

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



Oleh:

CINTA HERTANTIARA BINTANG

NIM

IF 11 07

**S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

1. DASAR TEORI

Golang, atau Go, adalah bahasa pemrograman yang dikembangkan oleh Google, dirancang untuk efisiensi, kemudahan, dan keandalan. Bahasa ini semakin populer karena sintaksnya yang bersih dan cocok untuk berbagai proyek, mulai dari pengembangan web hingga aplikasi sistem.

Struktur Program Go

Setiap file program Go harus memiliki **package**, dengan minimal satu file yang memiliki package **main**. File dengan package **main** ini akan dieksekusi pertama kali saat program dijalankan. Selain itu, Go menggunakan **keyword import** untuk mengimpor package lain agar fungsionalitas dari package tersebut bisa digunakan dalam program. Salah satu package bawaan Go yang sering digunakan adalah **fmt**, yang menyediakan fungsi untuk operasi input-output (I/O) terkait teks, seperti **fmt.Println()** untuk mencetak teks ke layar dan **fmt.Printf()** untuk menampilkan output dengan format tertentu.

Variabel dan Tipe Data

Keyword **var** digunakan untuk mendeklarasikan variabel baru. Go mendukung beberapa tipe data seperti:

- Tipe Numerik Non-Desimal: **uint** (bilangan cacah/positif) dan **int** (bilangan bulat positif/negatif).
- Tipe Numerik Desimal: **float32** dan **float64**, yang berbeda dalam cakupan nilai desimalnya.
- Tipe Data Boolean: Hanya memiliki dua nilai, yaitu **true** dan **false**, sering digunakan dalam seleksi kondisi dan perulangan.
- Tipe Data String: Diapit oleh tanda kutip dua (**"**).

Go juga mendukung **konstanta simbolik** yang digunakan untuk memberi nama pada nilai konstan, seperti **PI** untuk mewakili nilai π .

Struktur Kontrol Perulangan

Go hanya memiliki satu kata kunci untuk perulangan, yaitu **for**. Perulangan ini dapat diimplementasikan dalam dua bentuk utama:

1. **While-Loop**: Memastikan kondisi bernilai benar saat masuk ke dalam loop, dan keluar ketika kondisi bernilai salah.

2. **Repeat-Until:** Perulangan akan terus berlanjut hingga kondisi terpenuhi.

Struktur Kontrol Percabangan

Beberapa bentuk struktur percabangan, seperti:

1. **If-Else:** Struktur dasar untuk pengambilan keputusan.
2. **Switch-Case:** Ada dua bentuk, yaitu dengan ekspresi pada perintah **switch** atau tanpa ekspresi, di mana setiap **case** bisa berupa ekspresi boolean.

Operator

Operator untuk operasi aritmatika dan logika, seperti:

- + untuk penjumlahan dan penggabungan string,
- - untuk pengurangan,
- * untuk perkalian,
- / untuk pembagian,
- && untuk operator "dan"
- || untuk operator "atau"
- == untuk perbandingan sama dengan.

Fungsi Utama

Fungsi **main()** adalah fungsi yang wajib ada di dalam file yang berisi package main, dan berfungsi sebagai titik awal eksekusi program. Fungsi ini juga menggunakan berbagai alat seperti variabel dan tipe data yang sudah dijelaskan di atas, untuk menjalankan logika pemrograman yang diinginkan.

Variabel dan Operator

Go menggunakan dua cara untuk mendeklarasikan variabel:

1. Dengan mendefinisikan tipe datanya: `var nama_variabel tipe_data.`
2. Dengan langsung memberikan nilai dan tipe data akan ditentukan secara otomatis: `nama_variabel := nilai.`

Selain itu, Go juga mendukung struktur data seperti array untuk penyimpanan data dengan panjang tetap, dan slice yang merupakan array tanpa inisialisasi index maupun elemen.

2. GUIDED

1. Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

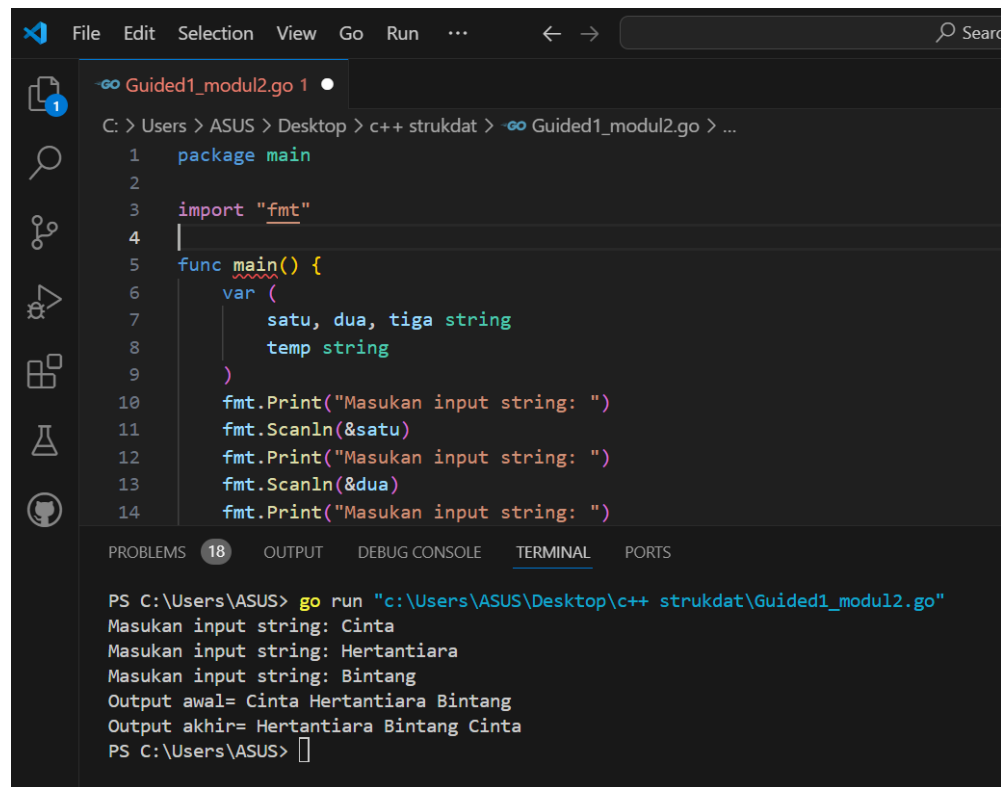
```
1 package main
2 import "fmt"
3
4 func main() {
5     var (
6         satu, dua, tiga string
7         temp string
8     )
9     fmt.Print("Masukan input string: ")
10    fmt.Scanln(&satu)
11    fmt.Print("Masukan input string: ")
12    fmt.Scanln(&dua)
13    fmt.Print("Masukan input string: ")
14    fmt.Scanln(&tiga)
15    fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
16    temp = satu
17    satu = dua
18    dua = tiga
19    tiga = temp
20    fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
21 }
```

