

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

**EVALUASI**



**Disusun oleh:**

**OFI ANDRE KHORIUNIZA**

**109082500061**

**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.Print("Input ketinggian piramida (N < 8): ")
        fmt.Scan(&n)

        if n < 8 {
            break
        }

        fmt.Println("Ketinggian harus kurang dari 8")
    }

    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(" ")

        }

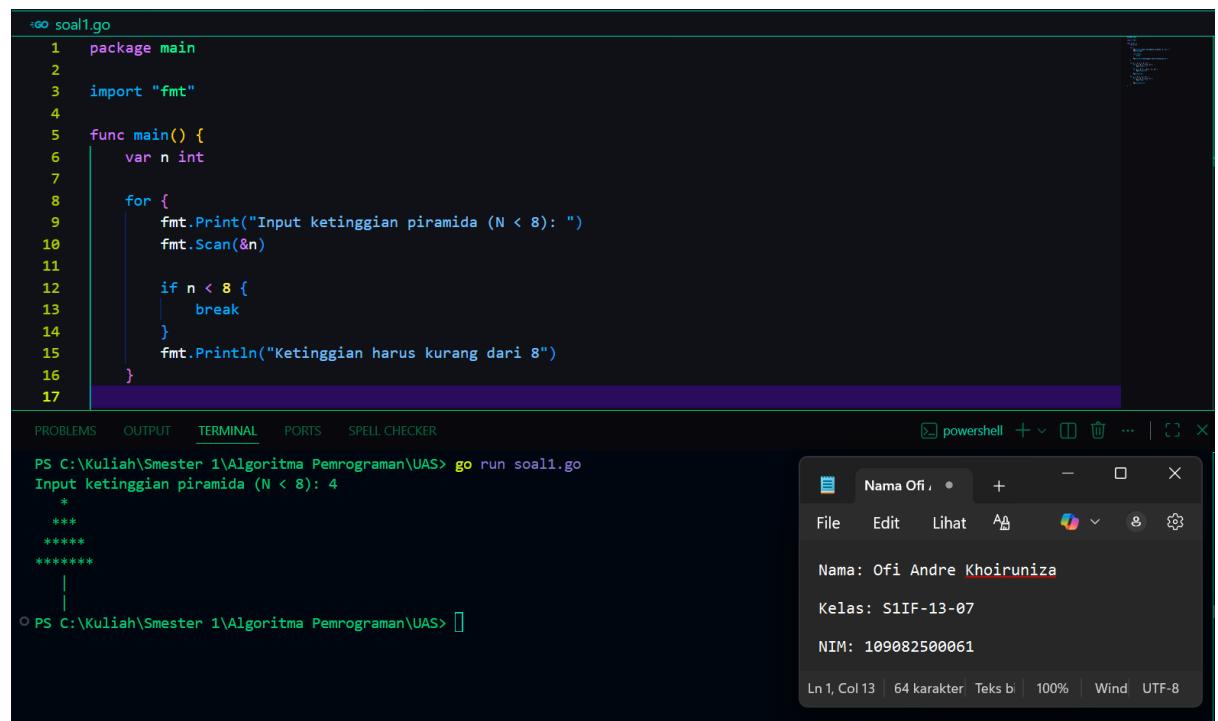
        fmt.Println(" | ")

