

LAPORAN PRAKTIKUM
Algoritma Pemrograman

EVALUASI



Disusun oleh:

Nayaka Farros Riandra

109082500055

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 ketinggian piramida = ")

    fmt.Scan(&n)

    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()
    }
}
```

Screenshot program

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

- EXPLORER:** Shows project structure under "PRAKTIKUM ALGORITMA".
- CODE EDITOR:** Displays the code for "soal1.go" (version 2). The code prints a pyramid of asterisks based on user input.
- TERMINAL:** Shows the command "PS D:\tugas kuliah\Praktikum Algoritma> go run "d:\tugas kuliah\Praktikum Algoritma\UAS ALPRO> soal1.go 1 main". The output is a pyramid with 8 rows of asterisks.
- OUTPUT:** Shows the printed pyramid pattern.
- PROBLEMS:** Shows 16 errors related to undeclared variables.
- RIGHT SIDE:** A terminal window showing student information: NAMA : NAYAKA FARROS RIANDRA, KELAS : S1IF-13-07, and NIM : 109082500055.

Deskripsi program

Program ini Adalah program untuk membuat piramida Bintang dengan ketinggian yang kita inginkan. Pertama-tama kita masukkan ketinggian piramida yang kita inginkan, misal 10. Maka program akan menyimpan nilai n sebagai 10, lalu masuk ke for Dimana kita tentukan nilai default i = 1, lalu di cek Ketika I kurang dari sama dengan n maka nilai i akan bertambah hingga nilainya sama dengan n.

selanjutnya di dalam perulangan i tersebut, terdapat dua proses (loop) bertingkat. **Pertama**, program menjalankan perulangan j dari 1 hingga n-i untuk mencetak spasi (" "). Fungsi ini berguna untuk memberi jarak agar bintang tersusun rapi di tengah (rata tengah). **Kedua**, program menjalankan perulangan k dari 1 hingga $(2*i)-1$. Rumus ini digunakan untuk mencetak karakter bintang (*) dengan jumlah ganjil (1, 3, 5, dst.) sesuai urutan barisnya, sehingga membentuk pola segitiga sama kaki atau piramida. Setelah bintang pada baris tersebut selesai dicetak, perintah `fmt.Println()` dieksekusi untuk memindahkan kursor ke baris baru. Proses ini akan berulang terus menerus hingga nilai i mencapai batas n yang telah ditentukan.

2. SOAL 2

Source Code

```
package main

import "fmt"

func main() {
    var pilihan, jumlah int

    fmt.Println("==== MENU EMPAL GENTONG MAS FUAD ====")
    fmt.Println("1. Empal Gentong Biasa - Rp22.000")
    fmt.Println("2. Empal Asem - Rp22.000")
    fmt.Println("3. Sate Kambing Muda - Rp40.000")
    fmt.Println("4. Nasi Lengko Pagongan - Rp15.000")

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

    fmt.Print("beli berapa? : ")
    fmt.Scan(&jumlah)

    var namaMenu string
    var harga int

    switch pilihan {
        case 1:
            namaMenu = "Empal Gentong Biasa"
            harga = 22000
        case 2:
            namaMenu = "Empal Asem"
```

```
        harga = 22000

    case 3:

        namaMenu = "Sate Kambing Muda"

        harga = 40000

    case 4:

        namaMenu = "Nasi Lengko Pagongan"

        harga = 15000

    default:

        fmt.Println("Menu tidak valid!")

        return

    }

total := harga * jumlah

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

fmt.Println("Menu yang dipilih: ", namaMenu)

fmt.Println("Harga per item: ", harga)

fmt.Println("Jumlah beli: ", jumlah)

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

}
```

Screenshot program

The screenshot shows a Go code editor interface with two tabs open: `soal1.go` and `soal2.go`. The `soal2.go` tab is active, displaying the following Go code:

```
package main

import "fmt"

func main(){
    var pilahan, jumlah int

    fmt.Println("== MENU EMPAL GENTONG MAS FUAD ==")
    fmt.Println("1. Empal Gentong Biasa - Rp22.000")
    fmt.Println("2. Empal Asem - Rp22.000")
    fmt.Println("3. Sate Kambing Muda - Rp40.000")
    fmt.Println("4. Nasi Lengko Pagongan - Rp15.000")
    fmt.Print("Pilih menu cuyy (1-4): ")
    fmt.Scan(&pilihan)
    fmt.Print("beli berapa? : ")
    fmt.Scan(&jumlah)

    var namaMenu string
    var harga int

    switch pilahan {
    case 1:
        namaMenu = "Empal Gentong Biasa"
        harga = 22000
    case 2:
        namaMenu = "Empal Asem"
        harga = 22000
    case 3:
        namaMenu = "Sate Kambing Muda"
        harga = 40000
    case 4:
        namaMenu = "Nasi Lengko Pagongan"
        harga = 15000
    default:
        fmt.Println("Menu tidak valid!")
        return
    }

    total := harga * jumlah
    fmt.Println("== STRUK PEMBAYARAN ==")
    fmt.Println("Menu yang dipilih: ", namaMenu)
    fmt.Println("Harga per item: ", harga)
    fmt.Println("Jumlah beli: ", jumlah)
    fmt.Println("Total pembayaran: ", total)
}
```

The output window shows the execution of the program:

```
PS D:\tugas kuliah\Praktikum Algoritma> go run "d:\tugas kuliah\Praktikum Algoritma\UAS ALPRO\tempCodeRunnerFile.go"
== MENU EMPAL GENTONG MAS FUAD ==
1. Empal Gentong Biasa - Rp22.000
2. Empal Asem - Rp22.000
3. Sate Kambing Muda - Rp40.000
4. Nasi Lengko Pagongan - Rp15.000
Pilih menu cuyy (1-4): 2
beli berapa? : 2
== STRUK PEMBAYARAN ==
Menu yang dipilih: Empal Asem
```

This screenshot shows the same Go code editor interface, but the menu options in the code have been modified. The `soal2.go` tab is active, displaying the following Go code:

```
package main

import "fmt"

func main(){
    var pilahan, jumlah int

    fmt.Println("== MENU EMPAL GENTONG MAS FUAD ==")
    fmt.Println("1. Empal Gentong Biasa - Rp22.000")
    fmt.Println("2. Empal Asem - Rp22.000")
    fmt.Println("3. Sate Kambing Muda - Rp40.000")
    fmt.Println("4. Nasi Lengko Pagongan - Rp15.000")
    fmt.Print("Pilih menu cuyy (1-4): ")
    fmt.Scan(&pilihan)
    fmt.Print("beli berapa? : ")
    fmt.Scan(&jumlah)

    var namaMenu string
    var harga int

    switch pilahan {
    case 2:
        namaMenu = "Empal Asem"
        harga = 22000
    case 3:
        namaMenu = "Sate Kambing Muda"
        harga = 40000
    case 4:
        namaMenu = "Nasi Lengko Pagongan"
        harga = 15000
    default:
        fmt.Println("Menu tidak valid!")
        return
    }

    total := harga * jumlah
    fmt.Println("== STRUK PEMBAYARAN ==")
    fmt.Println("Menu yang dipilih: ", namaMenu)
    fmt.Println("Harga per item: ", harga)
    fmt.Println("Jumlah beli: ", jumlah)
    fmt.Println("Total pembayaran: ", total)
}
```

The output window shows the execution of the program with the changes:

```
PS D:\tugas kuliah\Praktikum Algoritma> go run "d:\tugas kuliah\Praktikum Algoritma\UAS ALPRO\tempCodeRunnerFile.go"
== MENU EMPAL GENTONG MAS FUAD ==
1. Empal Gentong Biasa - Rp22.000
2. Empal Asem - Rp22.000
3. Sate Kambing Muda - Rp40.000
4. Nasi Lengko Pagongan - Rp15.000
Pilih menu cuyy (1-4): 2
beli berapa? : 2
== STRUK PEMBAYARAN ==
Menu yang dipilih: Empal Asem
Harga per item: 22000
Jumlah beli: 2
Total pembayaran: 44000
```

Deskripsi program

Program ini merupakan aplikasi sederhana berbasis teks yang ditulis dalam bahasa Go untuk mensimulasikan sistem pemesanan di rumah makan "Empal Gentong Mas Fuad". Proses dimulai dengan program mendeklarasikan variabel yang diperlukan, lalu menampilkan daftar menu makanan beserta harganya ke layar terminal agar dapat dibaca oleh pengguna. Setelah menu ditampilkan, program meminta pengguna untuk memasukkan dua data input, yaitu nomor pilihan menu (1-4) dan jumlah porsi yang ingin dibeli. Input tersebut kemudian ditangkap oleh fungsi fmt.Scan dan disimpan ke dalam variabel pilihan dan jumlah. Selanjutnya, program memproses input tersebut menggunakan struktur kontrol switch. Di dalam blok ini, program mencocokkan angka pilihan pengguna dengan daftar kasus (case) yang tersedia untuk menentukan nama menu dan harga satuan yang tepat. Akhirnya, program akan menampilkan "Struk Pembayaran" yang berisi rincian menu yang telah dipilih sebagai konfirmasi pesanan.

3. SOAL 3

Source Code

```
package main

import "fmt"

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

    fmt.Print("masukkan sisi a = ")
    fmt.Scan(&a)

    fmt.Print("masukkan sisi b = ")
    fmt.Scan(&b)

    fmt.Print("masukkan sisi c = ")
    fmt.Scan(&c)

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

    fmt.Println("Bukan segitiga")

} else {

    fmt.Println("segitiga sembarang")

}

}
```

Screenshot program

The screenshot shows a Go code editor interface with two windows. The main window displays a Go program named `soal3.go` which reads three integers `a`, `b`, and `c` from the user and prints out whether they form a right-angled triangle or not. The code includes logic for checking if any two sides squared sum up to the third side squared. The terminal window on the right shows the execution of the program and its output:

```
NAMA : NAYAKA FARROS RIANDRA  
KELAS : S1IF-13-07  
NIM : 109082500055
```

This screenshot is identical to the one above, showing the same Go code editor interface with the `soal3.go` file open. The terminal window on the right shows the execution of the program and its output:

```
NAMA : NAYAKA FARROS RIANDRA  
KELAS : S1IF-13-07  
NIM : 109082500055
```

Deskripsi program

Program ini berfungsi untuk mengidentifikasi jenis segitiga berdasarkan panjang ketiga sisinya yang diinputkan oleh pengguna. Pertama, program meminta pengguna memasukkan tiga nilai bilangan bulat yang merepresentasikan sisi a, b, dan c. Setelah data diterima, program melakukan serangkaian pengecekan kondisi menggunakan struktur percabangan if-else. Logika pengecekan dimulai dengan memeriksa apakah ketiga sisi sama panjang (segitiga sama sisi), kemudian memeriksa apakah ada dua sisi yang sama (segitiga sama kaki), serta memeriksa apakah sisi-sisinya memenuhi rumus Pythagoras (segitiga siku-siku). Program juga memiliki validasi untuk memastikan apakah ketiga sisi tersebut valid membentuk segitiga; jika jumlah dua sisi lebih kecil atau sama dengan sisi lainnya, program akan menyatakan "Bukan segitiga". Jika semua kondisi di atas tidak terpenuhi, program akan menyimpulkan bahwa bentuk tersebut adalah segitiga sembarang.