SOURCE CODE

```
package main

import "fmt"
func main() {
    var (
        satu, dua, tiga string
        temp string
    )
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&satu)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&dua)
    fmt.Print("Masukan input string: ")
    fmt.Scanln(&tiga)
    fmt.Println("Output awal= " + satu + " " + dua + " " +
tiga)
    temp = satu
    satu = dua
    dua = tiga
    tiga = temp
    fmt.Println("Output akhir= " + satu + " " + dua + " "
+ tiga)
}
```

SCREENSHOOT PROGRAM



```
File Edit Selection View Go Run ... Search
Guided1_modul2.go 1
C: > Users > ASUS > Desktop > c++ strukdat > - Guided1_modul2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var (
7         satu, dua, tiga string
8         temp string
9     )
10    fmt.Print("Masukan input string: ")
11    fmt.Scanln(&satu)
12    fmt.Print("Masukan input string: ")
13    fmt.Scanln(&dua)
14    fmt.Print("Masukan input string: ")
15
16    // Pertukaran nilai
17    temp = satu
18    satu = dua
19    dua = temp
20
21    fmt.Print("Output awal= ")
22    fmt.Print(satu, dua, tiga)
23    fmt.Print("\nOutput akhir= ")
24    fmt.Print(satu, dua, tiga)
25}
PROBLEMS 18 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\Guided1_modul2.go"
Masukan input string: Cinta
Masukan input string: Hertantiara
Masukan input string: Bintang
Output awal= Cinta Hertantiara Bintang
Output akhir= Hertantiara Bintang Cinta
PS C:\Users\ASUS>
```

DESKRIPSI PROGRAM

Program ini bertujuan untuk menerima tiga input string dari pengguna, menampilkan output awal, melakukan pertukaran nilai antar variabel string, dan menampilkan output setelah pertukaran.

Deklarasi Variabel: Terdapat empat variabel satu, dua, tiga, dan temp, semuanya bertipe string. Variabel satu, dua, dan tiga digunakan untuk menampung input string dari pengguna, sementara temp digunakan sebagai variabel sementara untuk proses pertukaran nilai.

Input dari Pengguna: Program meminta pengguna untuk memasukkan tiga string, satu untuk masing-masing variabel satu, dua, dan tiga.

Menampilkan Output Awal: Program mencetak nilai awal dari ketiga variabel tersebut dalam urutan: satu, dua, dan tiga.

Pertukaran Nilai: Proses pertukaran dilakukan dengan urutan sebagai berikut:

- Nilai satu disimpan dalam temp (penyimpanan sementara).
- Nilai dua dipindahkan ke satu. Nilai tiga dipindahkan ke dua.
- Nilai temp (yang berisi nilai awal dari satu) dipindahkan ke tiga.
- Hasilnya, nilai variabel akan berotasi: satu ke dua, dua ke tiga, dan tiga ke satu.

Menampilkan Output Akhir:

- Setelah pertukaran, program mencetak hasil akhir dari ketiga variabel dalam urutan baru.

Output yang Diperoleh:

- Output awal akan menampilkan string yang dimasukkan oleh pengguna dalam urutan asli.
- Output akhir akan menampilkan string dalam urutan yang sudah diputar, dengan nilai awal satu pindah ke tiga, dua ke satu, dan tiga ke dua.

Program ini berhasil melakukan rotasi nilai antar tiga variabel string menggunakan variabel sementara temp.

2. Tahun kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (true) atau bukan (false).

(Contoh input/output, Teks bergaris bawah adalah input dari user):

1	Tahun: <u>2016</u> Kabisat: true
2	Tahun: <u>2000</u> Kabisat: true
3	Tahun: <u>2018</u> Kabisat: false

SOURCE CODE

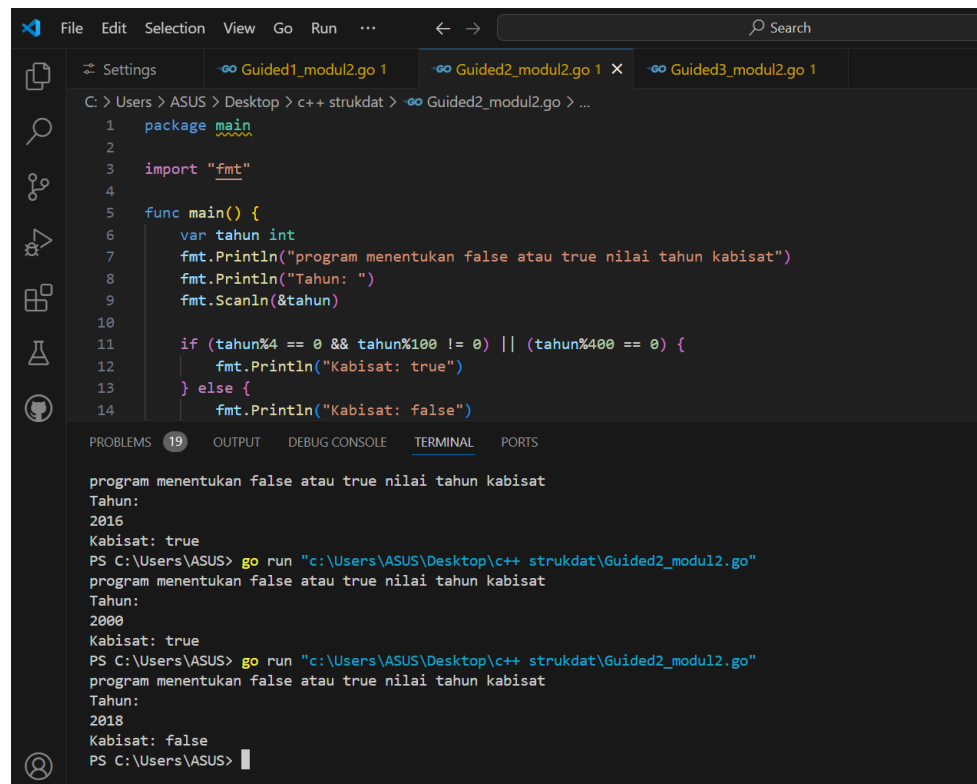
```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Println("program menentukan false atau true nilai tahun kabisat")
    fmt.Println("Tahun: ")
    fmt.Scanln(&tahun)

    if (tahun%4 == 0 && tahun%100 != 0) || (tahun%400 == 0)
    {
        fmt.Println("Kabisat: true")
    } else {
        fmt.Println("Kabisat: false")
    }
}
```

