTF-8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code + < > ... | < > x
--- Daftar Produk Toko Budi ---
1. Little Trees - Rp.35.000
2. Lap microfiber - Rp.25.000
3. Cover Steer - Rp.150.000
4. Sponge cuci mobil - Rp.10.000
Pilih menu (1-4): 1
Masukkan jumlah porsinya: 5
--- STRUK PEMBELIAN ---
Produk : Little Trees
Jumlah : 1
Total : 175000
PS C:\Users\acer\Desktop\assesmen2tipeB>
Ln 17, Col 40 Tab Size
```

### **Deskripsi program**

Variabel pilihan digunakan untuk menyimpan nomor produk yang dipilih oleh pengguna, variabel jumlah digunakan untuk menyimpan jumlah barang yang dibeli, variabel harga digunakan untuk menyimpan harga satuan produk, variabel total digunakan untuk menyimpan total harga yang harus dibayar, dan variabel namaProduk digunakan untuk menyimpan nama produk yang dipilih.

Program diawali dengan menampilkan daftar produk beserta harga masing-masing. Setelah itu, pengguna diminta memasukkan nomor produk dan jumlah pembelian. Nilai input tersebut dibaca menggunakan fmt.Scan dan disimpan ke dalam variabel yang sesuai. Selanjutnya, program menggunakan struktur switch case untuk menentukan nama produk dan harga berdasarkan pilihan pengguna. Jika pilihan tidak sesuai dengan daftar yang tersedia, program akan menampilkan pesan bahwa produk tidak ada dan program dihentikan. Jika pilihan valid, program menghitung total pembayaran dengan mengalikan harga produk dengan jumlah pembelian. Terakhir, program menampilkan struk pembayaran yang berisi nama produk, harga satuan, jumlah pembelian, dan total harga yang harus dibayar.

### **3. SOAL 3 Source**

#### **Code**

```
package main

import

"fmt"

func main() {

    var a, b, c int

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

    if a+b <= c || a+c <= b || b+c <= a
    {
        fmt.Println("Bukan segitiga")
    } else if a == b && b == c {
        fmt.Println("Segitiga sama sisi")
```

```

        } else if a*a+b*b == c*c || a*a+c*c == b*b || b*b+c*c ==
        a*a {             fmt.Println("Segitiga siku-siku") } else
        if a == b || a == c || b == c {
            fmt.Println("Segitiga sama kaki")
        } else {
            fmt.Println("Segitiga sembarang")
        }
    }
}

```

## Screenshot program

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface with the following details:

- Explorer View:** Shows a folder structure under "ASSESMEN2TIPEB" containing three subfolders: "soal1", "soal2", and "soal3". The file "soal3.go" is currently selected.
- Code Editor:** Displays the content of "soal3.go". The code checks if three sides of a triangle satisfy the conditions for being a right-angled triangle, an equilateral triangle, or an isosceles triangle. It uses the `fmt` package to print the results.
- Terminal:** Shows the command-line output of running the program. The terminal window title is "assesmen2tipeB". The output shows three runs of the program with different side inputs (1 3 3, 5 5 8, and 3 4 5) and their corresponding outputs ("Segitiga sama kaki", "Segitiga siku-siku", and "Segitiga sama kaki").
- Status Bar:** Shows the current file path as "C:\Users\acer\Desktop\assesmen2tipeB\soal3\soal3.go", the line number as "Ln 2, Col 13", the character count as "31 character", the encoding as "Plain text", and the file encoding as "UTF-8".

### **Deskripsi program**

Pada program ini, saya membuat sebuah program menggunakan bahasa Go untuk menentukan jenis segitiga berdasarkan tiga buah sisi yang dimasukkan oleh pengguna. Tiga variabel bertipe integer, yaitu a, b, dan c, digunakan untuk menyimpan panjang masing-masing sisi segitiga. Program dimulai dengan membaca tiga nilai sisi segitiga menggunakan `fmt.Scan` dan menyimpannya ke dalam variabel a, b, dan c. Setelah itu, program terlebih dahulu mengecek apakah ketiga sisi tersebut dapat membentuk sebuah segitiga atau tidak. Pengecekan dilakukan dengan menggunakan syarat segitiga, yaitu jumlah dua sisi harus lebih besar daripada sisi yang lainnya. Jika salah satu syarat tidak terpenuhi, program akan menampilkan keluaran "Bukan segitiga". Jika ketiga sisi memenuhi syarat segitiga, program kemudian menentukan jenis segitiga. Pertama, program mengecek apakah ketiga sisi memiliki panjang yang sama. Jika semua sisi sama, maka segitiga dikategorikan sebagai "segitiga sama sisi". Selanjutnya, program mengecek apakah segitiga tersebut merupakan segitiga siku-siku dengan menggunakan teorema Pythagoras. Jika salah satu kombinasi kuadrat sisi memenuhi, maka program menampilkan "Segitiga siku-siku". Jika segitiga bukan segitiga siku-siku, program akan mengecek apakah terdapat dua sisi yang sama panjang. Jika kondisi ini terpenuhi, maka segitiga dikategorikan sebagai "segitiga sama kaki". Apabila seluruh kondisi tersebut tidak terpenuhi, program akan menampilkan "Segitiga sembarang" sebagai hasil akhir.