

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
MODUL 2  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**Muhammad Hamzah Haifan Ma'ruf**

**2311102091**

**S1-IF-11-05**

**Dosen Pengampu :**

**Arif Amrulloh, S.Kom.,M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## **I. DASAR TEORI**

Go atau Golang adalah bahasa pemrograman open-source yang dikembangkan oleh Google.

Ketika Google mulai berkembang pesat, engineer Google merasa bahasa C++ terlalu rumit untuk dikelola. Kemudian mereka (Robert Griesemer, Rob Pike, dan Ken Thompson) berinisiatif mengembangkan bahasa yang mudah dikelola dan dipelajari. Bahasa baru inilah yang dinamakan Go.

Developer biasanya menggunakan Golang dalam berbagai sistem operasi dan framework untuk mengembangkan web, layanan cloud dan jaringan, serta jenis perangkat lunak lainnya.

Go menjadi salah satu bahasa pemrograman paling populer karena program diketik secara statis, bukan secara dinamis seperti JavaScript dan Python. Program yang diketik statis tidak akan berjalan sampai kesalahan sudah diperbaiki. Berbeda dengan bahasa yang diketik secara dinamis tetap bisa dijalankan bahkan jika ada error.

### **- If Statement**

Statement if digunakan untuk mengevaluasi kondisi dan menjalankan blok kode jika kondisinya benar (true). Go mendukung beberapa jenis loop, tetapi yang paling umum adalah loop for. Loop for digunakan untuk mengulangi blok kode selama kondisi yang ditentukan benar. Anda juga dapat menggunakan for untuk melakukan loop melalui elemen-elemen dalam sebuah slice atau map

### **- Switch Statement**

Pernyataan switch digunakan untuk memeriksa beberapa kondisi dan menjalankan blok kode yang sesuai dengan kondisi yang terpenuhi. Anda juga dapat menggunakan switch tanpa ekspresi untuk membuat struktur percabangan yang lebih bersih.

## II. GUIDED

### 1. Guided 1 - Soal Nama

#### Soal Studi Case

Membuat sebuah program sederhana yang meminta pengguna memasukkan nama mereka, lalu mencetak nama tersebut ke layar. Program ini dirancang menggunakan bahasa pemrograman Go dan bertujuan untuk memahami dasar penggunaan input/output serta variabel.

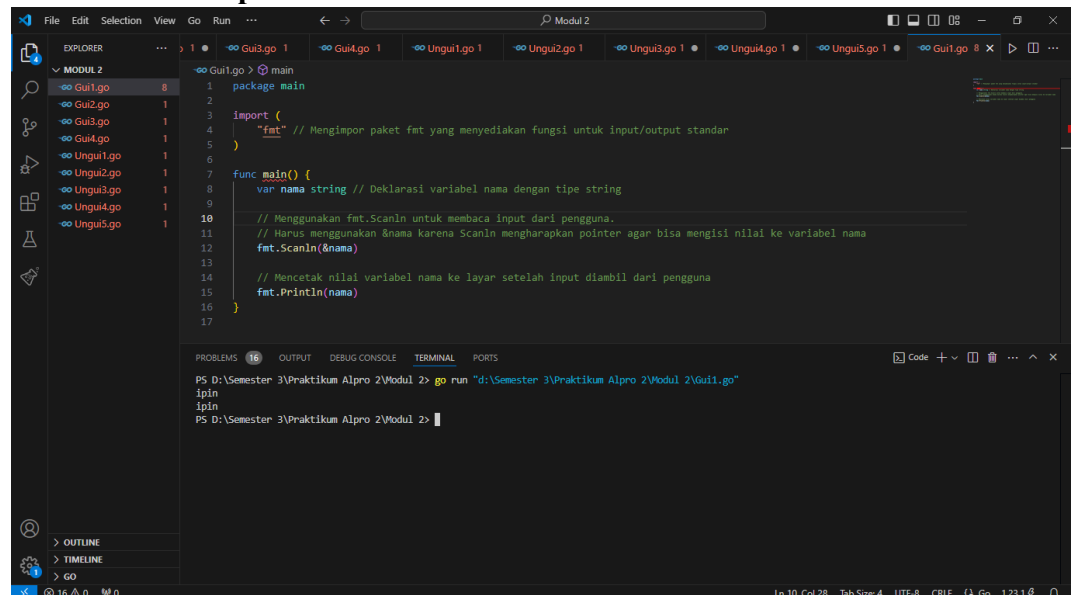
#### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var nama string
    fmt.Scanln(&nama)
    fmt.Println(nama)
}
```

#### Screenshoot Output



#### Deskripsi Program

Berikut adalah penjelasan singkat dari aplikasi Go dasar yang menerima input pengguna dan menampilkannya kembali ke layar. Paket "main" menunjukkan bahwa file ini adalah file utama yang akan dijalankan. Paket "fmt" diimpor agar dapat melakukan fungsi I/O seperti membaca input dan menampilkan output. Untuk menyimpan input pengguna, sebuah variabel bernama "nama" dibuat dengan tipe data "string". Fungsi "fmt.Scanln()" membaca input pengguna dan menyimpannya ke dalam variabel "nama", dan fungsi "fmt.Println()" menampilkan nilai dari variabel "nama" ke layar, yang juga mengandung input yang diberikan oleh pengguna.

## 2. Guided 2 – 2B – 1

### Soal Studi Case

Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berturutan adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang.

Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan true apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

### Sourcecode

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    // Urutan warna yang benar
```

```

    correctOrder := []string{"merah", "kuning",
    "hijau", "ungu"}

    // Membaca input untuk 5 percobaan
    reader := bufio.NewReader(os.Stdin)
    success := true

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)

        // Membaca input dari pengguna
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        // Memisahkan input berdasarkan spasi
        colors := strings.Split(input, " ")

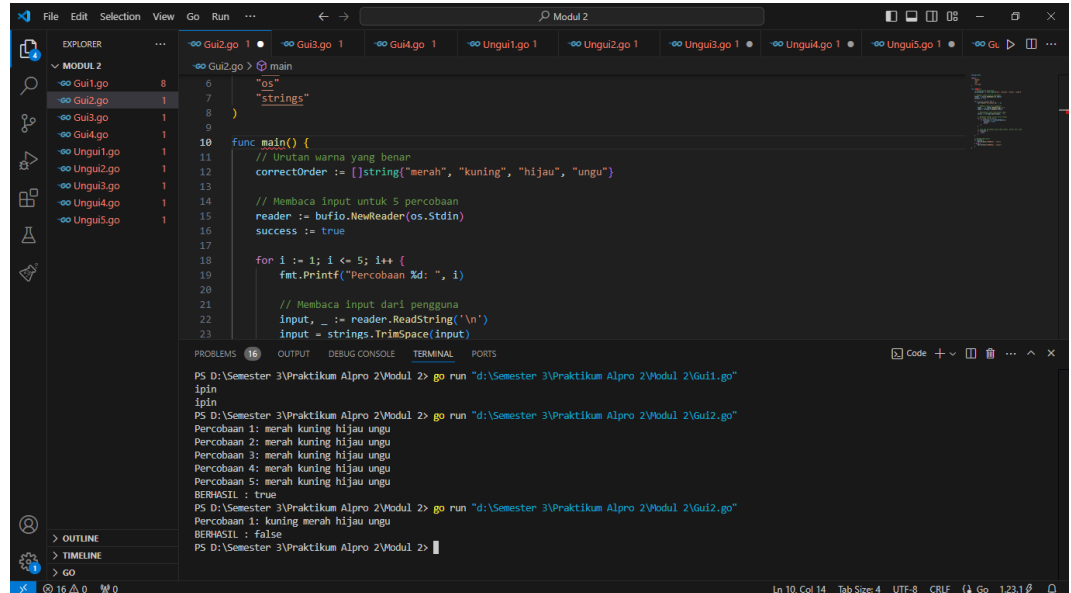
        // Mengecek apakah urutan warna sesuai
        for j := 0; j < 4; j++ {
            if colors[j] != correctOrder[j] {
                success = false
                break
            }
        }

        // Jika ada percobaan yang tidak sesuai,
        keluar dari loop
        if !success {
            break
        }
    }

    // Menampilkan hasil
    if success {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    }
}

