

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

Avrizal Setyo Aji Nugroho

2311102145

IF-11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

I. DASAR TEORI

Golang, juga dikenal sebagai "Go," adalah bahasa pemrograman yang dibuat oleh Google pada tahun 2007 dan dirilis sebagai open-source pada tahun 2009. Go dikembangkan oleh Robert Griesemer, Rob Pike, dan Ken Thompson untuk menangani masalah pemrograman yang membutuhkan kompilasi cepat, eksekusi efektif, dan manajemen sumber daya yang baik. Go adalah bahasa statically typed yang mendukung deteksi kesalahan pada waktu kompilasi. Salah satu fitur utamanya adalah dukungan concurrency dengan goroutines dan channels, yang memungkinkan proses parallel secara efisien. Selain itu, Go memiliki "garbage collection" untuk manajemen memori otomatis dan pustaka standar yang kaya untuk berbagai kebutuhan umum. Kemampuan "cross-platform", kompilasi cepat, dan deployment yang mudah melalui "binaries mandiri" yang fleksibel tanpa deklarasi eksplisit adalah keunggulan lainnya. Contoh sederhana dari aplikasi Go adalah program "Hello, World!"

```
package main

import "fmt"

func main() {
    fmt.Println("Hello, World!")
}
```

II. GUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

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 i dengan dengan informasi informasi yang yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya.

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

```
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\go2.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: kuning merah ungu hijau
Percobaan 5: merah kuning hijau ungu
BERHASIL : false
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\go2.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:\ALpro Sem3> 
```

Deskripsi Program

Program diatas sebuah program yang meminta input dari user sebanyak 5x untuk serangkaian warna dan harus dengan urutan yang benar. Tujuannya adalah untuk mengetahui apakah pengguna telah memasukkan urutan yang benar(merah, kuning, hijau, dan ungu). Jika urutan program benar, maka output akan menampilkan "True". Namun, jika salah satu percobaan tidak sesuai, maka output yang dihasilkan adalah "False"

2. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

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

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

Sourcecode

```
package main

import "fmt"

func main() {
    var nam float32
    var nmk string

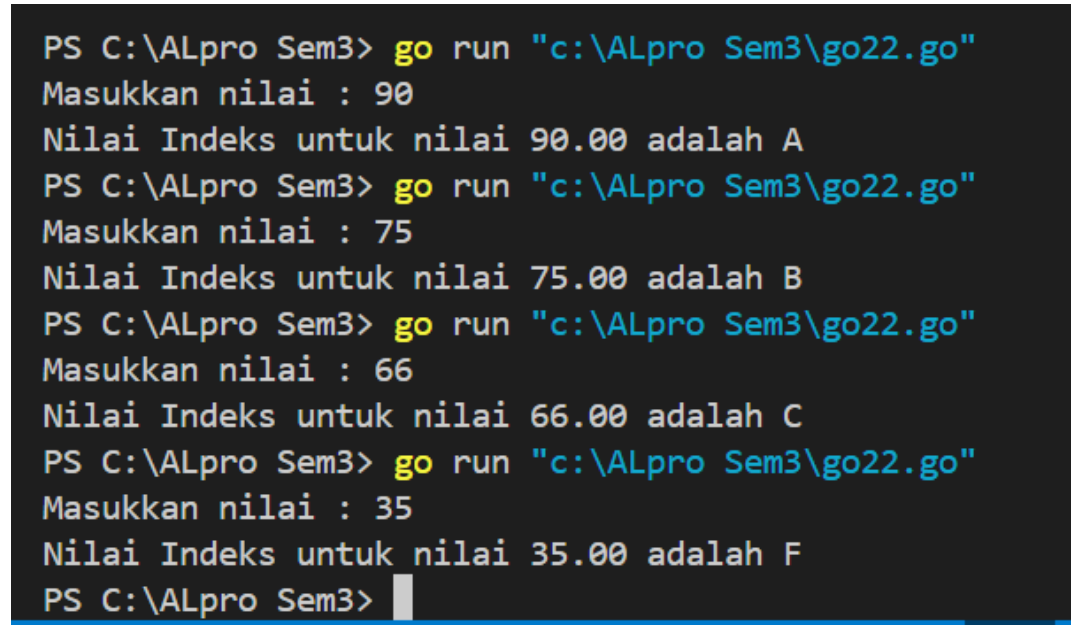
    // Meminta input nilai
    fmt.Print("Masukkan nilai : ")
    fmt.Scan(&nam)

    // 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\n%s\n", nam, nmk)
}
```

