

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

Rizky Tabriz Deanova

109082500177

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.Scan(&a)

    for i := 1; i <= a; i++ {
        for k := 1; k <= a-i; k++ {
            fmt.Print(" ")
        }
        for j := 1; j <= 2*i-1; j++ {
            fmt.Print("*")
        }
        fmt.Println()
    }

    for i := 2; i <= a/2; i++ {
        for k := 1; k <= a-1; k++ {
            fmt.Print(" ")
        }
        fmt.Println(" | ")
    }
}
```

Screenshot program

Deskripsi program

package main menandakan bahwa program ini dapat dieksekusi/dijalankan/*executable*

`import ("fmt")` berfungsi sebagai pemanggil fungsi atau library “`fmt`” untuk dipakai menjalankan program

func main () {} sebagai titik awal program mulai berjalan

`var` digunakan sebagai wadah atau tempat menyimpan nilai suatu variabel

`fmt.Scan` digunakan untuk membaca input dari user dan akan dimasukkan ke dalam variabel yang terkait di perintah `fmt.Scan`

`fmt.Println` untuk mencetak output atau hasil nilai dari program atau suatu variabel

`fmt.Println` untuk mencetak output atau hasil nilai dari program atau suatu variabel di baris baru

for berfungsi sebagai perulangan suatu program sebanyak kondisi yang tertera

Yang dilakukan program di dalam gambar di atas adalah **Mencetak Simbol Bintang Dan Garis Lurus Dalam Formasi Pohon** yang mana terlihat melalui kode dari **baris 8 hingga baris 21**.

- Terdapat 2 perulangan dengan masing-masing perulangan berfokus pada pencetakan space, simbol bintang, dan garis lurus
 - Perulangan pertama yang berfokus pada pencetak space dan simbol bintang untuk merepresentasikan daun pohon akan terus diulang selama kondisi variabel i kurang dari variabel a
 - Perulangan kedua yang berfokus untuk mencetak space dan garis lurus untuk merepresentasikan batang pohon akan terus diulang selama nilai variabel i kurang dari hasil pembagian variabel a dengan angka 2
 - Selama kondisi variabel i tidak sesuai dengan kondisi dalam perulangan, variabel i akan ditambah sebanyak 1 melalui increment (++)

Runtutan Eksekusi:

1. Menunggu user melakukan input bilangan untuk dimasukkan ke dalam variabel a
2. Perulangan dimulai dengan deklarasi variabel i beserta nilainya yaitu 1 dan perulangan akan terus diulang selama kondisi nilai variabel i kurang dari atau sama dengan nilai variabel a, selama perulangan diulang, maka nilai variabel i akan ditambah sebanyak 1
3. Terdapat perulangan lagi dengan deklarasi variabel k beserta nilainya sebanyak 1, perulangan akan terus diulang selama nilai variabel k kurang dari atau sama dengan dari hasil pengurangan variabel a dan i, selama perulangan diulang, nilai variabel k akan ditambah sebanyak 1
4. Jika kondisi variabel i dan k sesuai dengan ketentuan perulangan, maka akan dicetak space dan simbol bintang
5. Pencetakan garis baru
6. Dilakukan perulangan dengan deklarasi nilai i sebanyak 2 dan perulangan akan terus diulang selama nilai variabel i kurang dari atau sama dengan hasil pembagian variabel a dengan angka 2, selama perulangan terus diulang, variabel i akan bertambah nilainya sebanyak 1
7. Terdapat perulangan lagi dengan deklarasi variabel k beserta nilainya sebanyak 1 dan perulangan akan terus diulang selama kondisi variabel k kurang dari atau sama dengan dari hasil pengurangan variabel a dan angka 1, selama perulangan terus diulang, maka variabel k nilainya bertambah sebanyak 1
8. Jika kondisi variabel i dan k sesuai, maka akan dicetak space
9. Pencetakan garis lurus (|)

2. SOAL 2

Source Code