```

## Screenshoot Output



```
6  "os"
7  "strings"
8  )
9
10 func main() {
11     // Urutan warna yang benar
12     correctOrder := []string{"merah", "kuning", "hijau", "ungu"}
13
14     // Membaca input untuk 5 percobaan
15     reader := bufio.NewReader(os.Stdin)
16     success := true
17
18     for i := 1; i <= 5; i++ {
19         fmt.Printf("Percobaan %d: ", i)
20
21         // Membaca input dari pengguna
22         input, _ := reader.ReadString('\n')
23         input = strings.TrimSpace(input)
```

```
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui1.go"
ipin
ipin
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui2.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL : true
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui2.go"
Percobaan 1: kuning merah hijau ungu
BERHASIL : false
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> |
```

## Deskripsi Program

Deskripsi program ini mendefinisikan urutan warna yang benar dalam potongan "Order yang tepat", yaitu "merah", "kuning", "hijau", dan "ungu". Selama lima kali, pengguna diminta untuk memasukkan urutan warna. Program menggunakan "bufio.NewReader" untuk membaca input pengguna satu baris sekaligus. Pada setiap tes, input dibaca dan dibersihkan dari karakter newline ("\n") menggunakan "strings.TrimSpace". Kemudian, menggunakan "strings.Split", input dipisahkan menjadi elemen-elemen warna berdasarkan spasi, dan program memeriksa apakah urutan warna yang dimasukkan sesuai dengan urutan yang benar dalam "correctOrder". Program keluar dari loop dan menyatakan percobaan gagal jika salah satu percobaan memiliki urutan yang tidak sesuai. Program menampilkan hasil akhir setelah lima percobaan atau jika ada ketidaksesuaian. Program mencetak "BERHASIL: true" jika semua input benar, sedangkan jika tidak, program mencetak "BERHASIL: false".

### 3. Guided – 3 – Soal abcd Soal Studi Case

Membuat program sederhana yang memungkinkan pengguna memasukkan 5 angka integer, kemudian program menjumlahkan kelima angka tersebut dan menampilkan hasilnya.

### Sourcecode

```
package main

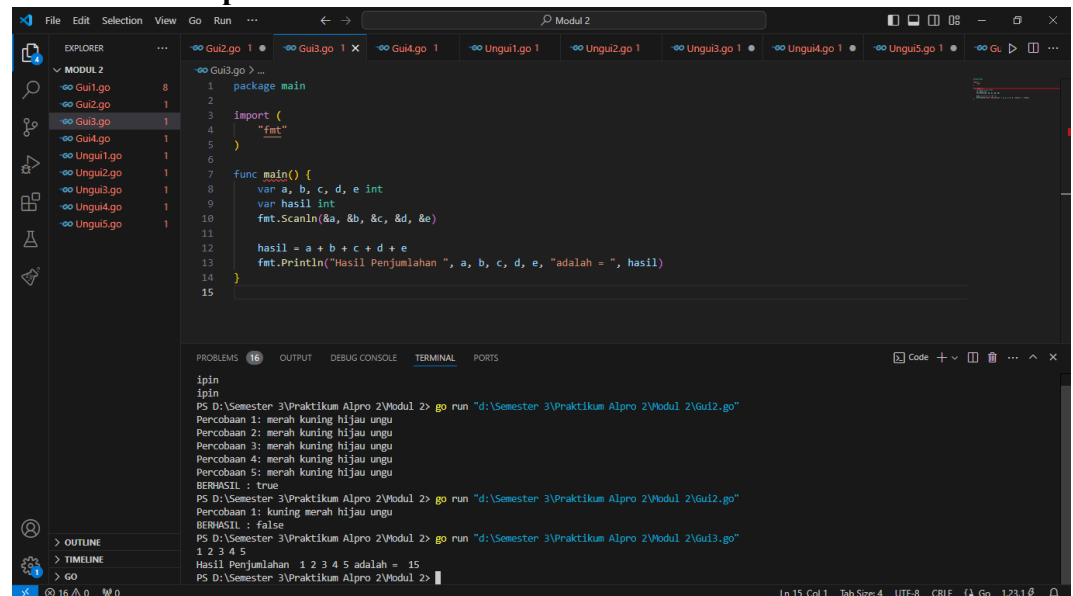
import "fmt"

func main() {

    var a, b, c, d, e int
    var hasil int
    fmt.Scanln(&a, &b, &c, &d, &e)

    hasil = a + b + c + d + e
    fmt.Println("Hasil Penjumlahan ", a, b, c,
d, e, "adalah = ", hasil)
}
```

