

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

Nama : Primatama Sigalingging

NIM : 109082500076

Kelas : 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

    fmt.Print("Masukkan tinggi pohon (N): ")

    fmt.Scan(&n)

    // Validasi tinggi pohon

    if n >= 8 {
        fmt.Println("Tinggi pohon terlalu besar, silakan kurangi
(kurang dari 8).")

        return
    }

    // Mencetak daun pohon (segitiga)

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

        // Spasi kiri

        for j := 1; j <= n-i; j++ {
            fmt.Print(" ")
        }

        // Asterisk

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

```

        fmt.Println()

    }

    // Mencetak batang pohon (2 baris)

    for i := 0; i < 2; i++ {

        for j := 0; j < n-1; j++ {

            fmt.Print(" ")

        }

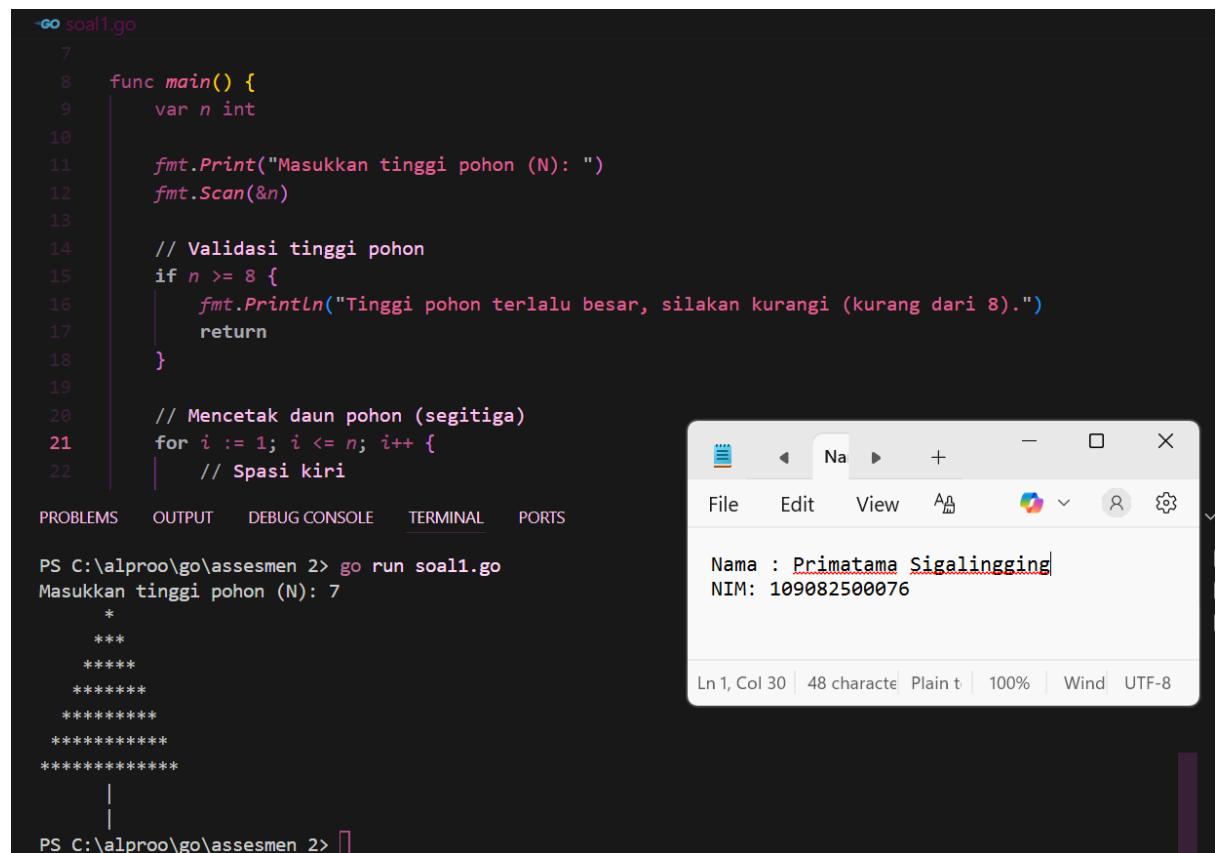
        fmt.Println("|")

    }

}

