

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

Anindya Rahadita Yumnaa

109082500138

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)

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

        return
    }

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

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

            fmt.Print(" ")

        }

        for k := 1; 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("|")

    }

}

```

Screenshoot program

The screenshot shows a Go program in VS Code and its output in a terminal window. The program is a recursive function to calculate the number of ways to climb a staircase of height n . The output shows the result for $n=8$ is 55, and for $n=13$ is 377.

```

1 package main
2
3 import "fmt"
4
5 func main() {
6     var n int
7
8     fmt.Print("Masukkan tinggi pohon (n): ")
9     fmt.Scan(&n)
10
11     if n >= 8 {
12         fmt.Println("Tinggi pohon terlalu besar, silakan kurangi ketinggian.")
13         return
14     }
15
16     for i := 1; i <= n; i++ {
17
18         for j := 1; j <= n-i; j++ {
19             fmt.Print(" ")
20         }
21
22         for k := 1; k <= (2*i - 1); k++ {
23             fmt.Print("+")
24         }
25         fmt.Println()
26     }
27
28     for i := 0; i < 2; i++ {
29         for j := 0; j < n-1; j++ {

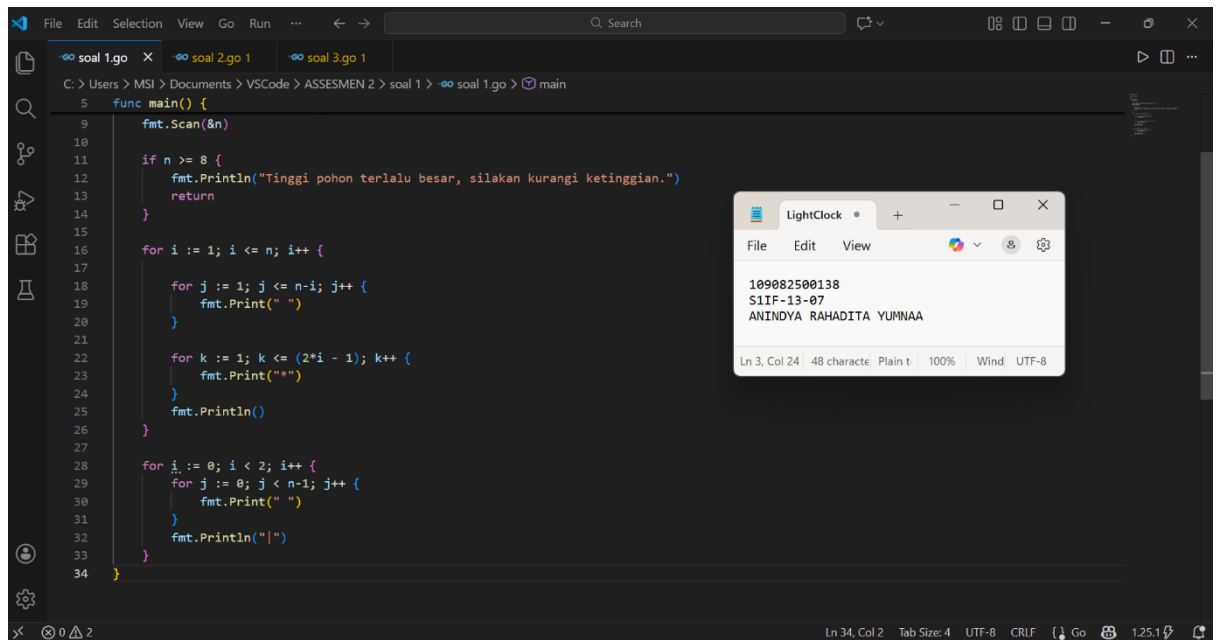
```

Output in terminal window:

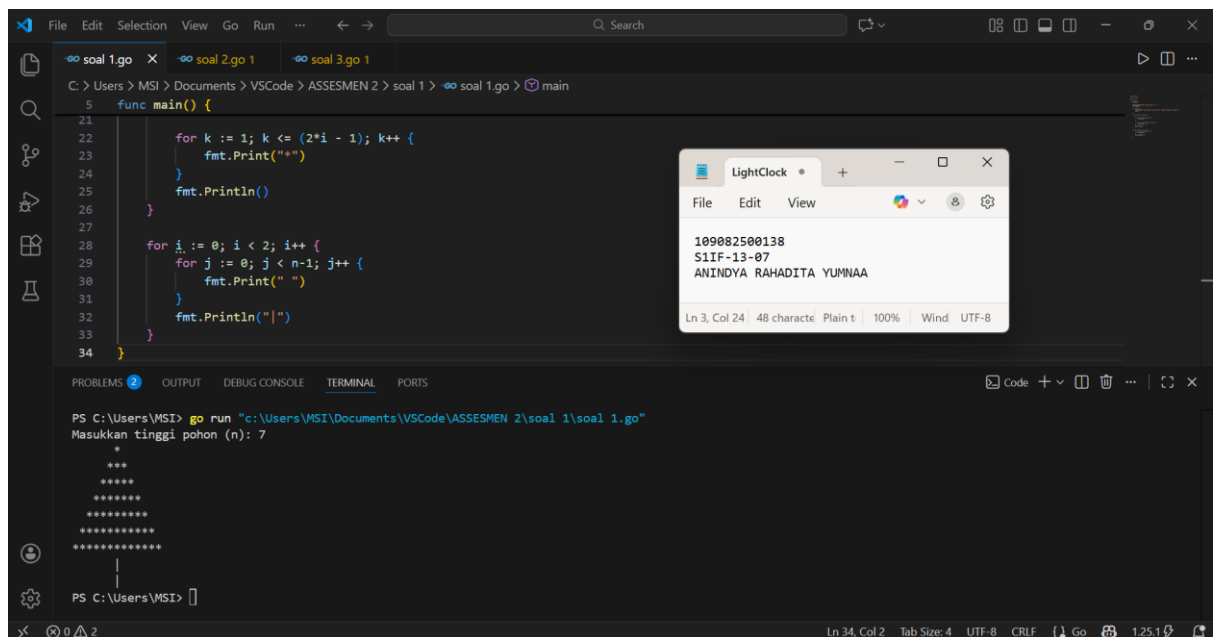
```

