

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

**EVALUASI**



**Disusun oleh:**

**M MAHDAN ARGYA SYARIF**

**109082500059**

**S1IF-13-07**

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

## SOAL

### SOAL 1

#### Source Code

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var n int
    fmt.Scan(&n)

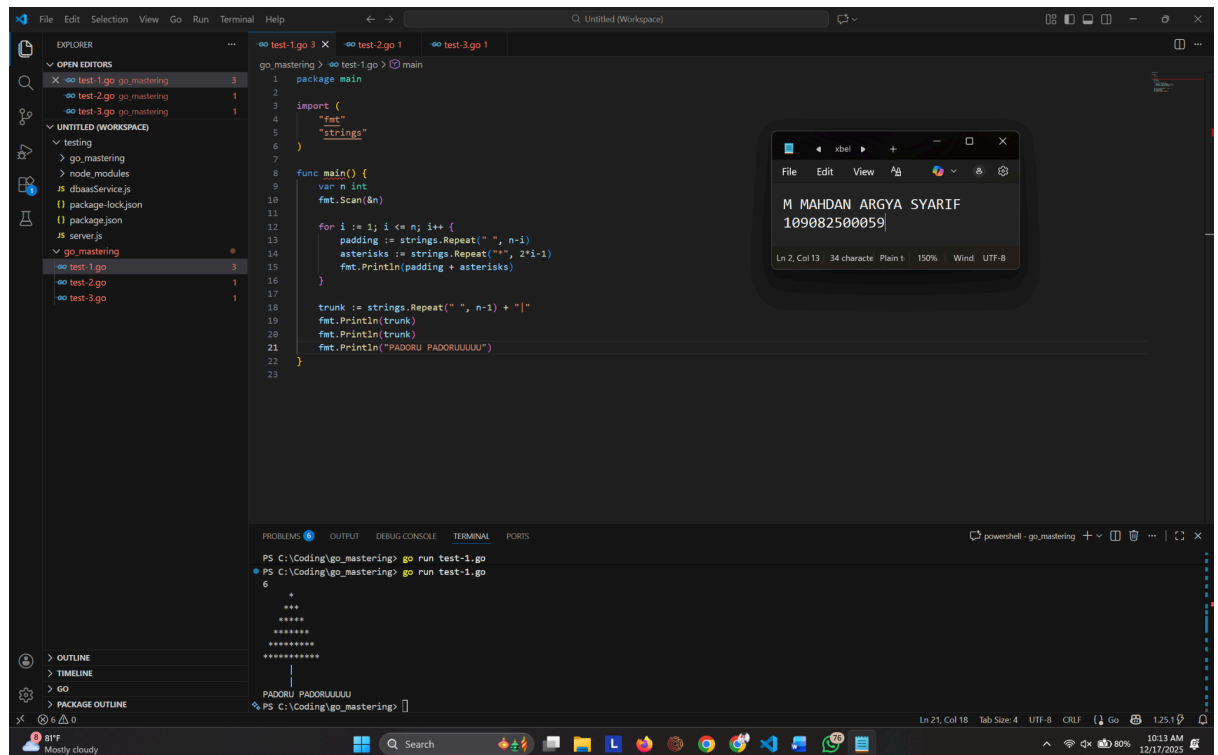
    for i := 1; i <= n; i++ {
        padding := strings.Repeat(" ", n-i)
        asterisks := strings.Repeat("*", 2*i-1)
        fmt.Println(padding + asterisks)
    }

