

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

Hanifan Bintang Wiraaji

109082500007

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 a int

    fmt.Print("Masukan ketinggian pohon : ")

    fmt.Scan(&a)

    jarak := (a - 1)

    jarakbatang := (a - 1)

    pohon := 1

    if a < 8 {

        fmt.Print("Masukan ketinggian pohon yang benar(minimal 8) : ")

        fmt.Scan(&a)

    }

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

        for b := (jarak); b > 0; b-- {

            fmt.Print(" ")

        }

        for j := 1; j <= pohon; j++ {

            fmt.Print("*")

        }

    }

}
```

```
}

for c := (jarak); c > 0; c-- {

    fmt.Println(" ")

}

fmt.Println("")

jarak--

pohon += 2

}

for y := 0; y <= 1; y++ {

    for d := jarakbatang; d > 0; d-- {

        fmt.Println(" ")

    }

    fmt.Println("|")

    for f := jarakbatang; f > 0; f-- {

        fmt.Println(" ")

    }

    fmt.Println("")

}

}
```

Screenshot program

The image shows two side-by-side screenshots of the Visual Studio Code (VS Code) interface. Both screenshots display a Go programming environment.

Top Screenshot:

- Explorer:** Shows a project structure under 'ASESMEN2' with files 'Soal1', 'Soal2', 'Soal3', and 'main.go'.
- Code Editor:** Displays the content of 'main.go'. The code prints a diamond shape based on user input for height 'a'. It uses nested loops to print stars ('*') and spaces (' ') to form the diamond pattern.
- Terminal:** Shows the command 'go run "d:\Project\Coding\Go\ngampus\ALPRO\Asesmen2\Soal1"' being run, followed by the output of the diamond pattern.
- Output:** Shows the generated diamond pattern:

```
*
 *
 ***
 ****
 *****
 *****
 ****
 ***
 *
*
```

Bottom Screenshot:

- Explorer:** Shows the same project structure.
- Code Editor:** Displays the same 'main.go' code.
- Terminal:** Shows the command 'go run "d:\Project\Coding\Go\ngampus\ALPRO\Asesmen2\Soal1"' being run, followed by the output of the diamond pattern.
- Output:** Shows the generated diamond pattern:

```

 *
 **
 ***
 ****
 *****
 *****
 ****
 ***
 **
 *
```

The screenshot shows a Go code editor interface with multiple tabs open. The active tab, `main.go Seal1`, contains the following Go code:

```
package main

import "fmt"

func main() {
    var a int
    fmt.Print("Masukan ketinggian pohon : ")
    fmt.Scan(&a)
    jarak := (a - 1)
    jarakbatang := (a - 1)
    pohon := 1
    if a < 8 {
        fmt.Println("Masukan ketinggian pohon yang benar(minimal 8) : ")
        fmt.Scan(&a)
    }
    for i := 0; i < a; i++ {
        for b := (jarak); b > 0; b-- {
            fmt.Print(" ")
        }
        for j := 1; j <= pohon; j++ {
            fmt.Print("*")
        }
        fmt.Println()
    }
}
```

The code is designed to print a triangular pattern of stars based on user input. The terminal window below shows the output for input values of 10 and 7.

TERMINAL PORTS

```
PS D:\Project\Coding\Go\ngampus\ALPRO\Asesmen2> go run "d:\Project\Coding\Go\ngampus\ALPRO\Asesmen2\Seal1"
Masukan ketinggian pohon : 10
*****
 *****
  ****
   ***
    *
  ****
  *****
 *****
```

```
PS D:\Project\Coding\Go\ngampus\ALPRO\Asesmen2>
```

OUTPUT

File Edit View 10908250007

Kelas 7

Hanifan Bintang Wiraaji

Ln 3, Col 8 46 character Plain t 100% Wind UTF-8

Deskripsi program

Program ini untuk membuat pohon natal di terminal kode. User akan diminta input angka untuk tinggi pohon, minimal 8 jika kurang akan ada peringatan dan diminta input ulang. Program akan membuat pohon dengan memberi jarak di kanan dan kiri, kemudian dari baris 1 titik pohon ada dimulai dari 1, kemudian tiap baris akan bertambah 2 titik pohnnya. Program akan membuat variable jarak, ini untuk kanan dan kiri dengan jarak adalah angka yang diinput dikurangin 1, kemudian variable awal untuk pohon adalah 1, seperti penjelasan tadi ini untuk baris 1. Kemudian ada perulangan yang meminta user input angka sama atau lebih dari 8. Kemudian ada perulangan kedua untuk membuat pohnnya. Pertama ada iterasi untuk baris dari 1 sampai angka yang diinput user/tinggi pohon. Kemudian ada perulangan untuk jarak kiri, ini mengulang spasi sebanyak jarak yang tadi sudah di buat. Kemudian ada perulangan lagi kali ini untuk pohon, akan mengeprint * sebanyak titik pohon yang sudah diatur tadi. Kemudian ada perulangan jarak kanan dengan jarak sama dengan yang dikiri. Kemudian aka nada logika untuk print line, dan ada logika untuk mengurangi jarak-- (mengurangi 1 tiap perulangan) dan menambah isi variael titik pohon 2 tiap perulangan. Jika perulangan ini sudah selesai dan sudah jadi bagian atas pohon, ada perulangan lagi yang terakhir untuk batang. Ini dibuat perulangan sebanyak 2 batang dengan jarak kanan kiri sama dengan seting jarak di awal (yang angka user – 1) batangnya menggunakan |.

2. SOAL 2

Source Code

package main