109082500138
511F-13-07
ANINDYA RAHADITA YUMNAA

```



```
5 func main() {
9     fmt.Scan(&n)
10
11     if n >= 8 {
12         fmt.Println("Tinggi pohon terlalu besar, silakan kurangi ketinggian.")
13         return
14     }
15
16     for i := 1; i <= n; i++ {
17
18         for j := 1; j <= n-i; j++ {
19             fmt.Print(" ")
20         }
21
22         for k := 1; k <= (2*i - 1); k++ {
23             fmt.Print("*")
24         }
25         fmt.Println()
26     }
27
28     for i := 0; i < 2; i++ {
29         for j := 0; j < n-1; j++ {
30             fmt.Print(" ")
31         }
32         fmt.Println("|")
33     }
34 }
```



```
PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\ASSESMENT 2\soal 1\soal 1.go"
Masukkan tinggi pohon (n): 7
*
***
*****
*****
*****
*****
*****
*****
|
```

Deskripsi program

Program pohon natal dibuat untuk menampilkan pola berbentuk segitiga menyerupai pohon natal menggunakan karakter bintang (*). Program menerima sebuah bilangan bulat positif n sebagai input yang menyatakan tinggi pohon. Berdasarkan nilai tersebut, program akan mencetak pola bintang secara bertahap dari baris pertama hingga baris ke-n. Program juga membatasi tinggi maksimum pohon agar output tetap rapi dan tidak berlebihan.

➤ Alur Kode Program

1. Program meminta pengguna memasukkan nilai tinggi pohon n.
2. Program melakukan pengecekan apakah nilai n melebihi batas maksimum yang ditentukan.

3. Jika nilai *n* terlalu besar, program menampilkan pesan peringatan dan menghentikan proses.
4. Jika nilai *n* valid, program menggunakan perulangan untuk mencetak baris pohon.
5. Setiap baris terdiri dari spasi di sebelah kiri dan karakter bintang (*) di tengah.
6. Jumlah spasi akan berkurang dan jumlah bintang akan bertambah pada setiap baris.
7. Proses ini diulang hingga mencapai tinggi pohon yang diinginkan.

➤ **Kesimpulan**