SCREENSHOOT PROGRAM



```
1 package main
2
3 import "fmt"
4
5 func main() {
6     var tahun int
7     fmt.Println("program menentukan false atau true nilai tahun kabisat")
8     fmt.Println("Tahun: ")
9     fmt.Scanln(&tahun)
10
11     if (tahun%4 == 0 && tahun%100 != 0) || (tahun%400 == 0) {
12         fmt.Println("Kabisat: true")
13     } else {
14         fmt.Println("Kabisat: false")
15     }
16 }
```

program menentukan false atau true nilai tahun kabisat
Tahun:
2016
Kabisat: true
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\Guided2_modul2.go"
program menentukan false atau true nilai tahun kabisat
Tahun:
2000
Kabisat: true
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\Guided2_modul2.go"
program menentukan false atau true nilai tahun kabisat
Tahun:
2018
Kabisat: false
PS C:\Users\ASUS>

DESKRIPSI PROGRAM

Program di atas adalah program yang menentukan apakah sebuah tahun merupakan tahun kabisat atau bukan. Program ini meminta pengguna untuk memasukkan nilai tahun, kemudian menggunakan logika untuk mengecek apakah tahun tersebut memenuhi syarat sebagai tahun kabisat. Syaratnya adalah: tahun harus habis dibagi 4, tetapi tidak boleh habis dibagi 100, kecuali jika juga habis dibagi 400. Jika tahun memenuhi syarat-syarat ini, program akan mencetak "Kabisat: true", jika tidak, akan mencetak "Kabisat: false".

3. Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam Fahrenheit

$$Celsius = (Fahrenheit - 32) \times \frac{5}{9} \quad Reamur = Celcius \times \frac{4}{5} \quad Kelvin = (Fahrenheit + 459.67) \times \frac{5}{9}$$

(Contoh input/output, Teks bergaris bawah adalah input dari user):

Temperatur Celsius: 50

Derajat Fahrenheit: 122

Lanjutkan program diatas, sehingga temperature dinyatakan juga dalam derajat Reamur dan Kelvin

(Contoh input/output, teks bergaris bawah adalah input dari user):

```
Temperatur Celsius: 50
Derajat Reamur: 40
Derajat Fahrenheit: 122
Derajat Kelvin: 323
```

SOURCE CODE

```
package main

import "fmt"

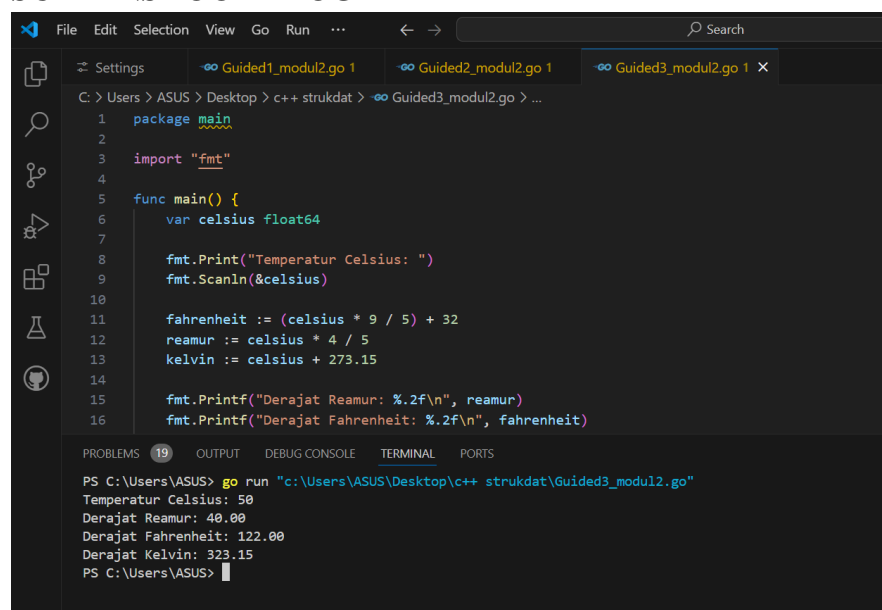
func main() {
    var celsius float64

    fmt.Print("Temperatur Celsius: ")
    fmt.Scanln(&celsius)

    fahrenheit := (celsius * 9 / 5) + 32
    reamur := celsius * 4 / 5
    kelvin := celsius + 273.15

    fmt.Printf("Derajat Reamur: %.2f\n", reamur)
    fmt.Printf("Derajat Fahrenheit: %.2f\n",
        fahrenheit)
    fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)
}
```

SCREENSHOOT PROGRAM



```
File Edit Selection View Go Run ... Search
Settings Guided1_modul2.go 1 Guided2_modul2.go 1 Guided3_modul2.go 1 X
C:\Users\ASUS> Desktop > c++ strukdat > Guided3_modul2.go > ...
1 package main
2
3 import "fmt"
4
5 func main() {
6     var celsius float64
7
8     fmt.Print("Temperatur Celsius: ")
9     fmt.Scanln(&celsius)
10
11     fahrenheit := (celsius * 9 / 5) + 32
12     reamur := celsius * 4 / 5
13     kelvin := celsius + 273.15
14
15     fmt.Printf("Derajat Reamur: %.2f\n", reamur)
16     fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)
17
18     PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\Guided3_modul2.go"
Temperatur Celsius: 50
Derajat Reamur: 40.00
Derajat Fahrenheit: 122.00
Derajat Kelvin: 323.15
PS C:\Users\ASUS>
```


DESKRIPSI PROGRAM

Program di atas adalah program yang mengonversi suhu dari derajat Celsius ke berbagai skala suhu lainnya, yaitu Reamur, Fahrenheit, dan Kelvin. Setelah meminta pengguna untuk memasukkan nilai suhu dalam Celsius, program menghitung nilai suhu dalam ketiga skala tersebut menggunakan rumus konversi yang sesuai. Hasil konversi untuk Reamur, Fahrenheit, dan Kelvin kemudian ditampilkan dengan format dua angka di belakang koma. Program ini memudahkan pengguna untuk mendapatkan nilai suhu dalam berbagai satuan berdasarkan input yang diberikan.

1. UNGUIDED

Soal Latihan 2B

1. 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 berturut-turut 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):

```
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

Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
```

SOURCE CODE

```
package main

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

func main() {
    correctOrder := []string{"merah", "kuning", "hijau", "ungu"}

    reader := bufio.NewReader(os.Stdin)

    berhasil := true

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

        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)
```

```

        warna := strings.Split(input, " ")

        for i := 0; i < 4; i++ {
            if warna[i] != correctOrder[i] {
                berhasil = false
                break
            }
        }

        if !berhasil {
            break
        }
    }

    if berhasil {
        fmt.Println("Berhasil: true")
    } else {
        fmt.Println("Berhasil: false")
    }
}

