

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI
(TIPE B)



Disusun oleh:
MUHAMMAD FIRDAUS ARDIANSYAH
109082500126
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, 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 a Go development environment with the following details:

- File Explorer:** Shows a project structure with files like `main.go`, `sosil1.go`, `sosil2.go`, `sosil3.go`, and `code.png`.
- Code Editor:** Displays the `main.go` file containing the provided Go code.
- Terminal:** Shows the command `go run "D:\Coding\Latihan GO\1\soasil\main.go"` and its output, which is a pyramid of asterisks.
- Message Box:** A modal window displays the student's information: MUHAMMAD FIRDAUS ARDIANSYAH, S1IF-13-07, and 109082500126.

Deskripsi program

Program ini dibuat untuk men-generate pola visual pohon yang terdiri dari piramida bintang dan batang. Alurnya dimulai dengan inisialisasi variabel n, yang langsung masuk ke validasi input menggunakan infinite loop. Di blok ini,kita dipaksa menginput nilai di bawah 8. Kalau kondisi $n < 8$ terpenuhi, program akan men-trigger break untuk keluar dari loop; sebaliknya, program akan terus me-looping permintaan input sampai valid. Validasi ini penting untuk menjaga proporsi gambar agar tidak berantakan di terminal.

Secara teknis, logika pembentukannya dipecah menjadi dua segmen. Segmen pertama Adalah nested loop untuk membangun piramida: loop luar (i) mengontrol jumlah baris, loop dalam (j) menangani spasi untuk indentasi, dan loop (k) mencetak bintang berdasarkan rumus deret ganjil $2*i - 1$ agar hasilnya simetris. Segmen kedua adalah loop terpisah untuk membuat batang pohon. Logikanya sederhana, cukup mencetak spasi sebanyak $n-1$ lalu diikuti karakter |, sehingga posisi batang otomatis *center* mengikuti puncak piramida.

2. SOAL 2

Source Code

```
package main

import "fmt"

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

    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.Print("Pilih produk (1-4): ")
```

```
fmt.Scan(&pilihan)

fmt.Println("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("Pilihan tidak valid")

    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 Go development environment with multiple windows. On the left, there's a file browser with files like `main.go`, `list.go`, `menu.go`, and `struktur.go`. The main code editor window contains the provided Go code. To the right, a terminal window shows the execution of the program and its output. The output window displays the program's interface and a sample transaction:

```

MUHAMMAD FIRDAUS ARDIANSYAH
S11F-13-07
1090082500126

PILIH produk (1-4):
1. Little Trees - Rp 35000
2. Lap Microfiber - Rp 50000
3. Cover Steer - Rp 100000
4. Sponge Cuci Mobil - Rp 100000

PILIH produk (1-4):
Produk : Little Trees
Harga : Rp 35000
Jumlah : 5
Total : Rp 175000

```

Deskripsi program

Program ini pada dasarnya adalah simulasi sistem kasir atau *Point of Sales* (POS) sederhana untuk "Toko Budi". program pertama-tama me-render daftar menu produk beserta harganya ke layar, lalu meminta kita menginput nomor pilihan produk dan jumlah barang yang ingin dibeli. Seperti yang terlihat di output terminal, saat kita memilih opsi 1 ("Little Trees") dengan kuantitas 5 items, program langsung memprosesnya menjadi struk pembayaran lengkap dengan total harga Rp 175.000.

program ini sangat bergantung pada struktur kontrol switch-case untuk memetakan input kita ke variabel data yang relevan. Setiap case (1-4) berfungsi untuk men-set nilai variabel namaProduk dan harga sesuai pilihan; jika input di luar itu, blok default akan me-return pesan error. Setelah variabel harga terisi, program tinggal melakukan operasi aritmatika perkalian (harga * jumlah) untuk mendapatkan grand total, lalu menampilkan rincian akhirnya menggunakan fmt.Printf agar format struknya rapi dan terstruktur.

3. SOAL 3

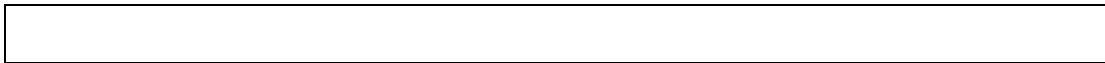
Source Code

```
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 == b || a == c || b == 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")
        }
    }
}
```



Screenshot program

Deskripsi program

Program ini memproses tiga variabel input dari kita guna menentukan klasifikasi bangun datar segitiga. Langkah awal algoritma berfokus pada validasi kelayakan bentuk; jika input melanggar aturan dasar geometri di mana total panjang dua sisi tidak melebihi sisi ketiga, maka program akan mencetak keluaran "bukan segitiga". Namun, apabila input terbukti valid, sistem melanjutkan proses seleksi untuk menentukan apakah bangun tersebut termasuk segitiga sama sisi, sama kaki, siku-siku, atau sekadar segitiga sembarang.

Implementasi logikanya disusun menggunakan percabangan if-else bertingkat yang menyeleksi kondisi secara berurutan. Penentuan jenis segitiga dilakukan dengan operator boolean: penggunaan `&&` untuk memastikan ketiga sisi identik, `||` untuk mengecek apakah ada dua sisi yang sama, serta rumus Pythagoras ($a^2 + b^2 = c^2$) untuk mendeteksi segitiga siku-siku. Blok kode terakhir disiapkan sebagai kondisi default untuk menangani segitiga sembarang, memastikan alur logika selalu menghasilkan output yang sesuai dengan karakteristik inputnya.