Program pohon natal berhasil menampilkan pola segitiga bintang sesuai dengan tinggi yang dimasukkan oleh pengguna. Program ini melatih penggunaan struktur perulangan dan logika pengaturan spasi serta karakter, sehingga membantu dalam memahami konsep dasar pemrograman seperti loop dan pengolahan pola teks.

2. SOAL 2

Source Code

```
package main

import "fmt"

func main() {

    produk := []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(produk); i++ {

        fmt.Printf("%d. %s - Rp%d\n", i+1, produk[i],
harga[i])

    }

    var pilihan int

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

    fmt.Scan(&pilihan)

    var jumlah int

    fmt.Print("Masukkan jumlah pembelian: ")

    fmt.Scan(&jumlah)

    index := pilihan - 1

    total := harga[index] * jumlah

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

    fmt.Println("Produk :", produk[index])

    fmt.Println("Harga :", harga[index])

    fmt.Println("Jumlah :", jumlah)

    fmt.Println("Total :", total)

    fmt.Println("\n=== Code Execution Successful ===")

}

```

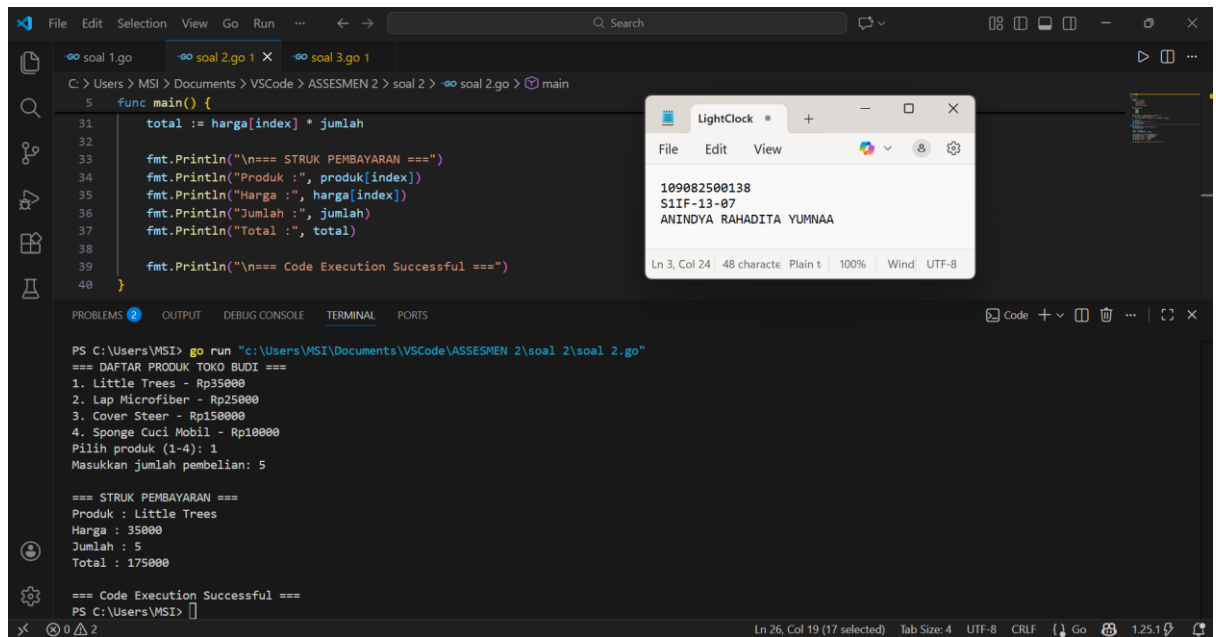
Screenshoot program

The screenshot shows the VS Code editor with a Go file named `soal 2.go`. The code defines a `main` function that initializes a product list and prices, prompts the user to select a product and enter a quantity, and prints the selected product and price. A `LightClock` window is open, displaying the user input: `109082500138`, `S1IF-13-07`, and `ANINDYA RAHADITA YUMNAA`.

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     produk := []string{
7         "Little Trees",
8         "Lap Microfiber",
9         "Cover Steen",
10        "Sponge Cuci Mobil",
11    }
12    harga := []int{
13        35000,
14        25000,
15        150000,
16        10000,
17    }
18    fmt.Println("=== DAFTAR PRODUK TOKO BUDI ===")
19    for i := 0; i < len(produk); i++ {
20        fmt.Printf("%d. %s - Rp%d\n", i+1, produk[i], harga[i])
21    }
22    var pilihan int
23    fmt.Print("Pilih produk (1-4): ")
24    fmt.Scan(&pilihan)
25
26    var jumlah int
27    fmt.Print("Masukkan jumlah pembelian: ")
28    fmt.Scan(&jumlah)
29}
```