```

SCREENSHOOT PROGRAM

The screenshot shows a Go program in VS Code. The editor displays the source code for `laprak2b_no1.go`, which includes imports for `bufio`, `fmt`, `os`, and `strings`. The `main` function defines a `correctOrder` slice with the values `["merah", "kuning", "hijau", "ungu"]` and creates a `reader` from `os.Stdin`. The program then enters a loop to read input and check if it matches the `correctOrder` slice. The terminal output shows the program being run twice. In the first run, the input sequence is "merah kuning hijau ungu", which matches the `correctOrder` slice, resulting in "Berhasil: true". In the second run, the input sequence is "merah kuning hijau ungu", which does not match the `correctOrder` slice (as the last element is "merah" instead of "ungu"), resulting in "Berhasil: false".

```

C: > Users > ASUS > Desktop > c++ strukdat > .vscode > -go laprak2b_no1.go > main
1  package main
2
3  import (
4      "bufio"
5      "fmt"
6      "os"
7      "strings"
8  )
9
10 func main() {
11     correctOrder := []string{"merah", "kuning", "hijau", "ungu"}
12
13     reader := bufio.NewReader(os.Stdin)
14
15     berhasil := true

```

PROBLEMS 9 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\.vscode\laprak2b_no1.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 C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\.vscode\laprak2b_no1.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Berhasil: false
PS C:\Users\ASUS>

```

DESKRIPSI PROGRAM

Program di atas adalah program yang meminta pengguna untuk memasukkan urutan warna dan memeriksa apakah urutan tersebut sesuai dengan urutan yang benar, yaitu "merah", "kuning", "hijau", dan "ungu". Program ini menggunakan loop untuk memberikan hingga lima percobaan kepada pengguna untuk memasukkan urutan warna yang benar. Setelah pengguna memberikan input, program memisahkan input menjadi array string dan membandingkannya dengan urutan yang benar. Jika ada warna yang tidak sesuai, program akan menghentikan pengecekan dan menyimpan status keberhasilan sebagai false. Di akhir program, jika semua percobaan berhasil sesuai urutan yang benar, program mencetak "Berhasil: true"; jika tidak, mencetak "Berhasil: false".

2. 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	

SOURCE CODE

```
package main

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

func main() {
    bunga := []string{}
    reader := bufio.NewReader(os.Stdin)

    for perulangan := 1; ; perulangan++ {
        fmt.Printf("Bunga %d : ", perulangan)

        input, _ := reader.ReadString('\n')

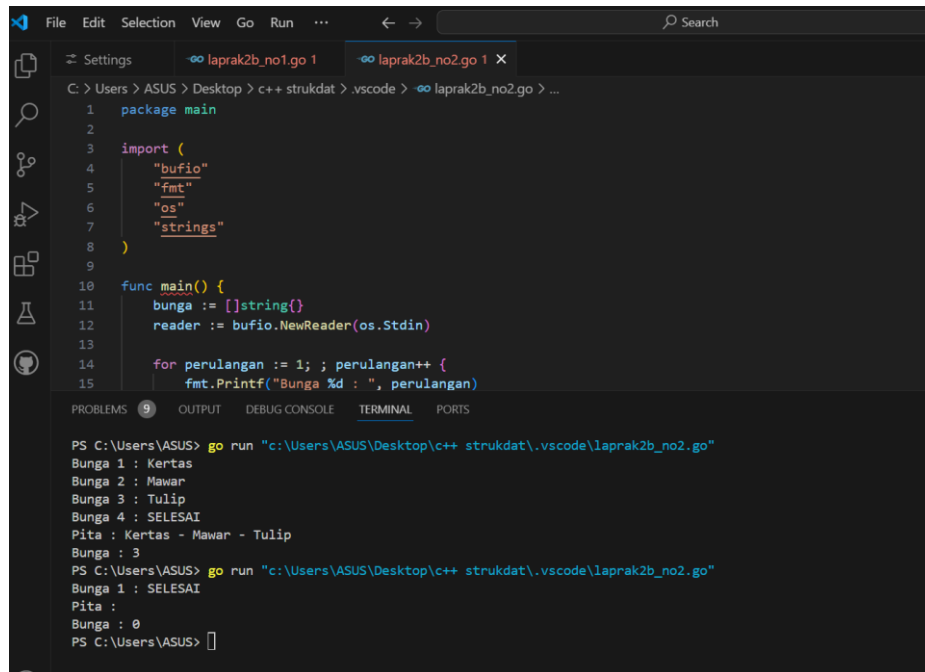
        input = strings.TrimSpace(input)

        if input == "SELESAI" {
            break
        }

        bunga = append(bunga, input)
    }

    result := strings.Join(bunga, " - ")
    fmt.Println("Pita : " + result)
    fmt.Printf("Bunga : %d\n", len(bunga))
}
```

SCREENSHOOT PROGRAM



The screenshot shows a Go program in a VS Code editor. The code is as follows:

```
1 package main
2
3 import (
4     "bufio"
5     "fmt"
6     "os"
7     "strings"
8 )
9
10 func main() {
11     bunga := []string{}
12     reader := bufio.NewReader(os.Stdin)
13
14     for perulangan := 1; ; perulangan++ {
15         fmt.Printf("Bunga %d : ", perulangan)
```

The terminal output shows the program being run twice. The first run shows the user inputting "Kertas", "Mawar", "Tulip", and "SELESAI", resulting in the output "Bunga : 3" and "Pita : Kertas - Mawar - Tulip". The second run shows the user inputting "SELESAI", resulting in the output "Bunga : 0" and "Pita : SELESAI".

DESKRIPSI PROGRAM

Program di atas adalah program yang memungkinkan pengguna untuk memasukkan nama bunga secara interaktif hingga pengguna mengetik "SELESAI". Program ini menggunakan loop tak terbatas yang meminta pengguna untuk memasukkan nama bunga setiap kali. Setiap input yang diberikan akan di-trim untuk menghapus spasi tambahan dan kemudian ditambahkan ke dalam slice `bunga`. Jika pengguna mengetik "SELESAI", loop akan berhenti. Setelah itu, program akan menggabungkan semua nama bunga yang dimasukkan menjadi satu string yang dipisahkan oleh tanda " - ". Di akhir, program mencetak string hasil tersebut serta jumlah total bunga yang telah dimasukkan oleh pengguna.

3. 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.

```
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.
```

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

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.
```

SOURCE CODE

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var kantongTerpal1, kantongTerpal2 float64
    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scanln(&kantongTerpal1, &kantongTerpal2)
        total := kantongTerpal1 + kantongTerpal2

        if total > 150 {
```

```

kg")
    fmt.Println("Total berat tidak boleh melebihi 150
kg")
    fmt.Println("Program Selesai")
    break

    } else if kantongTerpal1 < 0 || kantongTerpal2 < 0 {
        fmt.Println("Berat tidak boleh negatif")
    } else if math.Abs(kantongTerpal1-kantongTerpal2) < 9
{
        fmt.Println("Sepeda motor pak Andi akan oleng:
false")
    } else {
        fmt.Println("Sepeda motor pak Andi akan oleng:
true")
    }
}
}
}

```

SCREENSHOOT PROGRAM

The screenshot shows a Go program in VS Code. The code defines a `main` function that prompts the user to enter weights for two bags. It calculates the total weight and checks if it exceeds 150 kg. If it does, it prints a message and breaks the loop. Otherwise, it prints a message about the motorcycle and continues the loop.

