

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :
SYAHRUL ROMADHONI / 2311102261
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, yang juga dikenal sebagai Golang, adalah bahasa pemrograman yang dikembangkan oleh Google. Bahasa ini dirilis pertama kali pada tahun 2009 dan dirancang untuk meningkatkan efisiensi dalam pengembangan perangkat lunak, terutama dalam konteks pemrograman sistem dan aplikasi yang berskala besar. Berikut adalah beberapa poin kunci mengenai Go:

1. **Sintaks yang Sederhana:** Go memiliki sintaks yang bersih dan sederhana, sehingga memudahkan programmer untuk membaca dan menulis kode. Ini juga membantu mengurangi kemungkinan kesalahan.
2. **Kinerja Tinggi:** Go adalah bahasa yang terkompilasi, artinya kode sumbernya diubah menjadi kode mesin sebelum dijalankan. Ini memberikan kinerja yang tinggi dan efisiensi dalam penggunaan sumber daya.
3. **Kongruensi:** Salah satu fitur paling menonjol dari Go adalah dukungan untuk pemrograman konkuren melalui goroutine. Goroutine memungkinkan program untuk menjalankan beberapa fungsi secara bersamaan, membuatnya sangat efisien dalam menangani tugas-tugas yang dapat dilakukan secara paralel.
4. **Pengelolaan Memori:** Go dilengkapi dengan pengumpul sampah (garbage collector) yang membantu mengelola memori secara otomatis, mengurangi kemungkinan kebocoran memori dan memudahkan pengembang dalam manajemen sumber daya.
5. **Tooling yang Kuat:** Go memiliki ekosistem tooling yang kuat, termasuk sistem manajemen paket yang mudah digunakan, alat untuk pengujian, dan alat dokumentasi yang membantu pengembang dalam proses pengembangan.
6. **Statically Typed:** Go adalah bahasa yang terketik statis, yang berarti tipe variabel harus didefinisikan saat kompilasi. Ini membantu menemukan kesalahan lebih awal dalam proses pengembangan.
7. **Komunitas yang Aktif:** Go memiliki komunitas yang aktif dan beragam, yang berkontribusi pada pengembangan pustaka dan framework yang memperluas fungsionalitas bahasa ini.

Go banyak digunakan dalam pengembangan aplikasi web, layanan microservices, dan sistem terdistribusi, berkat kinerjanya yang tinggi dan kemudahan dalam penanganan konkuren. Dengan desain yang intuitif dan fitur-fitur modern, Go telah menjadi pilihan populer di kalangan pengembang perangkat lunak.

II. GUIDED

Soal Studi Case

1. Program membaca dan menampilkan nama

SOURCECODE

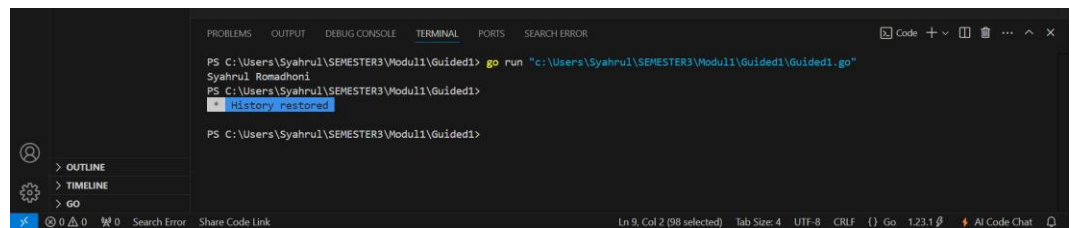
```
package main

import "fmt"

func main() {

    nama := "Syahrul Romadhoni"
    fmt.Print(nama)
}
```

SCREENSHOOT OUTPUT



DESKRIPSI PROGRAM

Program Go sederhana ini mendeklarasikan variabel nama yang berisi string "Syahrul Romadhoni". Program ini kemudian menggunakan fungsi `fmt.Print` untuk menampilkan isi dari variabel nama ke layar tanpa baris baru di akhir output. Saat dijalankan, program akan mencetak nama tersebut langsung di konsol.

2. Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. 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



DESKRIPSI PROGRAM

Program di atas menerima input berupa urutan warna dari 4 gelas reaksi untuk 5 kali percobaan dan memeriksa apakah setiap percobaan sesuai dengan urutan warna yang benar, yaitu merah, kuning, hijau, dan ungu. Input diberikan melalui terminal, dan program menggunakan `bufio.NewReader` untuk membaca setiap baris input, yang kemudian diproses dengan `strings.Split` untuk memisahkan warna berdasarkan spasi. Program memverifikasi setiap urutan warna dengan membandingkannya dengan urutan yang benar. Jika semua percobaan sesuai, program menampilkan "BERHASIL: true", dan jika ada satu percobaan yang salah, program langsung menghentikan pemeriksaan dan menampilkan "BERHASIL: false".

3. Penjumlahan dari angka yang diinputkan pengguna

SOURCECODE

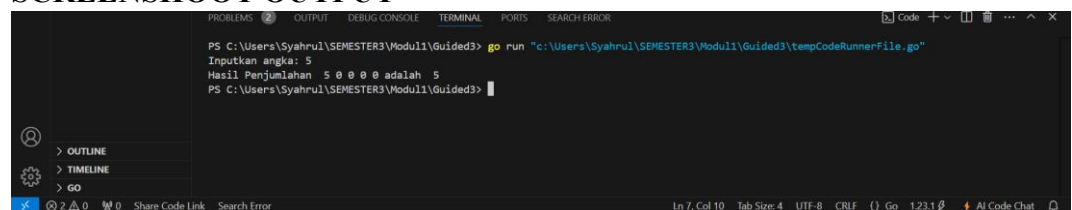
```
package main

import (
    "fmt"
)

func main(){
    var a,b,c,d,e int
    var hasil int
    fmt.Println("Inputkan angka: ")
    fmt.Scanln(&a, &b, &c, &d, &e)

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

SCREENSHOOT OUTPUT



DESKRIPSI PROGRAM

Program di atas adalah program sederhana dalam bahasa Go yang berfungsi untuk menjumlahkan lima buah bilangan yang diinput oleh pengguna. Program ini pertama-tama mendeklarasikan lima variabel integer (a, b, c, d, e) untuk menyimpan nilai-nilai input, serta satu variabel (hasil) untuk menyimpan hasil penjumlahan. Pengguna diminta memasukkan lima angka secara bersamaan, kemudian program menghitung total penjumlahan kelima angka tersebut dan menampilkannya ke layar dengan format yang jelas, menunjukkan bilangan yang dijumlahkan beserta hasil penjumlahannya.

4. Diberikan sebuah nilai akhir mata kuliah (NAM) [0...1000] dan standar penilaian data 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"

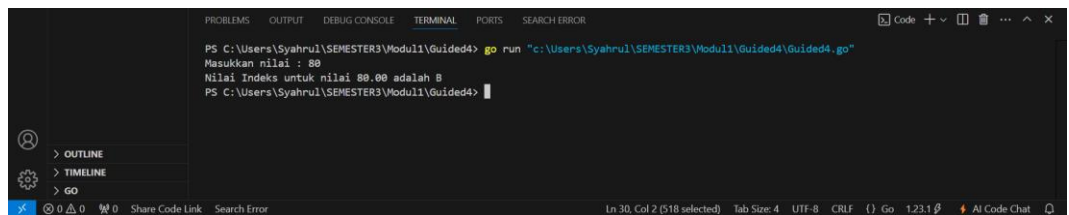
func main() {
    var nam float32
    var nmk string

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

    // Logika penentuan nilai huruf berdasarkan nilai numerik
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "B"
    } else if nam > 65 {
        nmk = "C"
    } else if nam > 50 {
        nmk = "D"
    }
```

```
} else if nam > 40 {  
    nmk = "E"  
}  
else {  
    nmk = "F"  
}  
  
// Menampilkan hasil  
fmt.Printf("Nilai Indeks untuk nilai %.2f adalah %s\n", nam, nmk)  
}
```