The screenshot shows the continuation of the Go code in `soal 2.go`. It calculates the total price based on the selected product and quantity, prints a receipt, and indicates successful code execution. The `LightClock` window remains open with the same user input.

```
16    10000,
17    }
18    fmt.Println("=== DAFTAR PRODUK TOKO BUDI ===")
19    for i := 0; i < len(produk); i++ {
20        fmt.Printf("%d. %s - Rp%d\n", i+1, produk[i], harga[i])
21    }
22    var pilihan int
23    fmt.Print("Pilih produk (1-4): ")
24    fmt.Scan(&pilihan)
25
26    var jumlah int
27    fmt.Print("Masukkan jumlah pembelian: ")
28    fmt.Scan(&jumlah)
29
30    index := pilihan - 1
31    total := harga[index] * jumlah
32
33    fmt.Println("\n=== STRUK PEMBAYARAN ===")
34    fmt.Println("Produk :", produk[index])
35    fmt.Println("Harga :", harga[index])
36    fmt.Println("Jumlah :", jumlah)
37    fmt.Println("Total :", total)
38
39    fmt.Println("\n=== Code Execution Successful ===")
40 }
```



Deskripsi program

- **Program POS** sederhana ini dibuat untuk membantu proses transaksi penjualan di Toko Budi. Program menampilkan daftar produk beserta harga, menerima input pilihan produk dan jumlah pembelian dari kasir, lalu menghitung total harga yang harus dibayar. Hasil perhitungan ditampilkan dalam bentuk struk pembayaran yang berisi informasi produk, harga satuan, jumlah beli, dan total pembayaran.
- **Alur Kode Program**
 1. Program menyimpan data produk dan harga ke dalam array.
 2. Program menampilkan daftar produk kepada pengguna.
 3. Pengguna memilih produk berdasarkan nomor yang tersedia.
 4. Pengguna memasukkan jumlah pembelian.
 5. Program menghitung total pembayaran dengan mengalikan harga produk dan jumlah pembelian.
 6. Program menampilkan struk pembayaran sebagai output.
- **Kesimpulan**

Program POS sederhana ini mampu melakukan proses transaksi penjualan dasar secara otomatis. Dengan adanya program ini, perhitungan harga menjadi lebih cepat, akurat, dan mengurangi kesalahan yang biasanya terjadi pada perhitungan manual.