    }

}

```

## Screenshot program



```

:~/soal1.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7
8     for {
9         fmt.Print("Input ketinggian piramida (N < 8): ")
10        fmt.Scan(&n)
11
12        if n < 8 {
13            break
14        }
15        fmt.Println("Ketinggian harus kurang dari 8")
16    }
17

```

PROBLEMS OUTPUT TERMINAL PORTS SPELL CHECKER

PS C:\Kuliah\Smester 1\Algoritma Pemrograman\UAS> go run soal1.go  
Input ketinggian piramida (N < 8): 4  
\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
|

PS C:\Kuliah\Smester 1\Algoritma Pemrograman\UAS>

Nama: Ofi Andre Khoiruniza  
Kelas: S1IF-13-07  
NIM: 109082500061

Ln 1, Col 13 | 64 karakter | Teks bi | 100% | Wind | UTF-8

## Deskripsi program

Program ini bekerja dengan dua tahap utama, yaitu validasi input dan pencetakan bentuk. Pertama, program menggunakan perulangan tak hingga (`for {}`) untuk meminta input ketinggian piramida dan memastikan nilai `n` kurang dari 8; jika syarat terpenuhi, perulangan dihentikan dengan `break`, sedangkan jika tidak, program menampilkan pesan kesalahan dan meminta input ulang. Setelah input valid, program

masuk ke tahap pencetakan piramida menggunakan perulangan bertingkat: perulangan luar mengatur jumlah baris piramida dari atas ke bawah, perulangan pertama di dalamnya mencetak spasi agar piramida berada di tengah, dan perulangan kedua mencetak karakter \* dengan jumlah ganjil yang bertambah di setiap baris sehingga membentuk segitiga simetris. Terakhir, program mencetak batang piramida dengan perulangan yang dijalankan dua kali, di mana setiap baris mencetak spasi dan satu karakter | tepat di tengah, sehingga keseluruhan output membentuk tampilan piramida yang utuh dan proporsional.

## 2. SOAL 2

### Source Code

```
package main

import "fmt"

func main() {
    var a, b int

    fmt.Println("==== Daftar Produk Toko Budi ===")
    fmt.Println("1. Little Trees      - 35.000")
    fmt.Println("2. Lap microfiber     - 25.000")
    fmt.Println("3. Cover Steer        - 150.000")
    fmt.Println("4. Sponge cuci mobil   - 10.000")

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

    fmt.Print("Masukkan jumlah porsi: ")
    fmt.Scan(&b)
```

```
harga := 0
nama := ""

switch a {
    case 1:
        harga = 35000
        nama = "Little Trees"
    case 2:
        harga = 25000
        nama = "Lap microfiber"
    case 3:
        harga = 150000
        nama = "Cover Steer"
    case 4:
        harga = 10000
        nama = "Sponge cuci mobil"
    default:
        fmt.Println("Menu tidak tersedia")
        return
}

total := harga * b

fmt.Println("\n==== STRUK PEMBELIAN ====")
fmt.Println("Produk :", nama)
```

```

        fmt.Println("Jumlah : ", a)

        fmt.Println("Total : ", total)

    }

```

## Screenshot program

The screenshot shows a terminal window with the following content:

```

:~/soal2.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b int
7
8     fmt.Println("== Daftar Produk Toko Budi ==")
9     fmt.Println("1. Little Trees      - 35.000")
10    fmt.Println("2. Lap microfiber   - 25.000")
11    fmt.Println("3. Cover Steer       - 150.000")
12    fmt.Println("4. Sponge cuci mobil - 10.000")
13
14    fmt.Print("Pilih menu (1-4): ")
15    fmt.Scan(&a)
16
17    fmt.Print("Masukkan jumlah porsi: ")

```

TERMINAL tab is selected. The output shows:

```

● PS C:\Kuliah\Semester 1\Algoritma Pemrograman\UAS> go run soal2.go
== Daftar Produk Toko Budi ==
1. Little Trees      - 35.000
2. Lap microfiber   - 25.000
3. Cover Steer       - 150.000
4. Sponge cuci mobil - 10.000
Pilih menu (1-4): 3
Masukkan jumlah porsi: 3