SCREENSHOOT OUTPUT

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and SEARCH ERROR. The terminal content shows the command 'go run "c:\Users\Syahrul\SEMESTER3\Modul1\Guided4\Guided4.go"' being executed. The output shows the user inputting '80', followed by the program output 'Nilai Indeks untuk nilai 80.00 adalah B'. The terminal status bar at the bottom indicates 'Ln 30, Col 2 (518 selected)', 'Tab Size: 4', 'UTF-8', 'CRLF', and 'Go 1.23.1'.

```
PS C:\Users\Syahrul\SEMESTER3\Modul1\Guided4> go run "c:\Users\Syahrul\SEMESTER3\Modul1\Guided4\Guided4.go"  
Masukkan nilai : 80  
Nilai Indeks untuk nilai 80.00 adalah B  
PS C:\Users\Syahrul\SEMESTER3\Modul1\Guided4>
```

DESKRIPSI PROGRAM

Program di atas bertujuan untuk mengonversi nilai numerik (dalam bentuk float) menjadi nilai huruf (indeks nilai) berdasarkan rentang yang telah ditentukan. Pengguna diminta untuk memasukkan nilai numerik, kemudian program akan menentukan nilai huruf (A, B, C, D, E, atau F) menggunakan beberapa kondisi if-else. Setiap rentang nilai memiliki kategori huruf tertentu, misalnya nilai lebih dari 80 akan mendapatkan huruf "A", lebih dari 72.5 akan mendapatkan "B", dan seterusnya hingga nilai kurang dari atau sama dengan 40 mendapatkan "F". Setelah itu, program akan menampilkan nilai huruf yang sesuai dengan nilai numerik yang dimasukkan.

III. UNGUIDED

Soal Studi Case

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

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

Sourcecode

```
package main

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

func main() {
    scanner := bufio.NewScanner(os.Stdin)

    var pita string
    var count int

    fmt.Println("Masukkan nama bunga (ketik 'SELESAI' untuk berhenti):")

    for {
        fmt.Printf("Masukkan nama bunga ke-%d: ", count+1)
        scanner.Scan()
        namaBunga := scanner.Text()

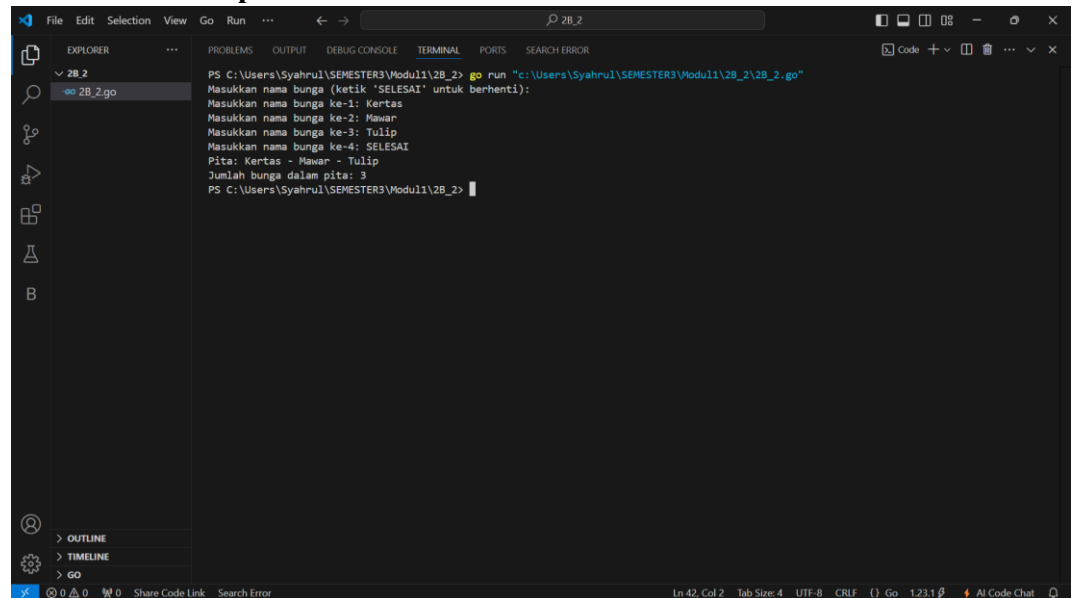
        if namaBunga == "SELESAI" {
            break
        }

        if pita == "" {
            pita = namaBunga
        } else {
            pita = pita + " - " + namaBunga
        }

        count++
    }

    fmt.Println("Pita:", pita)
    fmt.Printf("Jumlah bunga dalam pita: %d\n", count)
}
```