```
package main

import "fmt"

func main() {

    var menu string

    var porsi int

    fmt.Print("Pilih menu: ")

    fmt.Scan(&menu)

    fmt.Print("Banyaknya porsi: ")

    fmt.Scan(&porsi)

    switch menu {

        case "LittleTrees":
```

```

        menu = "1"

    case "LapMicrofiber":

        menu = "2"

    case "CoverSteer":

        menu = "3"

    case "SpongeCuciMobil":

        menu = "4"

default:

    fmt.Println("Menu tidak tersedia")

    return

}

fmt.Println("-----")

fmt.Println("Pilih produk (1-4): ", menu)

fmt.Println("Banyaknya porsi: ", porsi)

}

```

Screenshot program

The screenshot shows a Go development environment in Visual Studio Code. The code editor displays a main.go file with the following content:

```

package main
import "fmt"
func main() {
    var menu string
    var porsi int
    fmt.Print("Pilih menu: ")
    fmt.Scan(&menu)
    fmt.Print("Banyaknya porsi: ")
    fmt.Scan(&porsi)
    switch menu {
    case "LittleFree":
        menu = "1"
    case "LapMicrofiber":
        menu = "2"
    case "CoverSteer":
        menu = "3"
    case "SpongeCuciMobil":
        menu = "4"
    default:
    }
}

```

The terminal window shows the execution of the program and user interaction:

```

PS D:\dea - uni\main\subject\alpro praktik\go lang\uas> go run "D:\dea - uni\main\subject\alpro praktik\go lang\uas\s2\s2.go"
pilih menu: LapMicrofiber
banyaknya porsi: 5
-----
Pilih produk (1-4): 2
Banyaknya porsi: 5
Menu tidak tersedia
PS D:\dea - uni\main\subject\alpro praktik\go lang\uas>

```

The file explorer shows several Go files in a directory named 'UAS'.

Deskripsi program

```
package main menandakan bahwa program ini dapat dieksekusi/dijalankan/executable
import ("fmt") berfungsi sebagai pemanggil fungsi atau library "fmt" untuk dipakai menjalankan program
func main () {} sebagai titik awal program mulai berjalan
var digunakan sebagai wadah atau tempat menyimpan nilai suatu variabel
fmt.Scan digunakan untuk membaca input dari user dan akan dimasukkan ke dalam variabel yang terkait di perintah fmt.Scan
fmt.Print untuk mencetak output atau hasil nilai dari program atau suatu variabel
fmt.Println untuk mencetak output atau hasil nilai dari program atau suatu variabel di baris baru
switch berfungsi sebagai pengecekan kondisi dengan beberapa kasus yang sudah terdaftar
case digunakan untuk menjadi wadah atau tempat suatu kasus atau kondisi dengan lanjutan program yang ada di dalamnya jika input user sesuai dengan suatu case
default sebagai pengecekan kondisi yang mana dapat dibilang sebagai kondisi pengecualian jika kondisi input tidak sesuai dengan seluruh case
return digunakan untuk menghentikan jalannya program jika suatu input sesuai dengan kondisi default atau di luar case
```

Yang dilakukan program di dalam gambar di atas adalah **Mencetak Informasi Produk Dan Porsi Yang Dipesan** yang mana terlihat melalui kode dari **baris 12 hingga baris 27**.

- Terdapat 4 kasus yang berfungsi untuk mewakili nama menu dengan angka
- Perintah default digunakan sebagai pengecekan kondisi apabila menu yang dipesan tidak sesuai dengan seluruh kasus yang ada dalam program
- Output menu: Ditampilkan dengan informasi menu yang dipilih dalam representasi angka. Permisalan output, jika memesan LittleTrees, input LittleTrees akan dimasukkan ke dalam variabel menu dengan angka 1 atau dapat dibilang input LittleTrees akan diubah menjadi angka 1 dan dimasukkan ke dalam variabel menu
- Output porsi: Menampilkan informasi banyaknya porsi menu yang dipesan oleh pelanggan dalam bentuk angka. Untuk porsi sendiri tidak ada konversi atau pengubahan nilai ke suatu nilai lain atau tipe data lain

Runtutan Eksekusi:

10. Menampilkan output "Pilih menu: " dan program akan menunggu user untuk input data menu apa yang dipilih, kemudian akan dimasukkan ke dalam variabel menu
11. Menampilkan output "Banyaknya porsi: " dan program akan menunggu user untuk input banyaknya porsi yang diinginkan, kemudian akan dimasukkan ke dalam variabel porsi
12. Memulasi pengecekan kondisi di mana input menu akan diperiksa kesesuaianya dengan kasus yang mana. Jika terdapat kasus yang cocok atau sesuai, maka menu tersebut akan diubah menjadi representasi angka dari kasus tersebut

13. Jika tidak sesuai dengan kasus yang tercantum, maka akan masuk ke perintah default yang mana dijalankan apabila menu yang diinput oleh user tidak sesuai dengan kasus atau tidak terdapat dalam menu. Setelahnya program akan dihentikan oleh perintah return
14. Mencetak garis strip (-)
15. Mencetak "Pilih produk (1-4): " yang dilanjutkan hasil dari pemeriksaan menu dalam bentuk angka di dalam variabel menu
16. Mencetak "Banyaknya porsi: " yang dilanjutkan hasil dari nilai variabel porsi

3. SOAL 3

Source Code