```
import "fmt"

func main() {
    var menu, jumlah, hargaPilih int
    var menuBeli string

    fmt.Println("== Menu Empal Gentong Mas Fuad ==")

    fmt.Println("1. Empal Gentong Biasa - Rp 22.000")
    fmt.Println("2. Empal Asem - Rp 22.000")
    fmt.Println("3. Sate Kambing Muda - Rp 40.000")
    fmt.Println("4. Nasi Lengko Panggonan - Rp 15.000")

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

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

    switch menu {
        case 1:
            menuBeli = "Empal Gentong Biasa"
            hargaPilih = 22000
        case 2:
            menuBeli = "Empal Asem"
            hargaPilih = 22000
        case 3:
```

```
menuBeli = "Sate Kambing Muda"  
  
hargaPilih = 40000  
  
case 4:  
  
    menuBeli = "Nasi Lengko Panggonan"  
  
    hargaPilih = 15000  
  
default:  
  
    fmt.Println("Masukan input yang benar")  
  
}  
  
  
total := jumlah * hargaPilih  
  
fmt.Println("== Struk Pembayaran ==")  
  
fmt.Printf("Menu : %v\n", menuBeli)  
  
fmt.Printf("Harga : Rp %v\n", hargaPilih)  
  
fmt.Printf("Jumlah : %v\n", jumlah)  
  
fmt.Printf("Total : Rp %v\n", total)  
  
}
```

Screenshoot program

The screenshot shows a Microsoft Visual Studio Code (VS Code) window with the following details:

- File Explorer:** Shows a folder structure named "ASESMEN2" containing subfolders "Soal1", "Soal2", and "Soal3", and files "main.go".
- Code Editor:** Displays a Go file named "main.go" with the following code:

```
package main

import "fmt"

func main() {
    var menu, jumlah, hargaPilih int
    var menuBeli string

    fmt.Println("== Menu Empal Gentong Mas Fuad ==")
    fmt.Println("1. Empal Gentong Biasa - Rp 22.000")
    fmt.Println("2. Empal Asem - Rp 22.000")
    fmt.Println("3. Sate Kambing Muda - Rp 40.000")
    fmt.Println("4. Nasi Lengko Panggonan - Rp 15.000")
    fmt.Print("Pilih Menu (1-4): ")
    fmt.Scan(&menu)
    fmt.Print("Masukan jumlah beli: ")
    fmt.Scan(&jumlah)
}
```
- Terminal:** Shows the command "go run d:\Project\Coding\Go\ngampus\ALPRO\Asesmen2>" followed by the program's output:

```
PS D:\Project\Coding\Go\ngampus\ALPRO\Asesmen2> go run d:\Project\Coding\Go\ngampus\ALPRO\Asesmen2> al2/main.go
== Menu Empal Gentong Mas Fuad ==
1. Empal Gentong Biasa - Rp 22.000
2. Empal Asem - Rp 22.000
3. Sate Kambing Muda - Rp 40.000
4. Nasi Lengko Panggonan - Rp 15.000
Pilih Menu (1-4): 2
Masukan jumlah beli: 2
== Struk Pembayaran ==
Menu : Empal Asem
Harga : Rp 22000
Jumlah : 2
Total : Rp 44000
PS D:\Project\Coding\Go\ngampus\ALPRO\Asesmen2>
```
- Output Panel:** Displays the output of the program, which includes the menu options, user input for menu 2 and quantity 2, the receipt structure, and the total amount.
- Status Bar:** Shows the current line (Ln 13), column (Col 56), tab size (Tab Size: 4), and encoding (UTF-8). It also indicates the file is 46 characters long, 100% of the screen width is used, and the encoding is UTF-8.

Deskripsi program

Program ini untuk kasir watung empal gentong. Untuk awalan akan tampil menu dengan keterangan urutan makanan, dan harga, untuk set awal program akan ada variable harga makanan dan makanan yang dibeli. User diminta input menu(dalam urutannya) dan jumlah yang dibeli. Program akan membuat switch case yang dibuat sesuai urutan menu, setelah masuk sesuai menu yang dipilih akan di set variable makanan yang dibeli dan harga makanan sesuai dengan casenya. Setelah selesai akan noda logic total yang mengakumulasi harga makanan dikali jumlah yang dibeli. Kemduian hasil tadi akan di tampilin, makanan yang dibeli, harga, dan totalnya.

3. SOAL 3

Source Code