Screenshoot Output



```
PS C:\Users\Syahrul\SEMESTER3\Modul1\2B_2> go run "c:\Users\Syahrul\SEMESTER3\Modul1\2B_2\2B_2.go"
Masukkan nama bunga (ketik 'SELESAI' untuk berhenti):
Masukkan nama bunga ke-1: Kertas
Masukkan nama bunga ke-2: Mawar
Masukkan nama bunga ke-3: Tulip
Masukkan nama bunga ke-4: SELESAI
Pita: Kertas - Mawar - Tulip
Jumlah bunga dalam pita: 3
PS C:\Users\Syahrul\SEMESTER3\Modul1\2B_2> |
```

Deskripsi Program

Program ini menerima input berupa nama-nama bunga dari pengguna secara berulang menggunakan bufio.Scanner. Pengguna dapat memasukkan nama bunga satu per satu, dan program akan menyusun semua nama bunga yang dimasukkan ke dalam sebuah pita yang dipisahkan dengan tanda " - ". Proses input berlanjut hingga pengguna mengetikkan "SELESAI", yang akan menghentikan pengulangan. Setelah itu, program akan menampilkan pita yang berisi daftar nama-nama bunga yang dimasukkan dan juga menampilkan jumlah bunga yang berhasil dimasukkan ke dalam pita. Program ini menggunakan perulangan tak terbatas yang berhenti ketika kondisi tertentu terpenuhi (yaitu, saat pengguna mengetik "SELESAI").

2. Soal 2B_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.

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 beratKiri, beratKanan float64

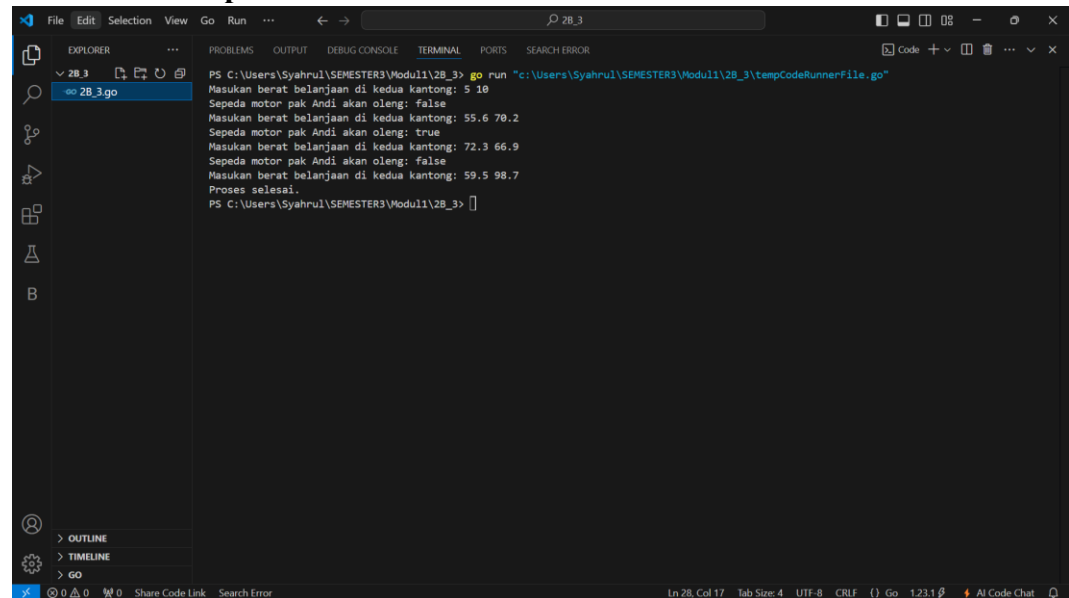
    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&beratKiri, &beratKanan)