```

Screenshot program



-∞ soal1.go

```

7
8 func main() {
9     var n int
10
11    fmt.Print("Masukkan tinggi pohon (N): ")
12    fmt.Scan(&n)
13
14    // Validasi tinggi pohon
15    if n >= 8 {
16        fmt.Println("Tinggi pohon terlalu besar, silakan kurangi (kurang dari 8).")
17        return
18    }
19
20    // Mencetak daun pohon (segitiga)
21    for i := 1; i <= n; i++ {
22        // Spasi kiri

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\assesmen 2> go run soal1.go
Masukkan tinggi pohon (N): 7

```

*
 ***
 ****
 *****
 ******
 *****
 ****
 ***
 |

```

Nama : Pramatama Sigalingging
NIM: 109082500076

Ln 1, Col 30 | 48 character| Plain t | 100% | Wind | UTF-8

Deskripsi program

➤ **Tujuan**

Program ini dibuat untuk menampilkan gambar pohon sederhana di layar menggunakan karakter * sebagai daun dan karakter | sebagai batang. Tujuan pembuatan program ini adalah untuk melatih pemahaman perulangan (loop), percabangan (if), serta pengaturan spasi dalam bahasa pemrograman Go.

➤ **Cara Kerja Program**

1. Program meminta pengguna untuk memasukkan tinggi pohon (N).
2. Program melakukan validasi input, yaitu jika nilai N lebih besar atau sama dengan 8, maka program akan menampilkan pesan peringatan dan berhenti.
3. Jika nilai N valid, program akan mencetak:
 - Daun pohon berbentuk segitiga menggunakan perulangan bersarang.
 - Spasi digunakan agar bentuk pohon simetris.
 - Karakter * digunakan untuk membentuk daun pohon.
 - Batang pohon dicetak sebanyak 2 baris dan diletakkan tepat di tengah daun pohon menggunakan spasi dan karakter |.

➤ **Input**

Sebuah bilangan bulat N sebagai tinggi pohon.

➤ **Output**

Tampilan gambar pohon berbentuk segitiga dengan batang di bawahnya sesuai dengan tinggi yang dimasukkan.

➤ **Konsep yang Digunakan**

- a) Input dan output (fmt.Scan, fmt.Print, fmt.Println)
- b) Percabangan (if)
- c) Perulangan (for)
- d) Perulangan bersarang
- e) Manipulasi spasi dan karakter

➤ **Kesimpulan**

Program ini menunjukkan bagaimana kombinasi logika perulangan dan percabangan dapat digunakan untuk membentuk pola visual sederhana di terminal menggunakan bahasa pemrograman Go.

2. SOAL 2

Source Code

```
package main

import (
    "fmt"
)

func main() {
    // Data produk
    products := []string{
        "Little Trees",
        "Lap Microfiber",
        "Cover Steer",
        "Sponge Cuci Mobil",
    }

    prices := []int{
        35000,
        25000,
        150000,
        10000,
    }

    // Tampilkan daftar produk
    fmt.Println("==== DAFTAR PRODUK TOKO BUDI ====")
    for i := 0; i < len(products); i++ {
        fmt.Printf("%d. %s - Rp%d\n", i+1, products[i],
    prices[i])
    }

    // Input pilihan produk
    var choice int
```

```
fmt.Println("Pilih produk (1-4): ")

fmt.Scan(&choice)

// Input jumlah beli
var qty int
fmt.Println("Masukkan jumlah beli: ")
fmt.Scan(&qty)

// Ambil data produk
index := choice - 1
productName := products[index]
price := prices[index]
total := price * qty

// Output struk
fmt.Println("\n==== STRUK PEMBAYARAN ===")
fmt.Println("Produk : ", productName)
fmt.Println("Harga : Rp", price)
fmt.Println("Jumlah : ", qty)
fmt.Println("Total : Rp", total)
}
```

Screenshot program

The screenshot shows a terminal window with the following content:

```
-go soal2.go
7
8 func main() {
9     // Data produk
10    products := []string{
11        "Little Trees",
12        "Lap Microfiber",
13        "Cover Steer",
14        "Sponge Cuci Mobil",
15    }
16
17    prices := []int{
18        35000,
19        25000,
20        150000,
21        10000,
22    }