```
package main

import "fmt"

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

    fmt.Scan(&a, &b, &c)

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

```

} else if a == b || b == c || c == a {

    fmt.Println("Segitiga sama kaki")

} else if ((a*a)+(b*b) == (c * c)) || ((a*a)+(c*c) == (b * b)) || ((c*c)+(b*b) == (a * a)) {

    fmt.Println("Segitiga siku siku")

} else if (a+b <= c) || (a+c <= b) || (b+c <= a) {

    fmt.Println("Bukan segitiga")

} else {

    fmt.Println("Segitiga sembarang")

}
}

```

Screenshoot program

The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project structure for 'ASESMEN2' containing files 'Soal1', 'Soal2', and 'Soal3'. The 'Soal3' folder is expanded, showing three 'main.go' files. The main code editor window shows the content of 'main.go' for 'Soal3'. The terminal window at the bottom shows the execution of the program with various inputs and outputs.

```

File Edit Selection View Go ...
File Explorer Search Problems Output Debug Console Terminal Powerhell
ASESMEN2
  Soal1
    main.go
  Soal2
    main.go
  Soal3
    main.go
    main.go
    main.go

Soal3 > main.go
1 package main
2
3 import "fmt"
4
5 func main() {
6     var a, b, c int
7     fmt.Scan(&a, &b, &c)
8     if a == b && b == c {
9         fmt.Println("Segitiga sama sisi")
10    } else if a == b || b == c || c == a {
11        fmt.Println("Segitiga sama kaki")
12    } else if ((a*a)+(b*b) == (c * c)) || ((a*a)+(c*c) == (b * b)) || ((c*c)+(b*b) == (a * a)) {
13        fmt.Println("Segitiga siku siku")
14    } else if (a+b <= c) || (a+c <= b) || (b+c <= a) {
15        fmt.Println("Bukan segitiga")
16    } else {
17        fmt.Println("Segitiga sembarang")
18    }
19 }
20
D:\Project\Coding\Golangampus\ALPRO\Asesmen2> go run "d:\Project\Coding\Golangampus\ALPRO\Asesmen2\Soal3\main.go"
3 3
Segitiga sama sisi
PS D:\Project\Coding\Golangampus\ALPRO\Asesmen2> go run "d:\Project\Coding\Golangampus\ALPRO\Asesmen2\Soal3\main.go"
5 5 8
Segitiga sama kaki
PS D:\Project\Coding\Golangampus\ALPRO\Asesmen2> go run "d:\Project\Coding\Golangampus\ALPRO\Asesmen2\Soal3\main.go"
3 4 5
Segitiga siku siku
PS D:\Project\Coding\Golangampus\ALPRO\Asesmen2> go run "d:\Project\Coding\Golangampus\ALPRO\Asesmen2\Soal3\main.go"
4 5 6
Segitiga sembarang
PS D:\Project\Coding\Golangampus\ALPRO\Asesmen2> go run "d:\Project\Coding\Golangampus\ALPRO\Asesmen2\Soal3\main.go"
1 2 3
Bukan segitiga
PS D:\Project\Coding\Golangampus\ALPRO\Asesmen2>

```

Deskripsi program

Program ini untuk memeriksa jenis segitiga. User di minta input 3 sisinya(diwakili dengan variable a, b, c. Program akan memeriksa dengan pengkondisian seperti ini;

1. Jika sisi a sama dengan b dan b sama dengan c maka hasilnya adalah ini segitiga sama sisi.
2. Jika a sama dengan b atau b sama dengan c atau c sama dengan a maka hasilnya segitiga sama kaki.

3. Jika $a^2 + b^2 = c^2$ dan begitu pula di sisi yang lain dan logik OR, maka hasilnya segitiga siku sikuut
4. Jika $a + b$ kurang dari sama dengan c , begitu pula di tiga sisi yang lain maka hasilnya ini adalah bukan segitiga
5. Else nya adalah segitiga sembarang.