        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        if beratKiri+beratKanan > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := math.Abs(beratKiri - beratKanan)
        if selisih >= 9 {
            fmt.Println("Sepeda motor pak Andi akan oleng: true")
        } else {
            fmt.Println("Sepeda motor pak Andi akan oleng: false")
        }
    }
}
```

Screenshoot Output



```
PS C:\Users\Syahru1\SEMESTER3\Modul1\28_3> go run "c:\Users\Syahru1\SEMESTER3\Modul1\28_3\tempCodeRunnerFile.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
Proses selesai.
PS C:\Users\Syahru1\SEMESTER3\Modul1\28_3>
```

Deskripsi Program

Program di atas bertujuan untuk mengevaluasi keseimbangan sepeda motor Pak Andi berdasarkan berat belanjaan yang dibawa di dua kantong. Pengguna diminta memasukkan berat belanjaan untuk kantong kiri dan kanan secara berulang. Program akan menghentikan proses jika salah satu berat belanjaan bernilai negatif atau jika total berat dari kedua kantong melebihi 150 kg. Jika perbedaan berat antara kedua kantong lebih dari atau sama dengan 9 kg, program akan menampilkan pesan bahwa "Sepeda motor pak Andi akan oleng: true", yang menandakan sepeda motor akan miring. Sebaliknya, jika perbedaan kurang dari 9 kg, program menampilkan "Sepeda motor pak Andi akan oleng: false".

3. Soal 3B_4

Diberikan sebuah persamaan sebagai berikut ini.

INFORMATICS 100

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

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 di belakang 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

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKiri, beratKanan float64

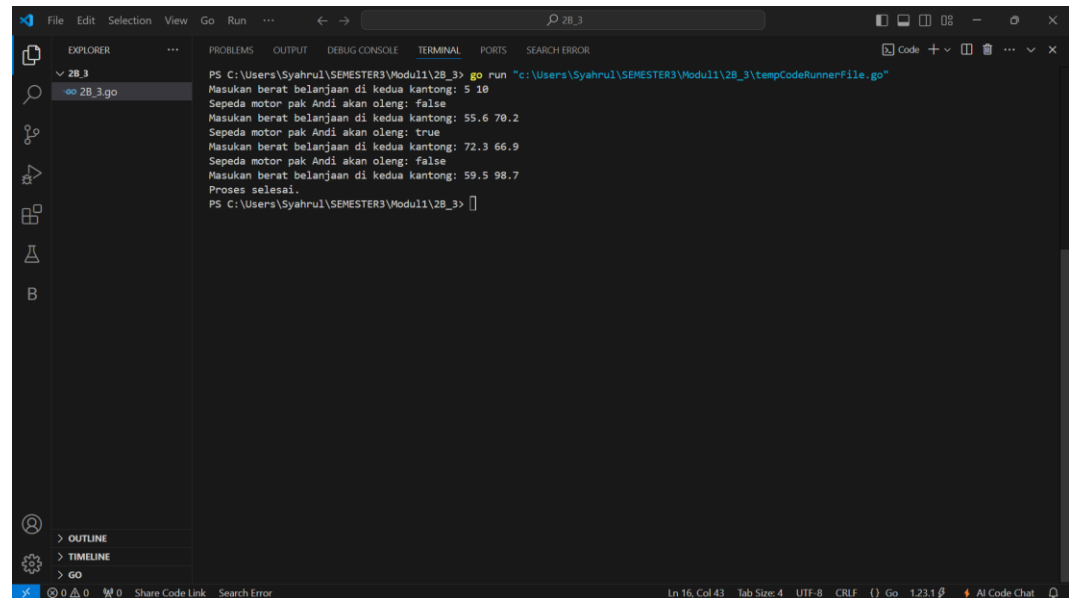
    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&beratKiri, &beratKanan)

        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        if beratKiri+beratKanan > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := math.Abs(beratKiri - beratKanan)
        if selisih >= 9 {
            fmt.Println("Sepeda motor pak Andi akan oleng: true")
        } else {
            fmt.Println("Sepeda motor pak Andi akan oleng: false")
        }
    }
}
```