### Screenshot Output

The screenshot shows an IDE with a Go file named 'Gui2.go' open. The code in the editor is the same as the source code provided. The terminal at the bottom shows the execution of the program. It starts with a prompt 'ipin' and then 'PS D:\Semester 3\Praktikum Alpro 2\Vmodul 2> go run "d:\Semester 3\Praktikum Alpro 2\Vmodul 2\Gui2.go"'. The output shows five prompts for input: 'Percobaan 1: merah kuning hijau ungu', 'Percobaan 2: merah kuning hijau ungu', 'Percobaan 3: merah kuning hijau ungu', 'Percobaan 4: merah kuning hijau ungu', and 'Percobaan 5: merah kuning hijau ungu'. The program then outputs 'BERHASIL : true' and 'PS D:\Semester 3\Praktikum Alpro 2\Vmodul 2> go run "d:\Semester 3\Praktikum Alpro 2\Vmodul 2\Gui2.go"'. The output shows 'Percobaan 1: kuning merah hijau ungu', 'BERHASIL : false', and 'PS D:\Semester 3\Praktikum Alpro 2\Vmodul 2> go run "d:\Semester 3\Praktikum Alpro 2\Vmodul 2\Gui3.go"'. The output shows '1 2 3 4 5', 'Hasil Penjumlahan 1 2 3 4 5 adalah = 15', and 'PS D:\Semester 3\Praktikum Alpro 2\Vmodul 2> '.

### Deskripsi Program

Deskripsi program ini mendefinisikan urutan warna yang benar dalam potongan "Order yang tepat", yaitu "merah", "kuning", "hijau", dan "ungu". Selama lima kali, pengguna diminta untuk memasukkan urutan warna. Program menggunakan "bufio.NewReader" untuk membaca input

pengguna satu baris sekaligus. Pada setiap tes, input dibaca dan dibersihkan dari karakter newline ("n") menggunakan "strings.TrimSpace". Kemudian, menggunakan "strings.Split", input dipisahkan menjadi elemen-elemen warna berdasarkan spasi, dan program memeriksa apakah urutan warna yang dimasukkan sesuai dengan urutan yang benar dalam "correctOrder". Program keluar dari loop dan menyatakan percobaan gagal jika salah satu percobaan memiliki urutan yang tidak sesuai. Program menampilkan hasil akhir setelah lima percobaan atau jika ada ketidaksesuaian. Program mencetak "BERHASIL: true" jika semua input benar, sedangkan jika tidak, program mencetak "BERHASIL: false".

#### 4. Guided 4

##### Soal Studi Case

Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

NAM	NMK
NAM > 80	A
72.5 < NAM <= 80	AB

65 < NAM <= 72.5	B
57.5 < NAM <= 65	BC
50 < NAM <= 57.5	C
40 < NAM <= 50	D
NAM <= 40	E

##### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var nilai float32
    var indeks string
```



```
// Meminta input nilai
fmt.Print("Masukkan nilai: ")
fmt.Scan(&nilai)

// Logika penentuan nilai huruf berdasarkan
nilai numerik
if nilai >= 80 {
    indeks = "A"
} else if nilai >= 70 {
    indeks = "B"
} else if nilai >= 65 {
    indeks = "C"
} else if nilai >= 45 {
    indeks = "D"
} else if nilai >= 40 {
    indeks = "E"
} else {
    indeks = "F"
}

// Menampilkan hasil
fmt.Printf("Nilai Indeks untuk nilai %.2f
adalah %s\n", nilai, indeks)
}
```

### Screenshot Output

```
7 func main() {
13     fmt.Scan(&nilai)
14
15     // Logika penentuan nilai huruf berdasarkan nilai numerik
16     if nilai >= 80 {
17         indeks = "A"
18     } else if nilai >= 70 {
19         indeks = "B"
20     } else if nilai >= 65 {
21         indeks = "C"
22     } else if nilai >= 45 {
23         indeks = "D"
24     } else if nilai >= 40 {
25         indeks = "E"
26     } else {
27         indeks = "F"
28     }
29 }
```

PROBLEMS (16) OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui3.go"  
1 2 3 4 5  
Hasil Penjumlahan 1 2 3 4 5 adalah = 15  
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui4.go"  
Masukkan nilai: 80.1  
Nilai Indeks untuk nilai 80.10 adalah A  
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui4.go"  
Masukkan nilai: 93.5  
Nilai Indeks untuk nilai 93.50 adalah A  
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui4.go"  
Masukkan nilai: 70.6  
Nilai Indeks untuk nilai 70.60 adalah B  
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> go run "d:\Semester 3\Praktikum Alpro 2\Vbdl 2\Gui4.go"  
Masukkan nilai: 49.5  
Nilai Indeks untuk nilai 49.50 adalah D  
PS D:\Semester 3\Praktikum Alpro 2\Vbdl 2> |

## Deskripsi Program

- A. Jika nilai diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?

Program menampilkan hasil eksekusi yang sesuai dengan persyaratan, dengan nilai 80.1 lebih besar dari 80, yang menghasilkan nilai A.

- B. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!

Masalah Batas Nilai: Ada kesalahan dalam menentukan rentang nilai; misalnya, pada kondisi " $nam > 80$ ", nilai 80.0 tidak termasuk dalam kategori "A". Untuk menyelesaikan masalah ini, Anda dapat menggunakan operator " $\geq$ " untuk memastikan bahwa batas bawah juga tercakup dengan benar, misalnya, nilai 80.0 harus termasuk dalam kategori "A". Penggunaan Float32: Tidak perlu menggunakan tipe data float32. Sebaliknya, gunakan float64, yang lebih umum dan lebih presisi. Alur Program yang Tepat: Nilai \*nam\* harus diproses dengan rentang yang benar, dengan operator  $\geq$  digunakan untuk batas bawah setiap kategori. Misalnya, alur program seharusnya dimulai dengan memeriksa nilai tertinggi (A), kemudian secara bertahap memeriksa nilai-nilai yang lebih rendah hingga mencapai nilai "F".

- C. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

=