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\alproo\go\assesmen 2> go run soal2.go
== DAFTAR PRODUK TOKO BUDI ==
1. Little Trees - Rp35000
2. Lap Microfiber - Rp25000
3. Cover Steer - Rp150000
4. Sponge Cuci Mobil - Rp10000
Pilih produk (1-4): 1
Masukkan jumlah beli: 3

== STRUK PEMBAYARAN ==
Produk : Little Trees
Harga : Rp 35000
```

A separate window shows the input and output:

File	Edit	View	Plain t	100%	Wind	UTF-8
Nama : Primatama Sigalingging NIM: 109082500076						

The screenshot shows a terminal window with the following content:

```
-go soal2.go
8 func main() {
24     // Tampilkan daftar produk
25     fmt.Println("== DAFTAR PRODUK TOKO BUDI ==")
26     for i := 0; i < Len(products); i++ {
27         fmt.Printf("%d. %s - Rp%d\n", i+1, products[i], prices[i])
28     }

29
30     // Input pilihan produk
31     var choice int
32     fmt.Print("Pilih produk (1-4): ")
33     fmt.Scan(&choice)

34
35     // Input jumlah beli
36     var qty int
37     fmt.Print("Masukkan jumlah beli: ")
38     fmt.Scan(&qty)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

2. Lap Microfiber - Rp25000
3. Cover Steer - Rp150000
4. Sponge Cuci Mobil - Rp10000
Pilih produk (1-4): 1
Masukkan jumlah beli: 3

== STRUK PEMBAYARAN ==
Produk : Little Trees
Harga : Rp 35000
Jumlah : 3
Total : Rp 105000
PS C:\alproo\go\assesmen 2>
```

A separate window shows the input and output:

File	Edit	View	Plain t	100%	Wind	UTF-8
Nama : Primatama Sigalingging NIM: 109082500076						

Deskripsi program

➤ **Tujuan**

Program ini dibuat untuk mensimulasikan proses pembelian produk di Toko Budi. Tujuan utamanya adalah melatih penggunaan array/slice, perulangan, serta input dan output dalam bahasa pemrograman Go.

➤ **Cara Kerja Program**

1. Program menyimpan data nama produk dan harga produk ke dalam dua slice, yaitu products dan prices.
2. Program menampilkan daftar produk beserta harganya kepada pengguna menggunakan perulangan for.
3. Pengguna diminta untuk:
 - Memilih produk berdasarkan nomor.
 - Memasukkan jumlah barang yang ingin dibeli.
4. Program mengambil data produk sesuai pilihan pengguna.
5. Program menghitung total harga dengan mengalikan harga produk dengan jumlah beli.
6. Program menampilkan struk pembayaran yang berisi nama produk, harga satuan, jumlah beli, dan total pembayaran.

➤ **Input**

- a) Nomor produk (1–4)
- b) Jumlah barang yang dibeli

➤ **Output**

- a) Daftar produk dan harga
- b) Struk pembayaran yang berisi:
- c) Nama produk
- d) Harga satuan
- e) Jumlah pembelian
- f) Total harga

➤ **Konsep yang Digunakan**

- a) Slice / array
- b) Perulangan (for)
- c) Input dan output (fmt.Scan, fmt.Print, fmt.Println, fmt.Printf)
- d) Operasi aritmatika (perkalian)
- e) Pengolahan data sederhana

➤ **Kesimpulan**

Program ini dapat digunakan sebagai contoh sederhana sistem kasir. Dengan program ini, pengguna dapat memahami bagaimana data disimpan, dipilih, dan diolah untuk menghasilkan total pembayaran menggunakan bahasa pemrograman Go.

3. SOAL 3

Source Code

```
package main

import "fmt"

func main() {
    var a, b, c int
    fmt.Scan(&a, &b, &c)

    // Cek syarat segitiga
    if a+b <= c || a+c <= b || b+c <= a {
        fmt.Println("Bukan segitiga")
        return
    }

    // Segitiga sama sisi
    if a == b && b == c {
        fmt.Println("Segitiga sama sisi")
        return
    }