    trunk := strings.Repeat(" ", n-1) + "|"
    fmt.Println(trunk)
    fmt.Println(trunk)
    fmt.Println("PADORU PADORUUU")
}
```

## Screenshoot program

//tambahkan tangkapan layar dari program (boleh lebih dari 1 jika diperlukan)

CONTOH TANGKAPAN LAYAR: (GUNAKAN NOTEPAD)



## Deskripsi program

Program ini ditulis dalam bahasa pemrograman Go. Tujuan utama program adalah menerima input bilangan bulat (integer) dari pengguna yang merepresentasikan tinggi pohon, kemudian mencetak pola visual pohon Natal (segitiga sama kaki) menggunakan karakter asterisk (\*) dan batang pohon menggunakan karakter pipe (|). Program diakhiri dengan pesan string statis.

Konsep Pemrograman Kunci yang Diterapkan :

1. Algoritma Sekuensial & Iterasi: Menggunakan loop for untuk membangun pola geometris berbasis baris.
2. Manipulasi String: Penggunaan fungsi strings.Repeat menggantikan kebutuhan nested loop (loop di dalam loop), membuat kode lebih bersih (clean code) dan efisien secara komputasi ( $O(n)$  untuk ruang).
3. Pointers: Pemahaman tentang manajemen memori saat mengambil input (&n). Aritmatika Integer: Penerapan rumus matematika sederhana ( $2i-1$ ) untuk pola visual.

## SOAL 2

### Source Code

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    menus := []string{
        "Empal Gentong Biasa",
        "Empal Asem",
        "Sate Kambing Muda",
        "Nasi Lengko Pagongan",
    }

    prices := []int{22000, 22000, 40000, 15000}

    fmt.Println("=== MENU POS ===")

    for i, menu := range menus {
        fmt.Printf("%d. %s - Rp %d\n", i+1, menu, prices[i])
    }

    fmt.Println(strings.Repeat("-", 40))

    var menuChoice, xontol int

    fmt.Print("Pilih menu (1-4): ")

    fmt.Scan(&menuChoice)

    fmt.Print("Berapa porsi: ")

    fmt.Scan(&xontol)
```

```

    fmt.Println(strings.Repeat("-", 40))

    if menuChoice < 1 || menuChoice > 4 || xontol < 1 {

        fmt.Println("Go Fuck Yourself dude...")

        return

    }

    selectedMenu := menus[menuChoice-1]

    selectedPrice := prices[menuChoice-1]

    total := selectedPrice * xontol

    fmt.Println("\n" + strings.Repeat("=", 40))

    fmt.Println("Menu          : " + selectedMenu)

    fmt.Println("Harga           : Rp " + fmt.Sprintf(selectedPrice))

    fmt.Println("Porsi           : " + fmt.Sprintf(xontol))

    fmt.Println("Total            : Rp " + fmt.Sprintf(total))

    fmt.Println(strings.Repeat("=", 40))

}

```

## Screenshoot program

The screenshot displays a Go IDE with the following components:

- EXPLORER:** Shows the project structure with folders like `dbaseServices`, `package-lock.json`, `package.json`, `server.js`, and `go_mastering`.
- EDITOR:** Displays the `main.go` file with the following code:
 

```

1 package main
2
3 import (
4     "fmt"
5     "strings"
6 )
7
8 func main() {
9     menus := []string{
10         "Empal Gentong Blassa",
11         "Empal Asen",
12         "Sate Kambing Muda",
13         "Masi Lengko Pagongan",
14     }
15     prices := []int{22000, 22000, 40000, 15000}
16
17     fmt.Println("=== MENU POS ===")
18     for i, menu := range menus {
19         fmt.Printf("%d. %s - Rp %d\n", i+1, menu, prices[i])
20     }
21
22     fmt.Println(strings.Repeat("-", 40))
23     var menuChoice, xontol int
24     fmt.Print("Pilih menu (1-4): ")
25     menuChoice = fmt.Scan(&menuChoice)
26     fmt.Print("Berapa porsi: ")
27     xontol = fmt.Scan(&xontol)
28     fmt.Println(strings.Repeat("-", 40))
29     if menuChoice < 1 || menuChoice > 4 || xontol < 1 {
30         fmt.Println("Go Fuck Yourself dude...")
31         return
32     }
33
34     selectedMenu := menus[menuChoice-1]

```
- TERMINAL:** Shows the execution output:
 

```

PS C:\Coding\go_mastering> go run test-2.go
=== MENU POS ===
1. Empal Gentong Blassa - Rp 22000
2. Empal Asen - Rp 22000
3. Sate Kambing Muda - Rp 40000
4. Masi Lengko Pagongan - Rp 15000
-----
Pilih menu (1-4): 4
Berapa porsi: 0
Go Fuck Yourself dude...
PS C:\Coding\go_mastering>

```
- PROBLEMS:** Empty.
- OUTPUT:** Empty.
- DEBUG CONSOLE:** Empty.
- PORTS:** Empty.