Screenshoot Output



```
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\go22.go"
Masukkan nilai : 90
Nilai Indeks untuk nilai 90.00 adalah A
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\go22.go"
Masukkan nilai : 75
Nilai Indeks untuk nilai 75.00 adalah B
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\go22.go"
Masukkan nilai : 66
Nilai Indeks untuk nilai 66.00 adalah C
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\go22.go"
Masukkan nilai : 35
Nilai Indeks untuk nilai 35.00 adalah F
PS C:\ALpro Sem3>
```

Deskripsi Program

Program di atas program untuk mengubah nilai numerik menjadi nilai huruf, atau indeks, berdasarkan kisaran tertentu. Program menjalankan sejumlah kondisi if-else setelah nilai diinputkan oleh user untuk

menentukan nilai indeks yang tepat. Jika nilai lebih dari 80, maka indeksnya adalah "A", jika lebih dari 72.5 adalah "B", dan seterusnya hingga indeks "F" untuk nilai yang kurang dari atau sama dengan 40. Setelah nilai huruf ditentukan, hasilnya ditampilkan dalam format yang rapi, dengan nilai numerik yang dimasukkan dalam dua digit desimal.

III. UNGUIDED

1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

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-teratal-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.

Modifikasi program sebelumnya, proses Input akan berhenti apabila user mengetikkan 'SELESAI'. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita

Sourcecode

```
package main

import (
    "bufio"
    "fmt"
```



```
"os"
"strings"
)

func main() {
    // Membuat scanner input
    scanner_45 := bufio.NewScanner(os.Stdin)

    // Input Nama Bunga
    var flowers_45 []string
    for {
        fmt.Printf("Bunga: ")
        scanner_45.Scan()
        flower := scanner_45.Text()
        if flower == "SELESAI" {
            break
        }
        flowers_45 = append(flowers_45, flower)
    }

    // Menggabungkan nama bunga menjadi satu
    pita_45 := strings.Join(flowers_45, " - ")

    // Menampilkan
    fmt.Println("Pita:", pita_45)
    fmt.Println("Banyak bunga: ", len(flowers_45))
}
```

Screenshot Output

```
ided1.go"
Bunga: mawar
Bunga: melati
Bunga: tulip
Bunga: teratai
Bunga: kamboja
Bunga: anggrek
Bunga: SELESAI
Pita: mawar - melati - tulip - teratai - kamboja - anggrek
Banyak bunga: 6
PS C:\ALpro Sem3> █
```

Deskripsi Program

Program diatas digunakan untuk menginput nama bunga secara dinamis, kemudian ditampilkan sebagai rangkaian (pita) yang dipisahkan oleh tanda strip ("-"). Algoritma bekerja dengan menggunakan loop yang terus meminta input dari pengguna untuk nama bunga. Pengguna dapat menginput nama bunga satu per satu, dan jika mengetik "SELESAI", program akan berhenti menerima input. Setiap nama bunga akan disimpan dalam slice. Setelah loop berhenti, algoritma menggabungkan semua nama bunga menjadi satu string dengan tanda pemisah strip, dan kemudian menampilkan hasil, termasuk jumlah total bunga yang dimasukkan.

2. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

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: 26
Masukan berat belanjaan di kedua kantong: 95.8
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 (
    "bufio"
    "fmt"
    "math"
    "os"
    "strconv"
    "strings"
)

func main() {
    scanner_45 := bufio.NewScanner(os.Stdin)

    var bag1, bag2 float64

    for {
        // perintah input
        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
        scanner_45.Scan()
        input := scanner_45.Text()

        //Jika input tidak valid
        weights := strings.Split(input, " ")
        if len(weights) != 2 {
            fmt.Println("Input tidak valid. Harap masukkan dua berat yang dipisahkan dengan spasi.")
            continue
        }

        // rubah input ke float
        w1, err1 := strconv.ParseFloat(weights[0], 64)
```

```

        w2, err2 := strconv.ParseFloat(weights[1], 64)
        if err1 != nil || err2 != nil {
            fmt.Println("Input tidak valid. Harap
masukkan angka.")
            continue
        }