```

1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func main() {
9     var kantongTerpal1, kantongTerpal2 float64
10    for {
11        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
12        fmt.Scanln(&kantongTerpal1, &kantongTerpal2)
13        total := kantongTerpal1 + kantongTerpal2
14
15        if total > 150 {
16            fmt.Println("Total berat tidak boleh melebihi 150 kg")
17            fmt.Println("Program Selesai")
18            break
19        }
20    }
21 }

```

The terminal output shows the program running and the user entering weights for two bags. The program prints the total weight and checks if it exceeds 150 kg. If it does, it prints a message and breaks the loop. Otherwise, it prints a message about the motorcycle and continues the loop.

```

PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\.vscode\laprak2b_no3.go"
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
Total berat tidak boleh melebihi 150 kg
Program Selesai
PS C:\Users\ASUS>

```

DESKRIPSI PROGRAM

Program di atas adalah program yang digunakan untuk memeriksa berat belanjaan yang dimasukkan dalam dua kantong dan menentukan apakah sepeda motor Pak Andi akan oleng atau tidak berdasarkan berat tersebut. Program berfungsi dengan meminta pengguna untuk memasukkan berat belanjaan di kedua kantong secara berulang. Setelah pengguna memasukkan berat, program menghitung total berat dari kedua kantong. Jika total berat melebihi 150 kg,

program akan mencetak pesan bahwa total berat tidak boleh melebihi 150 kg dan menghentikan program. Jika salah satu dari berat kantong adalah negatif, program akan memberi tahu pengguna bahwa berat tidak boleh negatif. Selain itu, jika selisih antara berat kedua kantong kurang dari 9 kg, program akan mencetak "Sepeda motor pak Andi akan oleng: false", menunjukkan bahwa sepeda motor tersebut tidak akan oleng. Jika selisihnya lebih dari atau sama dengan 9 kg, program mencetak "Sepeda motor pak Andi akan oleng: true", menandakan bahwa sepeda motor akan oleng. Program ini memberikan umpan balik langsung kepada pengguna berdasarkan input yang diberikan.

4. 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

$\sqrt{2}$ merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihampiri dengan rumus berikut:

$$\sqrt{2} = \prod_{k=0}^{\infty} \frac{(4k + 2)^2}{(4k + 1)(4k + 3)}$$

Modifikasi program sebelumnya yang menerima input integer K dan menghitung $\sqrt{2}$ untuk K tersebut. Hampiran $\sqrt{2}$ dituliskan dalam ketelitian 10 angka dibelakang koma.

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

1	Nilai K = <u>10</u> Nilai akar 2 = 1.4062058441
2	Nilai K = <u>100</u> Nilai akar 2 = 1.4133387072
3	Nilai K = <u>1000</u> Nilai akar 2 = 1.4141252651

SOURCE CODE

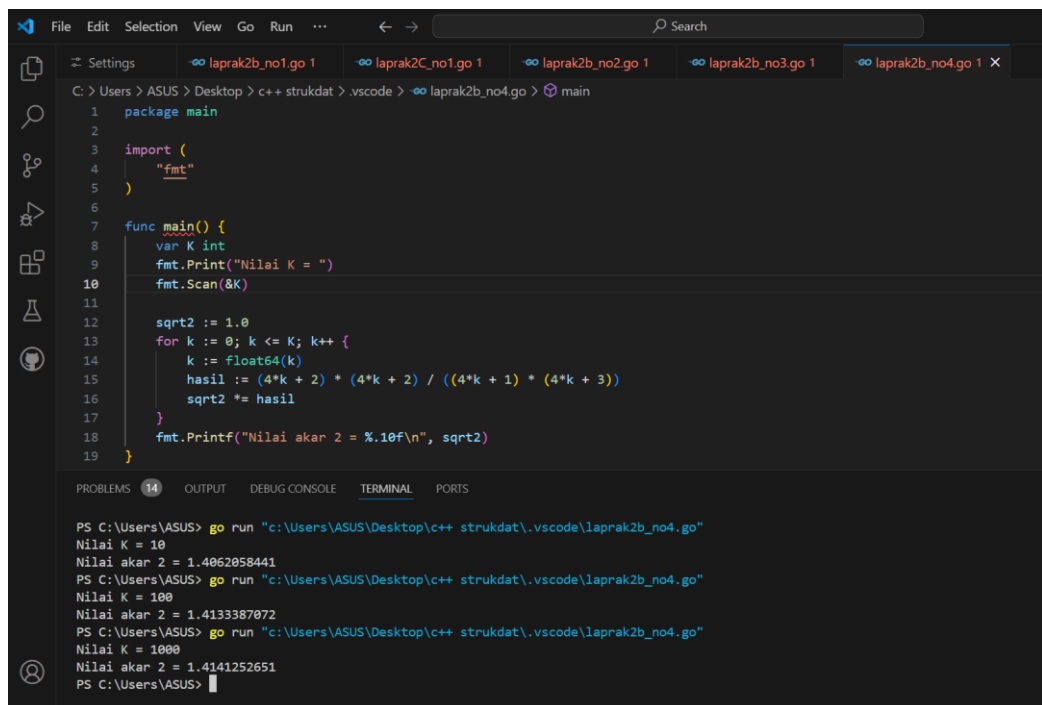
```
package main

import (
    "fmt"
)

func main() {
    var K int
    fmt.Print("Nilai K = ")
    fmt.Scan(&K)

    sqrt2 := 1.0
    for k := 0; k <= K; k++ {
        k := float64(k)
        hasil := (4*k + 2) * (4*k + 2) / ((4*k + 1) * (4*k + 3))
        sqrt2 *= hasil
    }
    fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2)
}
```

SCREENSHOOT PROGRAM