```
package main

import (
    "fmt"
)

func main() {
    var nilai float32
    var indeks string

    // Meminta input nilai
    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&nilai)

    // Logika penentuan nilai huruf berdasarkan
    nilai numerik
    if nilai >= 80 {
        indeks = "A"
    } else if nilai >= 70 {
        indeks = "B"
    } else if nilai >= 65 {
        indeks = "C"
    } else if nilai >= 45 {
        indeks = "D"
    } else if nilai >= 40 {
        indeks = "E"
    } else {
        indeks = "F"
    }

    // Menampilkan hasil
    fmt.Printf("Nilai Indeks untuk nilai %.2f
    adalah %s\n", nilai, indeks)
}
```

Nilai numerik dapat dikonversi menjadi indeks huruf menggunakan program yang disebutkan di atas. Ini dilakukan dengan mengatur rentang nilai tertentu. Pertama, pengguna diminta untuk memasukkan nilai melalui input. Kemudian, program memeriksa nilai tersebut dengan beberapa kondisi "jika-jika". Setelah semua kondisi diperiksa, program menampilkan hasil dalam bentuk indeks yang sesuai, dengan nilai input dalam format dua

angka desimal. Indeks "A" diberikan untuk nilai yang lebih besar atau sama dengan 80, indeks "B" diberikan untuk nilai yang lebih rendah dari 80, dan indeks "F" diberikan untuk nilai di bawah 40. Sebagai contoh, jika pengguna memasukkan nilai 75.5, program akan menghasilkan output berikut: "Nilai indeks untuk nilai 75.50 adalah B."

## D. UNGUIDED

### 1. Unguided1 – 2B – 2

#### Soal Studi Case

Suatu pita (string) berisi kumpulan nama-nama bunga yang dipisahkan oleh spasi dan ‘-’, contoh pita diilustrasikan seperti berikut ini. Pita: **mawar – melati – tulip – teratai – kamboja – anggrek**  
Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita.(Petunjuk: gunakan operasi penggabungan string dengan operator “+” ). Tampilkan isi pita setelah proses input selesai. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

N: <u>3</u>	N : <u>0</u>
Bunga 1: <u>Kertas</u>	Pita :
Bunga 2: <u>Mawar</u>	
Bunga 3: <u>Tulip</u>	
Pita: Kertas - Mawar - Tulip -	

Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan ‘SELESAI’. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bunga 1: <u>Kertas</u>	Bunga 1: <u>SELESAI</u>
Bunga 2: <u>Mawar</u>	Pita :
Bunga 3: <u>Tulip</u>	Bunga: 0
Bunga 4: <u>SELESAI</u>	
Pita: Kertas - Mawar - Tulip -	
Bunga: 3	

#### Sourcecode

```
package main
```

```
import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    var bunga []string
    scanner := bufio.NewScanner(os.Stdin)

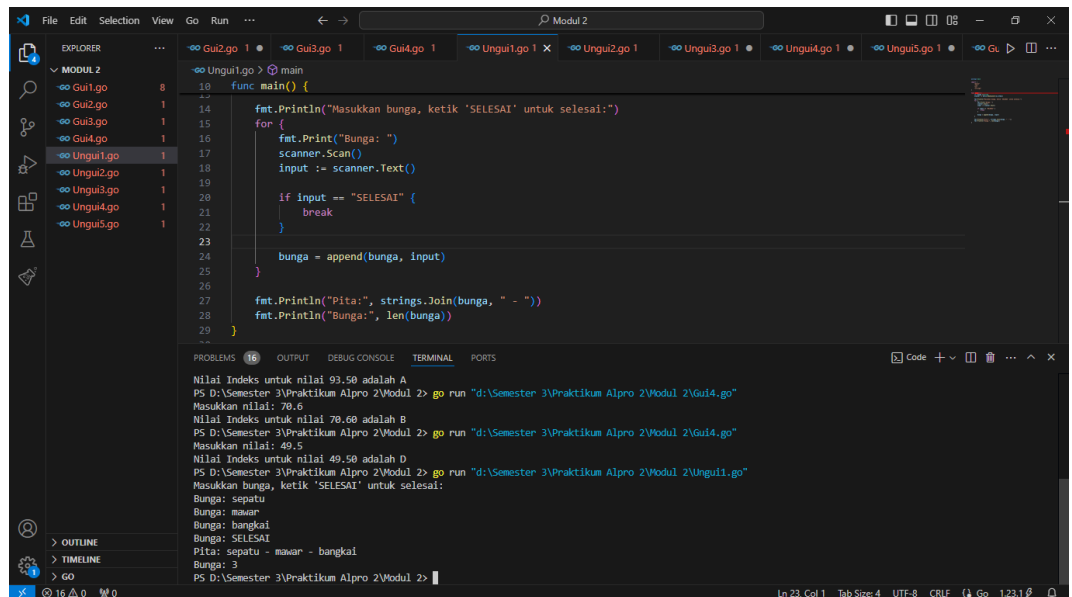
    fmt.Println("Masukkan bunga, ketik 'SELESAI'
untuk selesai:")
    for {
        fmt.Print("Bunga: ")
        scanner.Scan()
        input := scanner.Text()

        if input == "SELESAI" {
            break
        }

        bunga = append(bunga, input)
    }

    fmt.Println("Pita:", strings.Join(bunga, " -
"))
    fmt.Println("Bunga:", len(bunga))
}
```

### Screenshoot Output