Screenshoot



```
PS C:\Users\Syahru1\SEMESTER3\Modul1\28_3> go run "c:\Users\Syahru1\SEMESTER3\Modul1\28_3\tempCodeRunnerFile.go"
Masukan berat belanja di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanja di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanja di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanja di kedua kantong: 59.5 98.7
Proses selesai.
PS C:\Users\Syahru1\SEMESTER3\Modul1\28_3>
```

Deskripsi Program

Program ini bertujuan untuk menentukan apakah sepeda motor Pak Andi akan oleng berdasarkan perbedaan berat barang di dua kantong belanjaan yang ia bawa. Pengguna diminta untuk memasukkan berat belanjaan di kedua kantong secara berulang. Jika perbedaan berat antara kantong kiri dan kanan mencapai 9 kilogram atau lebih, program akan menampilkan pesan bahwa sepeda motor akan oleng. Jika selisihnya kurang dari 9 kilogram, program menampilkan bahwa sepeda motor tidak akan oleng. Program akan terus meminta input hingga berat kantong yang diberikan negatif atau total berat di kedua kantong melebihi 150 kilogram, yang menyebabkan program berhenti dengan pesan "Proses selesai."

1. Soal 2C_1

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 hitungBiayaKirim(beratParsel int) int {
    beratKg := beratParsel / 1000
    sisaGram := beratParsel % 1000

    biayaKg := beratKg * 10000
    var biayaSisaGram int

    if beratKg > 10 {
        biayaSisaGram = 0
    } else if sisaGram >= 500 {
        biayaSisaGram = sisaGram * 5
    } else {
        biayaSisaGram = sisaGram * 15
    }

    totalBiaya := biayaKg + biayaSisaGram
    return totalBiaya
}

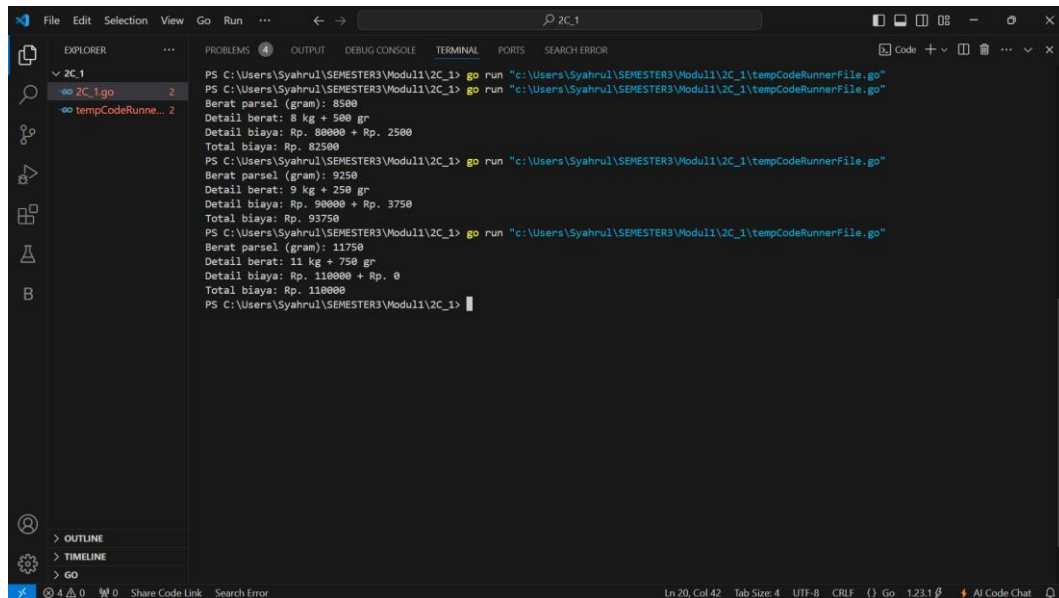
func main() {
    var beratParsel int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scanln(&beratParsel)