```
File Edit Selection View Go Run ... Search
laprak2b_no1.go 1 laprak2C_no1.go 1 laprak2b_no2.go 1 laprak2b_no3.go 1 laprak2b_no4.go 1 X
C:\Users\ASUS\Desktop>c++ strukdat\vscode>- laprak2b_no4.go> main
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var K int
9     fmt.Print("Nilai K = ")
10    fmt.Scan(&K)
11
12    sqrt2 := 1.0
13    for k := 0; k <= K; k++ {
14        k := float64(k)
15        hasil := (4*k + 2) * (4*k + 2) / ((4*k + 1) * (4*k + 3))
16        sqrt2 *= hasil
17    }
18    fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2)
19 }
PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\vscode\laprak2b_no4.go"
Nilai K = 10
Nilai akar 2 = 1.4062058441
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\vscode\laprak2b_no4.go"
Nilai K = 100
Nilai akar 2 = 1.4133387072
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\vscode\laprak2b_no4.go"
Nilai K = 1000
Nilai akar 2 = 1.4142135624
PS C:\Users\ASUS>
```

DESKRIPSI PROGRAM

Program di atas adalah program yang menghitung nilai akar dua ($\sqrt{2}$) menggunakan metode perhitungan berulang berdasarkan rumus matematis tertentu. Program dimulai dengan meminta pengguna untuk memasukkan nilai integer K, yang menentukan jumlah iterasi dalam perhitungan. Kemudian,

program menginisialisasi variabel sqrt2 dengan nilai 1.0. Selanjutnya, sebuah loop dijalankan dari 0 hingga

K, di mana pada setiap iterasi, program menghitung nilai hasil berdasarkan rumus $(4k+2)^2/((4k+1)(4k+3))$ dan mengalikan sqrt2 dengan hasil tersebut.

Setelah loop selesai, program mencetak nilai akhir dari sqrt2 dengan format sepuluh angka di belakang koma. Dengan menggunakan metode ini, program berusaha mendekati nilai akar dua dengan semakin banyak iterasi yang dilakukan, menghasilkan akurasi yang lebih tinggi.

Soal Latihan 2C

1. PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program BlayaPos 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 Nibebankan 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

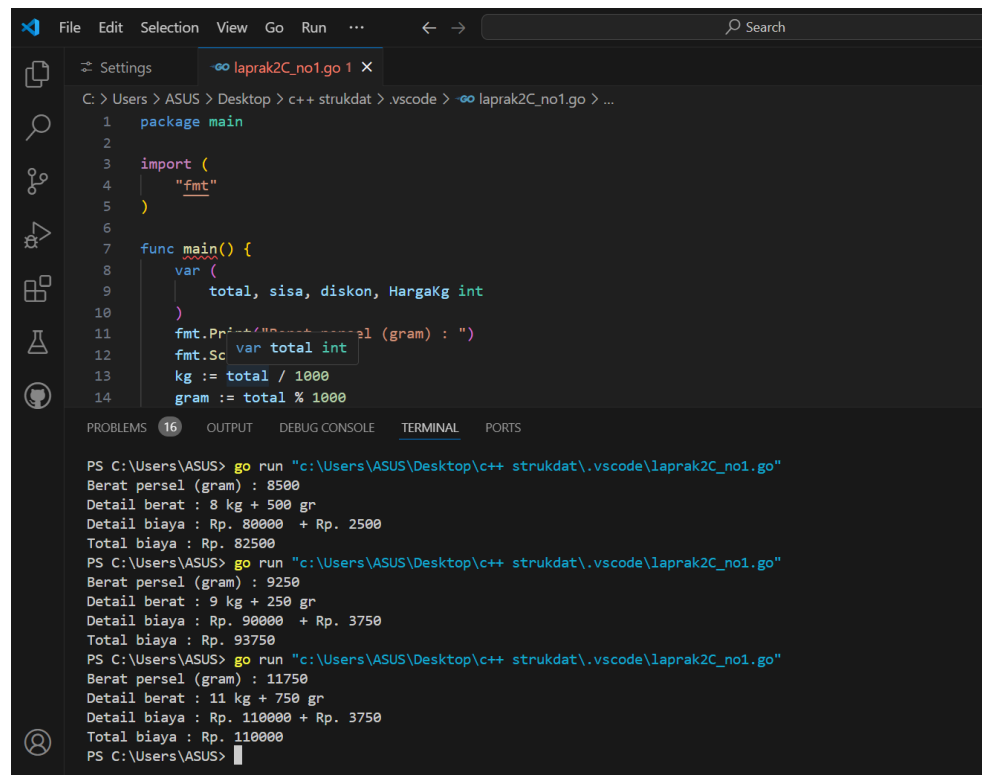
SOURCE CODE

```
package main

import (
    "fmt"
)

func main() {
    var (
        total, sisa, diskon, HargaKg int
    )
    fmt.Print("Berat persel (gram) : ")
    fmt.Scanln(&total)
    kg := total / 1000
    gram := total % 1000
    fmt.Printf("Detail berat : %d kg + %d gr", kg,
        gram)
    if total > 10000 {
        HargaKg = kg * 10000
        if gram >= 500 {
            sisa = gram * 5
        } else if gram < 500 {
            sisa = gram * 15
        }
        fmt.Println("\nDetail biaya : Rp.", HargaKg, "+ Rp.",
sisa)
        if sisa <= 1000 {
            diskon = 0
        }
        totalbiaya := HargaKg + diskon
        fmt.Println("Total biaya : Rp.", totalbiaya)
    } else {
        HargaKg = kg * 10000
        if gram >= 500 {
            sisa = gram * 5
        } else if gram < 500 {
            sisa = gram * 15
        }
        fmt.Println("\nDetail biaya : Rp.", HargaKg, " + Rp.",
sisa)
        totalbiaya := HargaKg + sisa
        fmt.Println("Total biaya : Rp.", totalbiaya)
    }
}
```

SCREENSHOOT PROGRAM



The screenshot shows a Go program in VS Code. The source code is as follows:

```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var (
9         total, sisa, diskon, HargaKg int
10    )
11    fmt.Printf("Berat persel (gram) : ")
12    fmt.Scanln(&total)
13    kg := total / 1000
14    gram := total % 1000
```

The terminal output shows three runs of the program:

```
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\.vscode\laprak2C_no1.go"
Berat persel (gram) : 8500
Detail berat : 8 kg + 500 gr
Detail biaya : Rp. 80000 + Rp. 2500
Total biaya : Rp. 82500

PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\.vscode\laprak2C_no1.go"
Berat persel (gram) : 9250
Detail berat : 9 kg + 250 gr
Detail biaya : Rp. 90000 + Rp. 3750
Total biaya : Rp. 93750

PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\.vscode\laprak2C_no1.go"
Berat persel (gram) : 11750
Detail berat : 11 kg + 750 gr
Detail biaya : Rp. 110000 + Rp. 3750
Total biaya : Rp. 110000

PS C:\Users\ASUS>
```

DESKRIPSI PROGRAM

Program di atas adalah program yang menghitung biaya berdasarkan berat persel (dalam gram) dan memberikan rincian biaya dengan mempertimbangkan diskon tertentu. Proses dimulai dengan meminta pengguna untuk memasukkan berat persel dalam gram. Program kemudian menghitung dan menampilkan berat dalam kilogram dan sisa gram. Jika total berat lebih dari 10.000 gram (10 kg), harga per kilogram ditetapkan menjadi Rp 10.000. Untuk sisa gram, biaya dihitung: jika sisa gram lebih dari atau sama dengan 500, dikenakan biaya Rp 5 per gram, dan jika kurang dari 500 gram, dikenakan biaya Rp 15 per gram. Program kemudian mencetak rincian biaya untuk berat dalam kilogram dan biaya sisa gram. Jika total biaya dari sisa gram kurang dari atau sama dengan Rp 1.000, diskon yang diterapkan adalah Rp 0. Total biaya akhir dihitung dengan menambahkan harga per kilogram dan sisa biaya (termasuk diskon jika ada) dan dicetak ke layar. Jika total berat tidak melebihi 10.000 gram, program menghitung dan mencetak biaya dengan cara yang sama tetapi tanpa mempertimbangkan diskon.

2. 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

Program berikut menerima input sebuah bilangan riil yang menyatakan NAM. Program menghitung NMK dan menampilkannya.