3. SOAL 3

Source Code

```
package main

import "fmt"
```



```
func main() {  
    var a, b, c int  
  
    fmt.Print("Masukkan tiga bilangan bulat: ")  
    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 == b || a == c || b == c {  
        fmt.Println("Segitiga Sama Kaki")  
        return  
    }  
  
    fmt.Println("Segitiga Sembarang")  
}
```

Screenshoot program

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c int
7
8     fmt.Print("Masukkan tiga bilangan bulat: ")
9     fmt.Scan(&a, &b, &c)
10
11     if a+b <= c || a+c <= b || b+c <= a {
12         fmt.Println("Bukan Segitiga")
13         return
14     }
15     if a == b && b == c {
16         fmt.Println("Segitiga Sama Sisi")
17         return
18     }
19     if a == b || a == c || b == c {
20         fmt.Println("Segitiga Sama Kaki")
21         return
22     }
23     fmt.Println("Segitiga Sembarang")
24 }
```

LightClock

File Edit View

109082500138
S1IF-13-07
ANINDYA RAHADITA YUMNAA

Ln 3, Col 24 48 character Plain t 100% Wind UTF-8

```
5 func main() {
11     if a+b <= c || a+c <= b || b+c <= a {
12         fmt.Println("Bukan Segitiga")
13         return
14     }
15     if a == b && b == c {
16         fmt.Println("Segitiga Sama Sisi")
17         return
18     }
19     if a == b || a == c || b == c {
20         fmt.Println("Segitiga Sama Kaki")
21         return
22     }
23     fmt.Println("Segitiga Sembarang")
24 }
```

LightClock

File Edit View

109082500138
S1IF-13-07
ANINDYA RAHADITA YUMNAA

Ln 3, Col 24 48 character Plain t 100% Wind UTF-8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\ASSESMENT 2\soal 3\soal 3.go"

Masukkan tiga bilangan bulat: 3 3 3

Segitiga Sama Sisi

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\ASSESMENT 2\soal 3\soal 3.go"

Masukkan tiga bilangan bulat: 5 5 8

Segitiga Sama Kaki

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\ASSESMENT 2\soal 3\soal 3.go"

Masukkan tiga bilangan bulat: 4 5 6

Segitiga Sembarang

PS C:\Users\MSI> go run "c:\Users\MSI\Documents\VSCode\ASSESMENT 2\soal 3\soal 3.go"

Masukkan tiga bilangan bulat: 1 2 3

Bukan Segitiga

PS C:\Users\MSI>

Deskripsi program

Program ini dibuat untuk **menentukan jenis segitiga** berdasarkan tiga bilangan bulat positif yang mewakili panjang sisi segitiga. Program akan memeriksa apakah ketiga sisi memenuhi syarat sebagai segitiga. Jika memenuhi, program akan mengklasifikasikan segitiga ke dalam jenis segitiga sama sisi, sama kaki, siku-siku, atau sembarang.

➤ Alur Kode Program

1. Program menerima tiga input bilangan bulat positif dari pengguna.
2. Program memeriksa apakah ketiga sisi memenuhi syarat segitiga.
3. Jika tidak memenuhi syarat, program menampilkan "Bukan segitiga".

4. Jika memenuhi, program mengecek kesamaan panjang sisi untuk menentukan segitiga sama sisi.
5. Program mengecek teorema Pythagoras untuk menentukan segitiga siku-siku.
6. Program mengecek apakah terdapat dua sisi yang sama untuk menentukan segitiga sama kaki.
7. Jika semua sisi berbeda, program menampilkan segitiga sembarang.

➤ **Kesimpulan**

Program ini dapat menentukan jenis segitiga secara tepat berdasarkan input pengguna. Dengan penerapan kondisi logika yang berurutan, program mampu mengklasifikasikan segitiga secara sistematis dan sesuai dengan aturan matematika.