```

package main

import "fmt"

func main() {

    var a, b, c int

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

    if a <= 0 || b <= 0 || c <= 0 || a+b <= c || a+c <=
    b || b+c <= a {

        fmt.Println("Bukan segitiga")

        return
    }

    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

The screenshot shows a Visual Studio Code interface with several tabs open in the background: s1.go, s2.go, s3.go, trys2.go, and trys1.go. The current file being edited is s3.go, which contains a Go program for triangle classification. The terminal window on the right displays the output of running the program with various input values (e.g., 3 3 3, 5 5 8, 3 4 5, 4 5 6, 1 2 3). The terminal also shows the student's name and ID: rizky tabriz deanova 109082500177. The status bar at the bottom indicates the line number (Ln 12, Col 1), tab size (Tab Size: 4), and file encoding (UTF-8).

Deskripsi program

package main menandakan bahwa program ini dapat dieksekusi/dijalankan/executable

import ("fmt") berfungsi sebagai pemanggil fungsi atau library "fmt" untuk dipakai menjalankan program

func main () {} sebagai titik awal program mulai berjalan

var digunakan sebagai wadah atau tempat menyimpan nilai suatu variabel

fmt.Scan digunakan untuk membaca input dari user dan akan dimasukkan ke dalam variabel yang terkait di perintah fmt.Scan

fmt.Println untuk mencetak output atau hasil nilai dari program atau suatu variabel di baris baru

if berfungsi sebagai pemeriksaan kondisi dengan ketentuan yang terdapat di dalamnya
else if digunakan untuk pemeriksaan kondisi juga apabila kondisi input tidak sesuai dengan if

else sebagai pengecekan kondisi yang mana dijalankan apabila input tidak sesuai dengan seluruh kondisi else if ataupun if

Yang dilakukan program di dalam gambar di atas adalah **Memeriksa Jenis Segitiga Berdasarkan Keseuaian Input, Rumus, Dan Hasil** yang mana terlihat melalui kode dari **baris 8 hingga baris 19**.

- Terdapat 5 kondisi yang berfungsi untuk mewakili masing-masing jenis segitiga

Runtutan Eksekusi:

1. Menunggu user untuk melakukan input integer sebanyak 3 untuk dimasukkan berurutan ke dalam variabel a, b, dan c
2. Pemeriksaan kondisi pertama yang mana kondisi di dalam if mewakili kondisi input yang tidak sesuai untuk membangun segitiga. Jika sesuai dengan kondisi dalam if ini, maka akan dimunculkan output “Bukan segitiga”
3. Pemeriksaan kondisi kedua untuk jenis segitiga sama sisi di mana nilai variabel a sama dengan b dan nilai variabel b sama dengan c. Jika sesuai dengan kondisi dalam if ini, maka akan dimunculkan output “Segitiga sama kaki”
4. Pemeriksaan kondisi ketiga untuk jenis segitiga sama kaki bila nilai variabel a sama dengan b, nilai variabel a sama dengan c, dan nilai variabel b sama dengan c. Jika sesuai dengan kondisi dalam if ini, maka akan dimunculkan output “Segitiga siku-siku”
5. Pemeriksaan kondisi keempat untuk jenis segitiga sembarang yang mana inputnya tidak sesuai untuk jenis segitiga lainnya, tetapi masih dapat membentuk segitiga. Jika sesuai dengan kondisi dalam if ini, maka akan dimunculkan output “Segitiga sembarang”