```
1 package main
2 import "fmt"
3 func main() {
4     var nam float64
5     var nmk string
6     fmt.Println("Nilai akhir mata kuliah: ")
7     fmt.Scanln(&nam)
8     if nam > 80 {
9         nmk = "A"
10    }
11    if nam > 72.5 {
12        nmk = "AB"
13    }
14    if nam > 65 {
15        nmk = "B"
16    }
17    if nam > 57.5 {
18        nmk = "BC"
19    }
20    if nam > 50 {
21        nmk = "C"
22    }
23    if nam > 40 {
24        nmk = "D"
25    } else if nam <= 40 {
26        nmk = "E"
27    }
28    fmt.Println("Nilai mata kuliah: ", nmk)
29 }
```

Jawablah pertanyaan-pertanyaan berikut:

- Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
- Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
- Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

SOURCE CODE

```
package main

import (
    "fmt"
)

func main() {
    var NAM float32
    var NMK string
```

```

    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&NAM)

    if NAM >= 80 {
        NMK = "A"
    } else if NAM >= 70 {
        NMK = "B"
    } else if NAM >= 65 {
        NMK = "C"
    } else if NAM >= 45 {
        NMK = "D"
    } else if NAM >= 40 {
        NMK = "E"
    } else {
        NMK = "F"
    }

    fmt.Printf("Nilai Indeks untuk nilai %.2f adalah %s\n",
NAM, NMK)
}

```

SCREENSHOOT PROGRAM

The screenshot shows a Go IDE with two tabs: `laprak2C_no1.go` and `laprak2c_no2.go`. The active tab is `laprak2c_no2.go`, which contains the following Go code:

```

1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var NAM float32
9     var NMK string
10
11     fmt.Print("Masukkan nilai: ")
12     fmt.Scan(&NAM)
13
14     if NAM >= 80 {
15         NMK = "A"
16     } else if NAM >= 70 {
17         NMK = "B"
18     } else if NAM >= 65 {
19         NMK = "C"

```

The terminal at the bottom shows the execution of the program with three test cases:

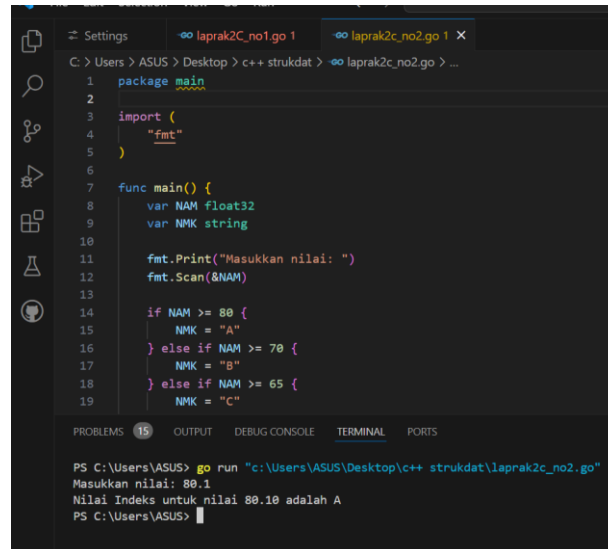
```

PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laprak2c_no2.go"
Masukkan nilai: 93.5
Nilai Indeks untuk nilai 93.50 adalah A
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laprak2c_no2.go"
Masukkan nilai: 70.6
Nilai Indeks untuk nilai 70.60 adalah B
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laprak2c_no2.go"
Masukkan nilai: 49.5
Nilai Indeks untuk nilai 49.50 adalah D
PS C:\Users\ASUS>

```

DESKRIPSI PROGRAM

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



```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var NAM float32
9     var NMK string
10
11     fmt.Print("Masukkan nilai: ")
12     fmt.Scan(&NAM)
13
14     if NAM >= 80 {
15         NMK = "A"
16     } else if NAM >= 70 {
17         NMK = "B"
18     } else if NAM >= 65 {
19         NMK = "C"
20     }
21 }
```

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laprak2c_no2.go"

Masukkan nilai: 80.1

Nilai Indeks untuk nilai 80.10 adalah A

PS C:\Users\ASUS>

Program menampilkan hasil eksekusi sesuai dengan spesifikasi yang diminta, yaitu nilai 80.1 lebih besar dari 80 sehingga memperoleh nilai indeks A.

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

Masalah Batas Nilai: Terdapat kesalahan dalam penentuan rentang nilai, seperti pada kondisi `nam > 80` yang membuat nilai 80.0 tidak termasuk dalam kategori "A". Solusinya adalah dengan menggunakan operator `>=` agar batas bawah juga tercakup dengan benar, misalnya nilai 80.0 harus termasuk dalam kategori "A".

Penggunaan Float32: Penggunaan tipe data float32 tidak diperlukan. Sebaiknya gunakan float64, karena lebih umum dan memberikan presisi yang lebih tinggi.

Alur Program yang Tepat: Nilai nam harus diproses dengan rentang yang benar, di mana batas bawah setiap kategori menggunakan operator `>=`. Sebagai contoh, nilai 80.0 termasuk dalam kategori "A". Alur program seharusnya dimulai dengan memeriksa nilai tertinggi (A), kemudian secara bertahap memeriksa nilai-nilai yang lebih rendah hingga mencapai "F".

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

SOURCE CODE

```
package main

import (
    "fmt"
)

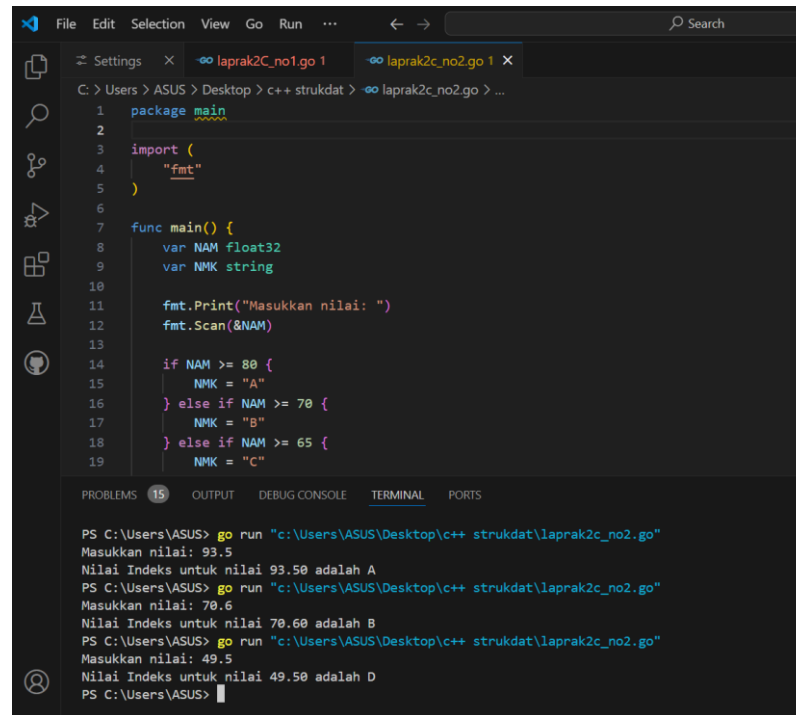
func main() {
    var NAM float32
    var NMK string

