

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

HAFIZD SMAA'I SYAMSI

109082500183

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("Masukan ketinggian piramida")
    fmt.Scan(&N)

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

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



Screenshot program

The screenshot shows a code editor with two tabs: 'soal1.go' and 'soal2.go'. The 'soal1.go' tab contains the following Go code:

```
1 package main
2 import "fmt"
3 func main() {
4     var N int
5     fmt.Println("Masukan ketinggian piramida")
6     fmt.Scan(&N)
7
8     for i := 1; i <= N; i++ {
9         for s := 0; s < N-i; s++ {
10            fmt.Print(" ")
11        }
12        for b := 0; b < 2*i-1; b++ {
13            fmt.Print("*")
14        }
15        fmt.Println()
16    }
17
18    for i := 0; i < 2; i++ {
19        for s:= 0; s < N-1; s++ {
20            fmt.Print(" ")
21        }
22    }
}
```

The terminal window shows the command `PS D:\go\UAS> go run soal1.go` followed by the user input `Masukan ketinggian piramida 7`. The output is a 7-line pyramid pattern:

```
*****
 ****
 ***
 *****
 ******
 *****
|
```

The right side of the terminal window shows the user's profile information: NAMA : HAFIZD SAMA'I SYAMSI, KELAS : S1IF-13-07, and NIM : 109082500183.

Deskripsi program

Input Data

- Program meminta pengguna memasukkan ketinggian piramida.
- Nilai tersebut disimpan dalam variabel N.

Pembentukan Badan Piramida

- Piramida dibentuk menggunakan perulangan for dari baris ke-1 hingga baris ke-N.
- Pada setiap baris:
 - Program mencetak spasi sebanyak $N - i$ untuk menggeser bintang ke tengah.
 - Program mencetak bintang (*) sebanyak $2*i - 1$ sehingga lebar piramida bertambah dua karakter di setiap baris ke bawah.
 - Setelah satu baris selesai, program berpindah ke baris berikutnya.

Pembentukan Batang Piramida

- Setelah seluruh badan piramida selesai dicetak, program mencetak batang piramida.
- Batang dicetak sebanyak dua baris menggunakan karakter | .
- Setiap batang diawali dengan N - 1 spasi agar posisinya tepat berada di tengah dasar piramida

2. SOAL 2

Source Code

```

package main

import"fmt"

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

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

    fmt.Println("==== DAFTAR PRODUK TOKO BUDI====")
    for i := 0; i < len(menu); i++ {
        fmt.Printf("%d. %-18s - Rp%d\n", i+1, menu[i],
        harga[i])
    }

    var pilih, jumlah int
    fmt.Print("Pilih produk (1-4)")
    fmt.Scan(&pilih)

    fmt.Print("Masukan jumlah beli")
    fmt.Scan(&jumlah)
}

```

```

        total := harga[pilih-1] * jumlah

        fmt.Println("==== STRUK PEMBAYARAN ===")

        fmt.Println("Produk : ", menu[pilih-1])

        fmt.Println("Harga : Rp", harga[pilih-1])

        fmt.Println("JUumlah :", jumlah)

        fmt.Println("Total : Rp", total)

        fmt.Println()

        fmt.Println("==== Code Excution Successful ===")

    }

```

Screenshoot program

The screenshot shows a code editor interface with the following details:

- File Bar:** Welcome, -go soal1.go 1, -go soal2.go 1 (highlighted), -go soal3.go 1, File, Edit, View.
- Code Area:**

```

1 package main
2 import"fmt"
3 func main() {
4     menu := [] string{
5         "Little Trees",
6         "Lap Microfiber",
7         "Cover Steer",
8         "Sponge Cuci Mobil",
9     }
10    harga := [] int{
11        35000,
12        25000,
13        150000,
14        10000,
15    }
16    fmt.Println("==== DAFTAR PRODUK TOKO BUDI====")
17    for i := 0; i < len(menu); i++ {
18        fmt.Printf("%d. %18s - Rp%d\n", i+1, menu[i], harga[i])
19    }
20    var pilih, jumlah int
21    fmt.Print("Pilih produk (1-4)")
22    fmt.Scan(&pilih)

```
- Output Area:**

```

NAMA :HAFIZD SAMA'I SYAMSI
KELAS : S1IF-13-07
NIM : 109082500183

```

Ln 3, Col 20 | 65 characters | Plain
- Bottom Navigation:** PROBLEMS 3, OUTPUT, DEBUG CONSOLE, TERMINAL (highlighted), PORTS.
- Terminal Output:**

```

goroutine 1 [running]:
main.main()
D:/go/UAS/go run soal2.go
exit status 2
PS D:\go\UAS> 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) []

```

Deskripsi program

Inisialisasi Data Produk

- Program menyimpan nama produk dalam slice menu bertipe string.
- Program menyimpan harga masing-masing produk dalam slice harga bertipe int.
- Setiap indeks pada menu berpasangan langsung dengan indeks yang sama pada harga, sehingga data produk dan harga saling terhubung.