```
10 func main() {
11     fmt.Println("Masukkan bunga, ketik 'SELESAI' untuk selesai:")
12     for {
13         fmt.Print("Bunga: ")
14         scanner.Scan()
15         input := scanner.Text()
16
17         if input == "SELESAI" {
18             break
19         }
20
21         bunga = append(bunga, input)
22     }
23
24     fmt.Println("Pita:", strings.Join(bunga, " - "))
25     fmt.Println("Bunga:", len(bunga))
26 }
```

PROBLEMS (16) OUTPUT DEBUG CONSOLE TERMINAL PORTS

Nilai Indeks untuk nilai 93.58 adalah A  
PS D:\Semester 3\Praktikum Alpro 2\Modul 2> go run "d:\Semester 3\Praktikum Alpro 2\Modul 2\Gui4.go"  
Masukkan nilai: 78.6  
Nilai Indeks untuk nilai 78.68 adalah B  
PS D:\Semester 3\Praktikum Alpro 2\Modul 2> go run "d:\Semester 3\Praktikum Alpro 2\Modul 2\Gui4.go"  
Masukkan nilai: 49.5  
Nilai Indeks untuk nilai 49.58 adalah D  
PS D:\Semester 3\Praktikum Alpro 2\Modul 2> go run "d:\Semester 3\Praktikum Alpro 2\Modul 2\Ungui1.go"  
Masukkan bunga, ketik 'SELESAI' untuk selesai:  
Bunga: sepatu  
Bunga: mawar  
Bunga: bangkai  
Bunga: SELESAI  
Pita: sepatu - mawar - bangkai  
Bunga: 3  
PS D:\Semester 3\Praktikum Alpro 2\Modul 2>

## Deskripsi Program

Deskripsi singkat dari program dalam bahasa Go adalah sebagai berikut: program meminta pengguna untuk memasukkan nama bunga berulang hingga mereka mengetik "SELESAI". Program juga mendeklarasikan slice "bunga" untuk menyimpan nama bunga yang dimasukkan dan variabel "scanner" untuk membaca input dari terminal menggunakan "bufio.NewScanner". Setelah pengguna mengetik "SELESAI", program akan berhenti menerima input. Kemudian, dengan menggunakan fungsi "strings.Join()", program menampilkan daftar bunga yang dimasukkan, yang dipisahkan dengan tanda strip (" - "). Selain itu, program juga menampilkan jumlah total bunga yang dimasukkan dengan menggunakan fungsi "len(bunga)."

## 2. Unguided 2 – 2B – 3

### Soal Studi Case

Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri-kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong sisi tidak lebih dari 9 kg. Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9 kg atau lebih. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

```
Masukan berat belanjaan di kedua kantong: 5.5 1.0
Masukan berat belanjaan di kedua kantong: 7.1 8.5
Masukan berat belanjaan di kedua kantong: 2 6
Masukan berat belanjaan di kedua kantong: 9 5.8
Proses selesai.
```

Pada modifikasi program tersebut, program akan menampilkan true jika selisih kedua isi kantong lebih dari atau sama dengan 9 kg. Program berhenti memproses apabila total berat isi kedua kantong melebihi 150 kg atau salah satu kantong beratnya negatif. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

```
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
```

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var kantong1, kantong2 float64
    for {
        fmt.Print("Masukan berat belanjaan di
kedua kantong: ")
        fmt.Scanln(&kantong1, &kantong2)

        total := kantong1 + kantong2

        if total > 150 {
            fmt.Println("Total berat tidak boleh
melebihi 150 kg.")
            fmt.Println("Program Selesai")
        }
    }
}
```

```

        break // Keluar dari loop dan program
selesai
    } else if kantong1 < 0 || kantong2 < 0 {
        fmt.Println("Berat tidak boleh
negatif.")
    } else if math.Abs(kantong1-kantong2) <
9 {
        fmt.Println("Sepeda motor akan oleng:
false")
    } else {
        fmt.Println("Sepeda motor akan oleng:
true")
    }
}
}

```

## Screenshot Output

The screenshot shows a Go IDE with the following components:

- EXPLORER:** A list of files including `MODUL 2`, `Gui1.go`, `Gui2.go`, `Gui3.go`, `Gui4.go`, `Ungui1.go`, `Ungui2.go`, `Ungui3.go`, `Ungui4.go`, and `Ungui5.go`.
- EDITOR:** The source code for `Ungui2.go` is displayed. It includes imports for `fmt` and `math`, a `main` function, and logic to calculate the total weight of two bags and determine if a motorcycle will wobble based on the weight difference.
- TERMINAL:** The output of the program is shown, including user input for bag weights and the resulting output messages.

**Source Code (Ungui2.go):**

```

4  "fmt"
5  "math"
6  )
7
8  func main() {
9      var kantong1, kantong2 float64
10     for {
11         fmt.Print("Masukan berat belanja di kedua kantong: ")
12         fmt.Scanln(&kantong1, &kantong2)
13
14         total := kantong1 + kantong2
15
16         if total > 150 {
17             fmt.Println("Total berat tidak boleh melebihi 150 kg.")
18             fmt.Println("Program Selesai")
19             break // Keluar dari loop dan program selesai
20         } else if kantong1 < 0 || kantong2 < 0 {
21             fmt.Println("Berat tidak boleh negatif.")
22         }
23     }
24 }