== STRUK PEMBELIAN ==
Produk : Cover Steer
Jumlah : 3
Total  : 450000
○ PS C:\Kuliah\Semester 1\Algoritma Pemrograman\UAS>

```

A separate window titled "Nama Ofi" is open, showing student information:

File	Edit	Lihat	AA	...	...
Nama: Ofi Andre Khoiruniza					
Kelas: S1IF-13-07					
NIM: 109082500061					

Ln 1, Col 13 | 64 karakter Teks bi | 100% | Wind | UTF-8

## Deskripsi program

Program ini mensimulasikan proses pembelian sederhana di Toko Budi dengan alur yang terstruktur. Pertama, program menampilkan daftar produk beserta harga agar pengguna mengetahui pilihan yang tersedia, kemudian menerima input nomor menu (a) dan jumlah pembelian (b). Selanjutnya, program menggunakan struktur switch-case untuk mencocokkan nomor menu dengan nama produk dan harga satuan yang sesuai; jika pengguna memasukkan menu di luar rentang 1–4, program langsung menampilkan pesan kesalahan dan dihentikan dengan return. Setelah produk valid, program menghitung total harga dengan mengalikan harga satuan dan jumlah pembelian, lalu menampilkan struk pembelian yang berisi nama produk, jumlah pembelian, dan total biaya, sehingga seluruh proses dari pemilihan barang hingga perhitungan pembayaran dilakukan secara runtut dan jelas.

## 3. SOAL 3

### Source Code

```

package main

```

```
import "fmt"

func main() {
    var a, b, c int

    fmt.Print("Masukkan tiga bilangan: ")

    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 a code editor with a Go file named `soal3.go`. The code implements a program to classify a triangle based on three input sides. It first checks if the sum of any two sides is less than or equal to the third, which would make it invalid. If valid, it checks for an equilateral triangle ( $a == b == c$ ). If not, it checks for a right-angled triangle using the Pythagorean theorem ( $a^2 + b^2 == c^2$ ,  $a^2 + c^2 == b^2$ , or  $b^2 + c^2 == a^2$ ). If none of these conditions are met, it's an isosceles triangle ( $a == b$ ,  $a == c$ , or  $b == c$ ). The code uses `fmt.Println` to output the classification.

```
soal3.go
4
5  func main() {
6      var a, b, c int
7      fmt.Print("Masukkan tiga bilangan: ")
8      fmt.Scan(&a, &b, &c)
9
10     if a+b <= c || a+c <= b || b+c <= a {
11         fmt.Println("bukan segitiga")
12     } else {
13         if a == b && b == c {
14             fmt.Println("segitiga sama sisi")
15         } else if (a*a + b*b == c*c) || (a*a + c*c == b*b) || (b*b + c*c == a*a) {
16             fmt.Println("segitiga siku-siku")
17         } else if a == b || a == c || b == c {
18             fmt.Println("segitiga sama kaki")
19         }
20     }
21 }
```

The terminal window shows the program being run with different sets of input values, and the output correctly classifies each triangle type.

```
PS C:\Kuliah\Semester 1\Algoritma Pemrograman\UAS> go run soal3.go
Masukkan tiga bilangan: 10 10 5
segitiga sama kaki
● PS C:\Kuliah\Semester 1\Algoritma Pemrograman\UAS> go run soal3.go
Masukkan tiga bilangan: 1 2 3
bukan segitiga
○ PS C:\Kuliah\Semester 1\Algoritma Pemrograman\UAS>
```

A floating note card displays student information:

File	Edit	Lihat	AA	...	...	X
Nama: Ofi Andre Khoiruniza						
Kelas: S1IF-13-07						
NIM: 109082500061						
Ln 1, Col 13   64 karakter   Teks b   100%   Wind   UTF-8						

## Deskripsi program

Program ini digunakan untuk menentukan jenis segitiga berdasarkan tiga bilangan sisi yang dimasukkan oleh pengguna. Pertama, program membaca tiga nilai input lalu memeriksa syarat valid segitiga dengan memastikan jumlah dua sisi selalu lebih besar dari sisi ketiga; jika syarat ini tidak terpenuhi, program menyimpulkan bahwa bangun tersebut bukan segitiga. Jika valid, program melanjutkan ke proses klasifikasi menggunakan percabangan bertingkat, dimulai dari pengecekan segitiga sama sisi (ketiga sisi sama), kemudian segitiga siku-siku dengan membandingkan kuadrat sisi sesuai teorema Pythagoras, dilanjutkan dengan segitiga sama kaki jika terdapat dua sisi yang sama, dan jika tidak memenuhi semua kondisi tersebut maka segitiga dikategorikan sebagai segitiga sembarang, sehingga setiap kemungkinan jenis segitiga dapat ditentukan secara logis dan sistematis.