### Deskripsi program

Tujuan dan Fungsionalitas Program ini adalah aplikasi konsol sederhana yang ditulis dalam bahasa Go untuk mensimulasikan sistem Point of Sale (POS) atau kasir rumah makan. Program ini dirancang untuk menampilkan daftar menu makanan khas Cirebon, menerima pesanan pengguna, memvalidasi ketersediaan menu, dan menghitung total harga transaksi.

### SOAL 3

#### Source Code

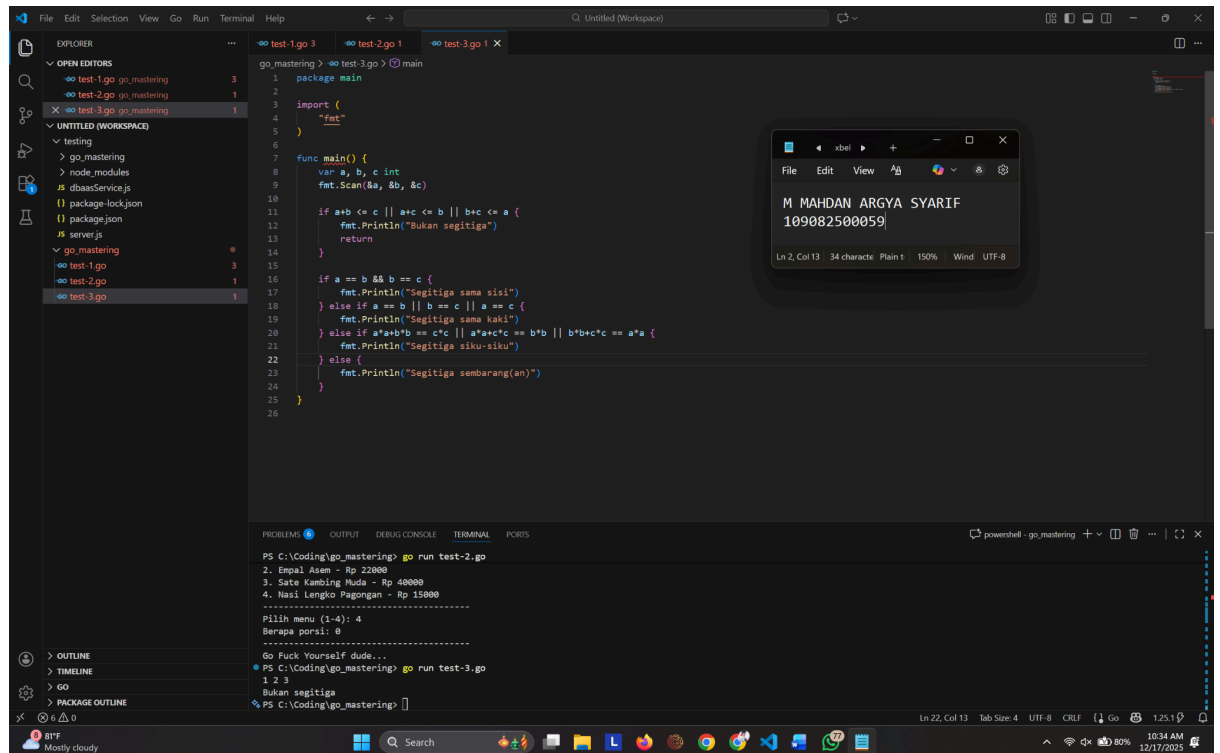
```
package main

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")  
  
        return  
  
    }  
  
    if a == b && b == c {  
  
        fmt.Println("Segitiga sama sisi")  
  
    } else if a == b || b == c || a == c {  
  
        fmt.Println("Segitiga sama kaki")  
  
    } 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 {  
  
        fmt.Println("Segitiga sembarang(an)")  
  
    }  
  
}
```

**Screenshoot program**



## Deskripsi program

Program ini adalah program sederhana yang ditulis dalam bahasa Go untuk memecahkan masalah geometri dasar. Program menerima tiga input bilangan bulat yang merepresentasikan panjang sisi-sisi sebuah segitiga (a, b, c), kemudian menganalisis apakah ketiga sisi tersebut dapat membentuk segitiga yang valid. Jika valid, program akan mengklasifikasikan segitiga tersebut ke dalam kategori spesifik (sama sisi, sama kaki, siku-siku, atau sembarang).

Ringkasan Alur Data :

1. Input.
2. Validasi.
3. Cek Sama Kaki.
4. Cek Siku-siku.
5. Output.