        // menambahkan input berat pada bag
        bag1 += w1
        bag2 += w2

        // Untuk mengecek apakah salah satu tas melebihi
9 kg
        if bag1 < 0 || bag2 < 0 || bag1 + bag2 > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        // Untuk mengecek apakah selisih tas
        diff := math.Abs(bag1 - bag2)
        if diff >= 9 {
            fmt.Println("sepeda motor pak andi akan
oleng : true")
        }else{
            fmt.Println("sepeda motor pak andi akan
oleng : false")
        }

    }
}

```

Screenshoot Output

```
Masukkan berat belanjaan di kedua kantong: 5 10
sepeda motor pak andi akan oleng : false
Masukkan berat belanjaan di kedua kantong: 55.6 70.2
sepeda motor pak andi akan oleng : true
Masukkan berat belanjaan di kedua kantong: 72.3 66.9
Proses selesai.
PS C:\ALpro Sem3>
```

Deskripsi Program

Program diatas digunakan untuk memastikan bahwa berat belanjaan yang dimasukkan ke dalam dua kantong yang Pak Andi bawa dengan sepeda motor seimbang. Algoritma program ini bekerja dengan meminta input berat belanjaan untuk dua kantong secara berulang-ulang. Setelah berat masing-masing kantong ditambahkan, algoritma menghitung selisih antara berat kedua kantong. Program akan memberikan peringatan bahwa sepeda motor akan oleng jika perbedaan berat antara kedua kantong lebih dari atau sama dengan 9 kg. Jika total berat kedua kantong melebihi 150 kg atau jika input tidak valid, program juga akan berhenti.

3. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

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

$\sqrt{2}$ merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihipir 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() {
    // deklarasi variable K
    var K_45 int
```

```

// Minta pengguna untuk memasukkan nilai K
fmt.Print("Masukkan nilai K: ")
fmt.Scan(&K_45)

// Variable untuk menyimpan product
product := 1.0

// Loop dari k = 0 ke K untuk dikali product
for k := 0; k <= K_45; k++ {
    // Hitung persyaratan untuk produk
    numerator := math.Pow(4*float64(k)+2, 2) //
(4k + 2)^2
    denominator := (4*float64(k) + 1) *
(4*float64(k) + 3) // (4k + 1)(4k + 3)
    term := numerator / denominator

    // Kalikan istilah tersebut ke produk
    product *= term
}

// Menampilkan output
fmt.Printf("Nilai akar 2 ≈ %.10f\n", product)
}

```

Screenshoot Output

```

PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Unguided3.go"
Masukkan nilai K: 10
Nilai akar 2 ≈ 1.4062058441
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Unguided3.go"
Masukkan nilai K: 100
Nilai akar 2 ≈ 1.4133387072
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\Unguided3.go"
Masukkan nilai K: 1000
Nilai akar 2 ≈ 1.4141252651
PS C:\ALpro Sem3> █

```


Deskripsi Program

Program diatas menghitung nilai pendekatan akar kuadrat dari 2 menggunakan rumus produk tak hingga berdasarkan nilai input integer K. Algoritma meminta user untuk memasukkan nilai K, yang menentukan batas iterasi. Selanjutnya, program menjalankan perulangan dari 0 hingga K, di mana setiap iterasi menghitung suatu suku dari persamaan menggunakan pembilang $(4k+2)^2(4k+2)^2$ dan penyebut $(4k+1)(4k+3)(4k+1)(4k+3)(4k+1)(4k+3)$. Suku tersebut dikalikan ke dalam hasil akhir, yang terus diperbarui setiap iterasi. Setelah perhitungan selesai, program menampilkan hasil pendekatan akar kuadrat dari 2 dengan 10 angka desimal.

4. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

Soal Studi Case

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

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    // Input berat parsel
    var beratParcel_45 int
    fmt.Print("Masukkan berat parcel (gram): ")
    fmt.Scan(&beratParcel_45)

    // Convert ke kg dan gram
    kg := beratParcel_45 / 1000
    gram := beratParcel_45 % 1000

    // hitung biaya dasar
    biayaTotal := kg * 10000

    //detailbiaya
    KGcost := kg * 10000
    gcost := 0
    if gram >= 500 {
        gcost = gram * 5
    } else {
        gcost = gram * 15
    }

    // Menentukan biaya tambahan untuk gram
    if beratParcel_45 <= 10000 { // Jika berat total
kurang dari atau sama dengan 10 kg
        if gram >= 500 {
            biayaTotal += gram * 5 // Rp. 5 per gram
        }
    }
    jika >= 500 gr
    } else {
        biayaTotal += gram * 15 // Rp. 15 per gram
    }
    jika < 500 gr
```