    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&NAM)

    if NAM >= 80 {
        NMK = "A"
    } else if NAM >= 70 {
        NMK = "B"
    } else if NAM >= 65 {
        NMK = "C"
    } else if NAM >= 45 {
        NMK = "D"
    } else if NAM >= 40 {
        NMK = "E"
    } else {
        NMK = "F"
    }

    fmt.Printf("Nilai Indeks untuk nilai %.2f\n", NAM, NMK)
}
```

SCREENSHOOT PROGRAM



```
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8     var NAM float32
9     var NMK string
10
11     fmt.Print("Masukkan nilai: ")
12     fmt.Scan(&NAM)
13
14     if NAM >= 80 {
15         NMK = "A"
16     } else if NAM >= 70 {
17         NMK = "B"
18     } else if NAM >= 65 {
19         NMK = "C"
20     }
21 }
```

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laprak2c_no2.go"
Masukkan nilai: 93.5
Nilai Indeks untuk nilai 93.50 adalah A
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laprak2c_no2.go"
Masukkan nilai: 70.6
Nilai Indeks untuk nilai 70.60 adalah B
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laprak2c_no2.go"
Masukkan nilai: 49.5
Nilai Indeks untuk nilai 49.50 adalah D
PS C:\Users\ASUS>
```

DESKRIPSI PROGRAM

Program di atas digunakan untuk mengonversi nilai numerik menjadi indeks huruf berdasarkan rentang nilai tertentu. Pertama, pengguna diminta memasukkan nilai melalui input. Program kemudian memeriksa nilai yang dimasukkan dengan serangkaian kondisi `if-else`. Jika nilai tersebut lebih besar atau sama dengan 80, indeks yang diberikan adalah "A". Jika nilainya berada antara 70 hingga kurang dari 80, diberikan indeks "B", dan seterusnya hingga nilai di bawah 40 yang mendapatkan indeks "F". Setelah semua kondisi diperiksa, program menampilkan hasil berupa indeks yang sesuai, dengan format dua angka desimal untuk nilai input. Sebagai contoh, jika pengguna memasukkan nilai 75.5, program akan menghasilkan output: "Nilai Indeks untuk nilai 75.50 adalah B".

3. 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: 12	Bilangan: 7
Faktor: 1 2 3 4 6 12	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: 12	Bilangan: 7
Faktor: 1 2 3 4 6 12	Faktor: 1 7
Prima: false	Prima: true

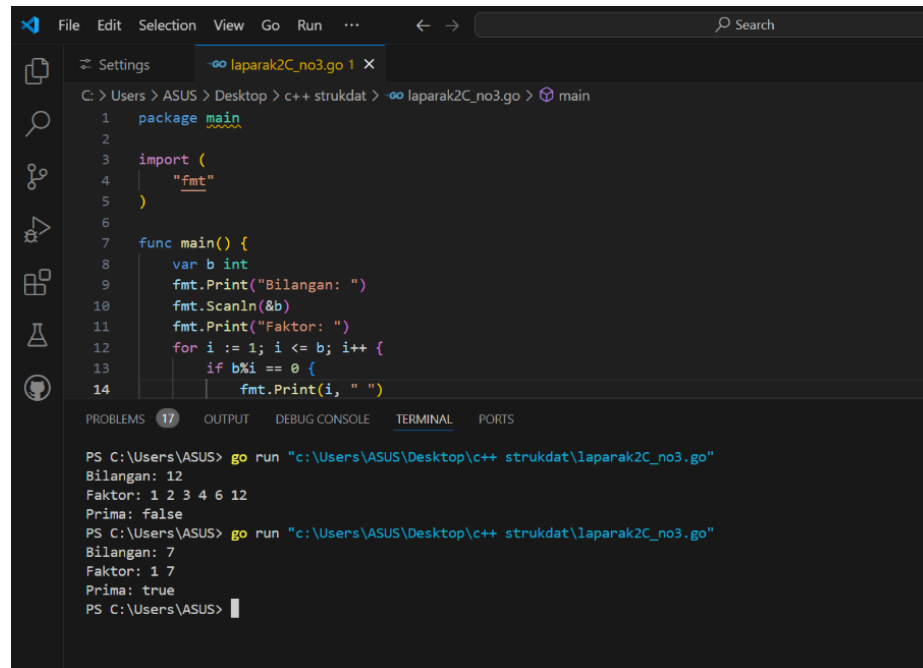
SOURCE CODE

```
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)
}
```

SCREENSHOOT PROGRAM



The screenshot shows a Go program in VS Code. The source code is as follows:

```
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    fmt.Print("Faktor: ")
12    for i := 1; i <= b; i++ {
13        if b%i == 0 {
14            fmt.Print(i, " ")
15        }
16    }
17    fmt.Print("\n")
18    if b == 1 {
19        fmt.Print("Prima: false\n")
20    } else {
21        fmt.Print("Prima: true\n")
22    }
23 }
```

The terminal output shows two runs of the program:

```
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laparak2C_no3.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\ASUS> go run "c:\Users\ASUS\Desktop\c++ strukdat\laparak2C_no3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\ASUS>
```

DESKRIPSI PROGRAM

Program ini adalah program yang meminta pengguna untuk memasukkan sebuah bilangan bulat dan kemudian menghitung serta menampilkan faktor-faktornya, serta menentukan apakah bilangan tersebut merupakan bilangan prima atau tidak.

Langkah pertama program adalah meminta input bilangan dari pengguna. Setelah mendapatkan input, program mencetak semua faktor dari bilangan tersebut dengan menggunakan loop yang memeriksa setiap bilangan dari 1 hingga bilangan tersebut. Jika bilangan tersebut dapat dibagi tanpa sisa oleh `i`, maka `i` dianggap sebagai faktor dan dicetak ke layar.

Setelah itu, program mengecek apakah bilangan tersebut adalah bilangan prima. Sebuah bilangan dikatakan prima jika hanya memiliki dua faktor, yaitu 1 dan bilangan itu sendiri. Program menginisialisasi variabel `prima` sebagai `true` dan melakukan pengecekan: jika bilangan kurang dari atau sama dengan 1, maka bilangan tidak prima. Untuk bilangan lebih besar dari 1, program memeriksa setiap bilangan dari 2 hingga akar kuadrat dari bilangan tersebut. Jika bilangan tersebut dapat dibagi oleh bilangan lain dalam rentang tersebut, maka `prima` diset menjadi `false`. Terakhir, program mencetak hasil apakah bilangan tersebut adalah bilangan prima atau bukan.