    totalBiaya := hitungBiayaKirim(beratParsel)
    beratKg := beratParsel / 1000
    sisaGram := beratParsel % 1000
    biayaKg := beratKg * 10000
    var biayaSisaGram int

    if beratKg > 10 {
        biayaSisaGram = 0
    } else if sisaGram >= 500 {
        biayaSisaGram = sisaGram * 5
    } else {
        biayaSisaGram = sisaGram * 15
    }

    fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, sisaGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayaSisaGram)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}
```

Screenshoot Output



```
PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_1> go run "c:\Users\Syahrul\SEMESTER3\Modul11\2C_1\tempCodeRunnerFile.go"
PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_1> go run "c:\Users\Syahrul\SEMESTER3\Modul11\2C_1\tempCodeRunnerFile.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 88000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_1> go run "c:\Users\Syahrul\SEMESTER3\Modul11\2C_1\tempCodeRunnerFile.go"
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_1> go run "c:\Users\Syahrul\SEMESTER3\Modul11\2C_1\tempCodeRunnerFile.go"
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_1>
```

Deskripsi Program

Program ini bertujuan untuk menghitung biaya pengiriman parcel berdasarkan beratnya dalam gram. Pertama, berat parcel yang diinput oleh pengguna dibagi menjadi kilogram (kg) dan sisa gram. Untuk setiap kilogram penuh, dikenakan biaya tetap sebesar Rp. 10.000 per kg. Jika berat sisa di bawah 500 gram, biaya tambahan adalah Rp. 15 per gram, sementara jika sisa lebih dari atau sama dengan 500 gram, biayanya adalah Rp. 5 per gram. Namun, jika berat parcel lebih dari 10 kg, tidak ada biaya tambahan untuk sisa gram. Program kemudian menampilkan rincian berat dan biaya pengiriman, serta total biaya yang harus dibayarkan.

2. Soal 2C_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: <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 bilangan int
    fmt.Print("Bilangan: ")
    fmt.Scanln(&bilangan)

    if bilangan <= 1 {
        fmt.Println("Bilangan harus lebih besar dari 1")
        return
    }

    fmt.Print("Faktor: ")
```

```

var faktorCount int
for i := 1; i*i <= bilangan; i++ {
    if bilangan%i == 0 {
        fmt.Print(i, " ")
        faktorCount++
        if i != bilangan/i {
            fmt.Print(bilangan/i, " ")
            faktorCount++
        }
    }
}

fmt.Println()
if faktorCount == 2 {
    fmt.Println("Prima: true")
} else {
    fmt.Println("Prima: false")
}
}

```

Screenshoot Output

```

PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_3> go run "c:\Users\Syahrul\SEMESTER3\Modul11\2C_3\2C_3.go"
Bilangan: 12
Faktor: 1 2 2 6 3 4
Prima: false
PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_3> go run "c:\Users\Syahrul\SEMESTER3\Modul11\2C_3\2C_3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\Syahrul\SEMESTER3\Modul11\2C_3>

```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan sebuah bilangan bulat lebih besar dari 1, lalu menampilkan semua faktor dari bilangan tersebut. Program juga memeriksa apakah bilangan tersebut adalah bilangan prima. Faktor bilangan dicari dengan memeriksa setiap bilangan dari 1 hingga akar kuadrat dari bilangan yang diberikan. Jika suatu bilangan merupakan faktor, maka hasil bagi juga ditampilkan sebagai faktor. Setelah itu, program menghitung jumlah faktor. Jika bilangan hanya memiliki 2 faktor (1 dan bilangan itu sendiri), maka program akan menyatakan bahwa bilangan tersebut adalah bilangan prima dengan mencetak "Prima: true", jika tidak, akan mencetak "Prima: false".