```

**Terminal Output:**

```

Bunga: mawar
Bunga: bangkai
Bunga: SELESAI
Pita: sepatu - mawar - bangkai
Bunga: 3
PS D:\Semester 3\Praktikum Alpro 2\Modul 2> go run "d:\Semester 3\Praktikum Alpro 2\Modul 2\Ungui2.go"
Masukan berat belanja di kedua kantong: 5 10
Sepeda motor akan oleng: false
Masukan berat belanja di kedua kantong: 55.6 70.2
Sepeda motor akan oleng: true
Masukan berat belanja di kedua kantong: 72.3 66.9
Sepeda motor akan oleng: false
Masukan berat belanja di kedua kantong: 59.5 98.7
Total berat tidak boleh melebihi 150 kg.
Program Selesai
PS D:\Semester 3\Praktikum Alpro 2\Modul 2>

```

## Deskripsi Program

Berikut adalah mengontrol berat belanja di dua kantong dan menentukan apakah sepeda motor akan oleng berdasarkan berat tersebut. Untuk menyimpan berat belanja di masing-masing kantong, program mendeklarasikan dua variabel `kantong1` dan `kantong2` bertipe `float64`. Kemudian, program masuk ke dalam loop yang meminta pengguna untuk memasukkan berat belanja di kedua kantong. Program menghitung berat total kedua kantong setelah menerima input.



Setelah itu, periksa situasi. Program memberikan peringatan jika berat total dalam kantong lebih dari 150 kg, dan mencetak "Program Selesai" untuk mengakhiri program. Jika berat dalam kantong negatif, program memberikan peringatan bahwa berat tidak boleh negatif. Selanjutnya, program menghitung berat kedua kantong. Jika kurang dari 9 kg, program menyatakan "Sepeda motor akan oleng: false". Jika kurang dari 9 kg, program menyatakan "Sepeda motor akan oleng: true".

#### Cara Program Berfungsi

Program dimulai dengan mengimpor paket yang diperlukan, seperti `fmt` dan `math`. Dalam loop `for`, program meminta pengguna untuk memasukkan berat dari dua kantong belanjaan. Program membaca input pengguna dan menyimpan nilainya ke dalam variabel kantong 1 dan kantong 2. Kemudian, program menjumlahkan kedua variabel tersebut untuk menghitung total berat belanjaan. Setelah itu, program memeriksa apakah berat total lebih dari 150 kg. Jika itu benar, pesan "Program Selesai" ditampilkan dan dikeluarkan. Program memberikan peringatan bahwa berat di salah satu kantong tidak boleh negatif. Program menyatakan bahwa sepeda motor akan stabil (tidak oleng) jika perbedaan berat kedua kantong kurang dari 9 kg. Jika perbedaan lebih besar dari 9 kg, program menyatakan bahwa sepeda motor akan oleng.

### 3. Unguided 3 – 2B – 4

#### Soal Studi Case

Diberikan sebuah persamaan sebagai berikut ini.

$$f(k) = \frac{(4k + 2)^2}{(4k + 1)(4k + 3)}$$

Buatlah sebuah program yang menerima input sebuah bilangan sebagai  $K$ , kemudian menghitung dan menampilkan nilai  $f(K)$  sesuai persamaan di atas. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Nilai $K$ = <u>100</u> Nilai $f(K)$ = 1.0000061880
---

#### Sourcecode

<pre>package main</pre>
-------------------------

```

import "fmt"

func main() {
    var k int = 100
    var f float64 = 1.0000061880

    fmt.Println("-----")
    fmt.Println("| Nilai K\t= ", k, " \t\t\t|")
    fmt.Println("| Nilai f(K)\t= ", f, " |")
    fmt.Println("-----")
}

```

## Screenshot Output

The screenshot shows a Go IDE with the following code in `Ungui3.go`:

```

1 import "fmt"
2
3 func main() {
4     var k int = 100
5     var f float64 = 1.0000061880
6
7     fmt.Println("-----")
8     fmt.Println("| Nilai K\t= ", k, " \t\t\t|")
9     fmt.Println("| Nilai f(K)\t= ", f, " |")
10    fmt.Println("-----")
11 }

```

The output window shows the result of running the program:

```

PS D:\Semester 3\Praktikum Alpro 2\Vbdul 2\Ungui3.go> go run "d:\Semester 3\Praktikum Alpro 2\Vbdul 2\Ungui3.go"
-----
| Nilai K      = 100          |
| Nilai f(K)   = 1.000006188 |
-----
PS D:\Semester 3\Praktikum Alpro 2\Vbdul 2>

```

## Deskripsi Program

Berikut ini adalah penjelasan singkat dari program sederhana dalam bahasa Go yang menampilkan nilai dari dua variabel: satu integer dan satu float. Program mendeklarasikan variabel "k" bertipe "int" dengan nilai 100 dan variabel "f" bertipe "float64" dengan nilai 1.0000061880. Setelah itu, program mencetak sebuah tabel sederhana yang menunjukkan nilai kedua variabel tersebut.

Program dapat membentuk tabel dengan menggunakan "fmt.Println()" dan menampilkan dua baris informasi. Nilai variabel "k" ditampilkan di baris

pertama dan nilai variabel "f" ditampilkan di baris kedua. Tabel output memiliki format dengan garis horizontal di bagian atas dan bawah, serta label dan nilai yang disusun secara rapi sehingga lebih mudah dibaca.

#### 4. Unguided 4 – 2C – 1

##### Soal Studi Case

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!

Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

1	Contoh #1 Berat parcel (gram): <u>8500</u> Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500
2	Contoh #2 Berat parcel (gram): <u>9250</u> Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750
3	Contoh #3 Berat parcel (gram): <u>11750</u> Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 110000