    // Urutkan sisi (supaya mudah cek siku-siku)
    x, y, z := a, b, c
    if x > y {
        x, y = y, x
    }
    if y > z {
        y, z = z, y
    }
    if x > y {
        x, y = y, x
    }
}
```

```
// Segitiga siku-siku (Pythagoras)
if x*x+y*y == z*z {
    fmt.Println("Segitiga siku-siku")
    return
}

// Segitiga sama kaki
if a == b || a == c || b == c {
    fmt.Println("Segitiga sama kaki")
    return
}

// Segitiga sembarang
fmt.Println("Segitiga sembarang")
}
```

Screenshot program

The screenshot shows a code editor with a Go file named `soal3.go`. The code implements logic to check if three given sides can form a triangle and, if so, classifies it as equilateral, isosceles, or scalene. The terminal window below shows the program's output for various input sets.

```
soal3.go
6 func main() {
10     // Cek syarat segitiga
11     if a+b <= c || a+c <= b || b+c <= a {
12         fmt.Println("Bukan segitiga")
13         return
14     }
15
16     // Segitiga sama sisi
17     if a == b && b == c {
18         fmt.Println("Segitiga sama sisi")
19         return
20     }
21
22     // Urutkan sisi (supaya mudah cek siku-siku)
23     x, y, z := a, b, c
24     if x > y {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\alproo\go\assesmen 2> go run soal3.go
4 6 8
Segitiga sembarang
PS C:\alproo\go\assesmen 2> go run soal3.go
2 2 2
Segitiga sama sisi
PS C:\alproo\go\assesmen 2> go run soal3.go
5 5 8
Segitiga sama kaki
PS C:\alproo\go\assesmen 2> go run soal3.go
1 2 3
Bukan segitiga
```

File Edit View Aa 🔍 ⚙️

Nama : Primatama Sigalingging
NIM: 109082500076

Ln 1, Col 30 | 48 caracte Plain t | 100% | Wind | UTF-8

Deskripsi program

➤ Tujuan

Program ini dibuat untuk menentukan jenis segitiga berdasarkan panjang sisi yang dimasukkan pengguna. Tujuan pembuatan program ini adalah untuk melatih penggunaan percabangan (if), logika matematika, dan validasi input dalam bahasa pemrograman Go.

➤ Cara Kerja Program

1. Program meminta pengguna memasukkan tiga bilangan bulat yang mewakili panjang sisi segitiga (a, b, c).
2. Program melakukan validasi syarat segitiga:
 - Sebuah segitiga valid jika jumlah dua sisi lebih besar dari sisi ketiga.
 - Jika tidak memenuhi, program menampilkan "Bukan segitiga" dan berhenti.
3. Jika valid, program menentukan jenis segitiga:
 - Segitiga sama sisi: ketiga sisi sama panjang.
 - Segitiga siku-siku: memeriksa menggunakan teorema Pythagoras (sisi-sisi diurutkan dulu dari terkecil ke terbesar).
 - Segitiga sama kaki: dua sisi sama panjang.
 - Segitiga sembarang: jika semua sisi berbeda dan tidak siku-siku.
4. Program menampilkan jenis segitiga yang sesuai di layar.

➤ **Input**

Tiga bilangan bulat a , b , dan c sebagai panjang sisi segitiga.

➤ **Output**

- a) Jenis segitiga:
- b) "Segitiga sama sisi"
- c) "Segitiga siku-siku"
- d) "Segitiga sama kaki"
- e) "Segitiga sembarang"
- f) "Bukan segitiga" (jika input tidak membentuk segitiga)

➤ **Konsep yang Digunakan**

- a) Input dan output (`fmt.Scan`, `fmt.Println`)
- b) Percabangan bersyarat (`if`, `return`)
- c) Logika matematika dan perbandingan
- d) Teorema Pythagoras untuk segitiga siku-siku
- e) Penukaran nilai untuk pengurutan variabel

➤ **Kesimpulan**

Program ini mempermudah pengguna mengidentifikasi jenis segitiga hanya dengan memasukkan panjang sisinya. Program ini juga mengajarkan konsep validasi data dan pengambilan keputusan dalam pemrograman Go.