Menampilkan Daftar Produk

- Program mencetak judul daftar produk.
- Menggunakan perulangan for, program menampilkan seluruh produk beserta harganya.
- Penomoran produk dimulai dari angka 1 agar mudah dipilih pengguna.
- Format output dirapikan menggunakan fmt.Println dengan pengaturan lebar teks.

Input Pilihan Produk

- Pengguna diminta memasukkan nomor produk (1–4).
- Nilai input disimpan dalam variabel pilih.
- Karena indeks slice dimulai dari 0, program mengakses data menggunakan pilih - 1.

Input Jumlah Pembelian

- Pengguna memasukkan jumlah barang yang ingin dibeli.
- Nilai ini disimpan dalam variabel jumlah.

Perhitungan Total Harga

- Total pembayaran dihitung dengan rumus:

$$\text{Total} = \text{harga produk} \times \text{jumlah pembelian}$$

Menampilkan Struk Pembayaran

- Program mencetak struk yang berisi:
- Nama produk yang dibeli
- Harga satuan produk
- Jumlah barang
- Total harga yang harus dibayar
- Seluruh informasi diambil dari slice berdasarkan pilihan pengguna.

Penutup Program

- Program menampilkan pesan bahwa eksekusi kode berhasil, menandakan proses telah selesai tanpa error.

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")
        return
    }

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

    if a*a+b*b == c*c || a*a+c*c == b*b || b*b+c*c == a*a {
        fmt.Println("Segitiga siku-siku")
        return
    }

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

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

Screenshot program

The screenshot shows a code editor interface with a Go file named `soal3.go`. The code defines a `main` package with a `main` function that reads three integers `a`, `b`, and `c` from standard input. It then checks various conditions to determine if the triangle is scalene (`Bukan segitiga`), equilateral (`Segitiga sama sisi`), isosceles (`Segitiga siku-siku`), or equiangular (`Segitiga sama kaki`). The terminal below shows the program's output for different input sets.

```
1 package main
2 import "fmt"
3 func main() {
4     var a, b, c int
5     fmt.Scan(&a, &b, &c)
6
7     if a+b <= c || a+c <= b || b+c <= a {
8         fmt.Println("Bukan segitiga")
9         return
10    }
11    if a == b && b == c {
12        fmt.Println("Segitiga sama sisi")
13        return
14    }
15    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        return
18    }
19    if a == b || a == c || b == c {
20        fmt.Println("Segitiga sama kaki")
21        return
22    }
}
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\go\UAS> go run soal3.go
5 5 8
Segitiga sama kaki
PS D:\go\UAS> go run soal3.go
3 4 5
Segitiga siku-siku
PS D:\go\UAS> go run soal3.go
4 5 6
Segitiga sembarang
PS D:\go\UAS> go run soal3.go
1 2 3
Bukan segitiga
PS D:\go\UAS>
```

Deskripsi program

Input Data

- Program membaca tiga bilangan bulat `a`, `b`, dan `c` menggunakan `fmt.Scan`.
- Ketiga nilai tersebut dianggap sebagai panjang sisi segitiga.

Validasi Segitiga

```
if a+b <= c || a+c <= b || b+c <= a
```

- Program mengecek syarat segitiga:
 - Jumlah dua sisi harus lebih besar dari sisi ketiga.
 - Jika salah satu kondisi tidak terpenuhi, program langsung menampilkan:
 - Bukan segitiga
- lalu menghentikan eksekusi dengan `return`.

Pemeriksaan Segitiga Sama Sisi

```
if a == b && b == c
```

- Jika semua sisi sama panjang, program mencetak:

- Segitiga sama sisi
- Eksekusi dihentikan agar tidak lanjut ke pengecekan lain.

Pemeriksaan Segitiga Siku-siku

```
if a*a+b*b == c*c || a*a+c*c == b*b || b*b+c*c == a*a
```

- Program menerapkan Teorema Pythagoras untuk menentukan apakah segitiga siku-siku.
- Semua kemungkinan posisi sisi miring diuji.
- Jika terpenuhi, program mencetak:
- Segitiga siku-siku

Pemeriksaan Segitiga Sama Kaki

```
if a == b || a == c || b == c
```

- Program mengecek apakah dua sisi memiliki panjang yang sama.
- Jika benar, program mencetak:
- Segitiga sama kaki

Segitiga Sembarang

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

- Jika tidak memenuhi seluruh kondisi sebelumnya, segitiga diklasifikasikan sebagai segitiga dengan tiga sisi berbeda dan bukan siku-siku.