##### Sourcecode

```
package main
```

```

import (
    "fmt"
)

func main() {
    // Contoh data parcel
    parcels := []map[string]interface{}{
        {"berat": 8500, "detailBerat": []int{8,
500}},
        {"berat": 9250, "detailBerat": []int{9,
250}},
        {"berat": 11750, "detailBerat": []int{11,
750}},
    }

    // Hitung biaya untuk setiap parcel
    for i, parcel := range parcels {
        totalBiaya :=
hitungBiayaParcel(parcel["berat"].(int))
        parcels[i]["totalBiaya"] = totalBiaya
        fmt.Printf("Contoh #%d\n", i+1)
        fmt.Printf("Berat parcel (gram): %d\n",
parcel["berat"])
        fmt.Printf("Detail berat: %d kg + %d
gr\n", parcel["detailBerat"].([]int)[0],
parcel["detailBerat"].([]int)[1])
        fmt.Printf("Total biaya: Rp. %d\n",
totalBiaya)
        fmt.Println("-----")
    }
}

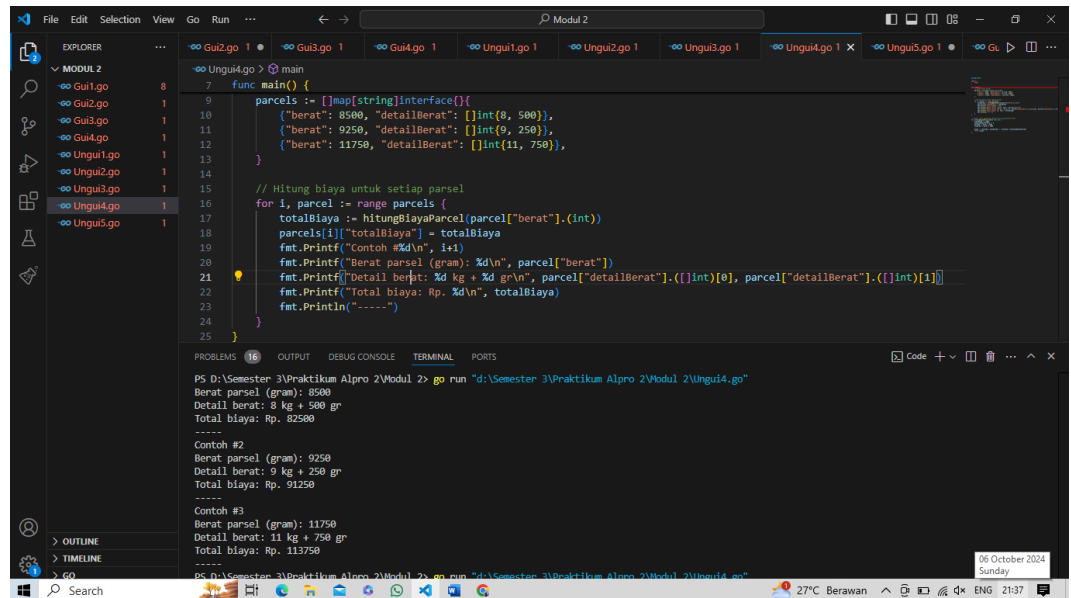
// Fungsi untuk menghitung biaya parcel
berdasarkan berat
func hitungBiayaParcel(berat int) int {
    biayaPerKg := 10000
    biayaTambahanPerGram := 5
    beratKg := berat / 1000
    sisaGram := berat % 1000

    biaya := (beratKg * biayaPerKg) + (sisaGram
* biayaTambahanPerGram)
    return biaya
}

```

```
}
```

## Screenshot Output



```
func main() {  
    parcels := []map[string]interface{}{  
        {"berat": 8500, "detailBerat": []int{8, 500}},  
        {"berat": 9250, "detailBerat": []int{9, 250}},  
        {"berat": 11750, "detailBerat": []int{11, 750}},  
    }  
  
    // Hitung biaya untuk setiap parcel  
    for i, parcel := range parcels {  
        totalBiaya := hitungBiayaParcel(parcel["berat"].(int))  
        parcels[i]["totalBiaya"] = totalBiaya  
        fmt.Printf("Contoh # %d\n", i+1)  
        fmt.Printf("Berat parcel (gram): %d\n", parcel["berat"])  
        fmt.Printf("Detail berat: %d kg + %d gr\n", parcel["detailBerat"].([]int)[0], parcel["detailBerat"].([]int)[1])  
        fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)  
        fmt.Println("-----")  
    }  
}
```

PS D:\Semester 3\Praktikum Alpro 2\Modul 2> go run "d:\Semester 3\Praktikum Alpro 2\Modul 2\Ungu4.go"

```
Berat parcel (gram): 8500  
Detail berat: 8 kg + 500 gr  
Total biaya: Rp. 82500  
-----  
Contoh #2  
Berat parcel (gram): 9250  
Detail berat: 9 kg + 250 gr  
Total biaya: Rp. 91250  
-----  
Contoh #3  
Berat parcel (gram): 11750  
Detail berat: 11 kg + 750 gr  
Total biaya: Rp. 113750  
-----
```

## Deskripsi Program

Sebuah program dalam bahasa Go yang menghitung dan menampilkan biaya pengiriman untuk beberapa parcel berdasarkan beratnya ditunjukkan di sini.

Program mendeklarasikan slice parcels, yang berisi peta (map) dengan berat dalam gram dan detail berat dalam gram. Kemudian, program menggunakan loop untuk melakukan iterasi melalui setiap parcel. Untuk setiap parcel, ada fungsi hitung yang disebut "BiayaParcel", yang bertugas menghitung biaya pengiriman total berdasarkan berat parcel tersebut.

### Cara Menghitung Biaya:

Fungsi Hitung BiayaParcel menggunakan rumus sederhana untuk menghitung biaya pengiriman. Berat parcel diukur dalam gram. Tarif Rp 10.000 per kilogram digunakan untuk menentukan biaya dasar. Di atas kilogram penuh, ada biaya tambahan Rp 5 per gram. Selanjutnya, fungsi ini mengembalikan nilai integer total biaya pengiriman.

Output Program: Program mencetak informasi detail tentang setiap parcel. Informasi ini mencakup nomor urut, berat dalam gram, rincian berat dalam gram, dan biaya pengiriman keseluruhan. Output disusun dengan format yang teratur, dengan pemisah untuk setiap parcel, sehingga mudah dibaca

dan dipahami. Secara keseluruhan, program ini membantu menghitung biaya pengiriman parcel berdasarkan berat dan menampilkannya secara sistematis.

## 5. Unguided 5 – 2C – 3

### Soal Studi Case

Sebuah bilangan bulat  $b$  memiliki faktor bilangan  $f > 0$  jika  $f$  habis membagi  $b$ . Contoh: 2 merupakan faktor dari bilangan 6 karena 6 habis dibagi 2.

Buatlah program yang menerima input sebuah bilangan bulat  $b$  dan  $b > 1$ . Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut!

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bilangan: <u>12</u> Faktor: 1 2 3 4 6 12	Bilangan: <u>7</u> Faktor: 1 7
---	-----------------------------------

Bilangan bulat  $b > 0$  merupakan bilangan prima  $p$  jika dan hanya jika memiliki persis dua faktor bilangan saja, yaitu 1 dan dirinya sendiri.

Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat  $b > 0$ . Program tersebut mencari dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah  $b$  merupakan bilangan prima.

Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read):

Bilangan: <u>12</u> Faktor: 1 2 3 4 6 12 Prima: false	Bilangan: <u>7</u> Faktor: 1 7 Prima: true
---	--

### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var b int
    fmt.Print("Bilangan: ")
    fmt.Scanln(&b)
```

```

    fmt.Print("Faktor: ")
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            fmt.Print(i, " ")
        }
    }
    fmt.Println()

    prima := true
    if b <= 1 {
        prima = false
    } else {
        for i := 2; i*i <= b; i++ {
            if b%i == 0 {
                prima = false
                break
            }
        }
    }

    fmt.Println("Prima:", prima)
}

```

## Screenshot Output

The screenshot shows an IDE with a Go file named `Ungu5.go` and its output in the terminal. The code defines a `main` function that prompts the user for a number, calculates its factors, and checks if it is a prime number. The terminal output shows the results for two test cases: a composite number (12) and a prime number (7).

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var b int
9     fmt.Print("Bilangan: ")
10    fmt.Scanln(&b)
11
12    fmt.Print("Faktor: ")
13    for i := 1; i <= b; i++ {
14        if b%i == 0 {
15            fmt.Print(i, " ")
16        }
17    }
18    fmt.Println()
19
20    prima := true
21    if b <= 1 {
22        prima = false
23    } else {
24        for i := 2; i*i <= b; i++ {
25            if b%i == 0 {
26                prima = false
27                break
28            }
29        }
30    }
31
32    fmt.Println("Prima:", prima)
33 }

```

```

Total biaya: Rp. 91250
-----
Contoh #3
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Total biaya: Rp. 113750
-----
PS D:\Semester 3\Praktikum Alpro 2\Vmodul 2> go run "d:\Semester 3\Praktikum Alpro 2\Vmodul 2\Ungu5.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Semester 3\Praktikum Alpro 2\Vmodul 2> go run "d:\Semester 3\Praktikum Alpro 2\Vmodul 2\Ungu5.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Semester 3\Praktikum Alpro 2\Vmodul 2>

```

## Deskripsi Program

Program dalam bahasa Go ini meminta pengguna untuk memasukkan sebuah bilangan bulat, kemudian menentukan faktor-faktor dari bilangan tersebut serta memeriksa apakah bilangan itu merupakan bilangan prima. Pertama, program meminta pengguna untuk memasukkan bilangan bulat yang disimpan dalam variabel `b`. Selanjutnya, program melakukan iterasi dari angka 1 hingga `b`, memeriksa apakah `b` habis dibagi oleh angka `i`. Jika ya, maka `i` dicetak sebagai faktor dari `b`. Untuk memeriksa apakah bilangan `b` adalah bilangan prima, program menginisialisasi variabel `prima` dengan nilai `true`. Jika `b` kurang dari atau sama dengan 1, variabel `prima` diubah menjadi `false`, karena bilangan negatif, nol, dan 1 bukan bilangan prima. Untuk bilangan lebih besar dari 1, program memeriksa apakah `b` dapat dibagi oleh angka dari 2 hingga akar kuadrat dari `b`. Jika ditemukan pembagi, `prima` diubah menjadi `false`, menandakan bahwa bilangan tersebut bukan bilangan prima. Setelah semua pemeriksaan selesai, program menampilkan hasilnya, yaitu apakah bilangan `b` adalah bilangan prima atau bukan, dengan mencetak nilai dari variabel `prima`. Dengan demikian, program ini memberikan informasi tentang faktor-faktor bilangan dan status primalitasnya.

## E. KESIMPULAN

Materi di atas mencakup berbagai program sederhana yang ditulis dalam bahasa Go, masing-masing dengan fungsi dan tujuan tertentu:

1. Program Input dan Output: Program ini menerima input pengguna dan menampilkannya ke layar.
2. Pengurutan Warna: Program yang memastikan bahwa urutan warna yang dimasukkan pengguna sesuai dengan urutan yang telah ditentukan dan memberikan hasil akhir berdasarkan validasi input.
3. Jumlah Angka: Program yang menghitung lima angka yang dimasukkan pengguna dan mencetak hasilnya.
4. Pengelolaan Daftar Bunga: Ini adalah aplikasi yang meminta pengguna memasukkan nama bunga hingga mereka mengetik "SELESAI". Kemudian, program menampilkan daftar bunga yang dimasukkan dan jumlah total bunga yang dimasukkan.
5. Biaya Pengiriman Parsel: Program yang menghitung dan menampilkan biaya pengiriman parsel berdasarkan beratnya dan mencetak informasi rinci.



6. Faktor dan Bilangan Prima: Aplikasi yang meminta pengguna memasukkan angka bulat dan menampilkan faktor dari angka tersebut.

## F. REFERENSI

- i. Modul 2 Praktikum Algoritma 2
- ii. BuildWithAngga. (n.d.). *Struktur Kontrol dalam Go: Penggunaan If, For, dan Switch* | BuildWithAngga. BuildWithAngga. <https://buildwithangga.com/tips/struktur-kontrol-dalam-go-penggunaan-if-for-dan-switch>
- iii. *Apa itu Golang? Arti, Fungsi, Contoh, FAQs 2024* | RevoU. (n.d.). <https://revou.co/kosakata/golang>