```

    }

    // Menampilkan detail dan total biaya
    fmt.Printf("Detail berat: %d kg + %d gr\n", kg,
gram)
    fmt.Printf("detail biaya: Rp. %d + %d\n",
KGcost,gcost)
    fmt.Printf("Total biaya: Rp. %d\n", biayaTotal)
}

```

Screenshoot Output

```

PS C:\ALpro Sem3> go run "c:\ALpro Sem3\UnguidedC1.go"
Masukkan berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
detail biaya: Rp. 80000 + 2500
Total biaya: Rp. 82500
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\UnguidedC1.go"
Masukkan berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
detail biaya: Rp. 90000 + 3750
Total biaya: Rp. 93750
PS C:\ALpro Sem3> 

```

Deskripsi Program

Program diatas menghitung biaya pengiriman parcel berdasarkan berat yang diinput oleh user dalam satuan gram. Algoritma pertama-tama mengonversi berat total parcel ke satuan kilogram dan gram. Setelah itu, biaya dasar dihitung sebesar Rp 10.000 per kilogram. Jika berat sisa dalam gram lebih dari atau sama dengan 500 gram, maka biaya tambahan dihitung Rp 5 per gram, sedangkan jika kurang dari 500 gram, biayanya dihitung Rp 15 per gram. Jika berat total parcel melebihi 10 kg, biaya tambahan untuk gram ditiadakan. Selanjutnya, program menampilkan

berat parcel, biaya per gram dan kilogram, dan biaya pengiriman keseluruhan.

5. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

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 5 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. A

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>	Bilangan: <u>7</u>
---------------------	--------------------

Faktor: 1 2 3 4 5 6 12 Prima: false	Faktor: 1 7 Prima: true
--	----------------------------

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    // Deklarasi variabel bilangan
    var b_45 int

    // Meminta Input user
    fmt.Print("Masukkan bilangan bulat (b > 1): ")
    fmt.Scan(&b_45)

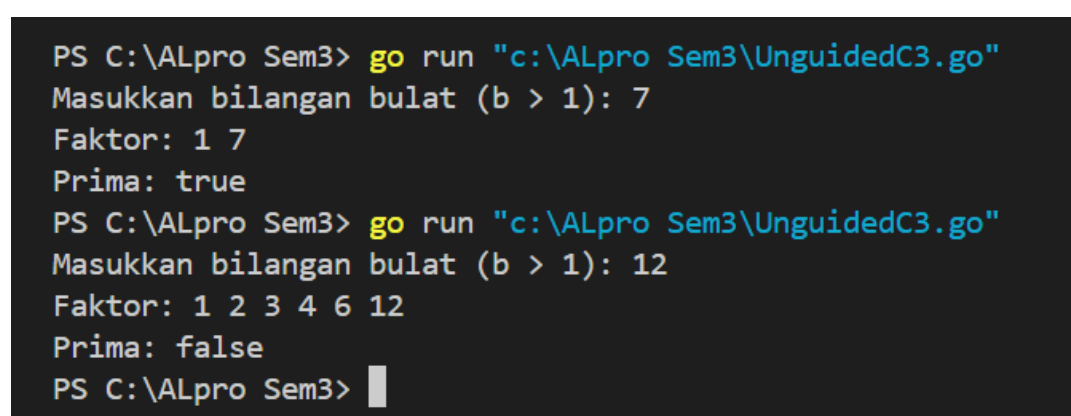
    // Memastikan B_45 lebih besar dari 1
    if b_45 <= 1 {
        fmt.Println("Bilangan harus lebih besar dari
1.")
        return
    }

    // Menampilkan faktor bilangan
    fmt.Printf("Faktor: ")
    factorCount := 0
    for i := 1; i <= b_45; i++ {
        if b_45%i == 0 { // cek apakah i adalah faktor
            b_45
                fmt.Printf("%d ", i)
                factorCount++ // Menghitung jumlah faktor
            }
        }
    }
    fmt.Println()
```

```
// Menentukan apakah b_45 adalah bilangan prima
isPrime := (factorCount == 2) // Bilangan prima
hanya memiliki dua faktor

// Menampilkan hasil apakah bilangan prima
(true/false)
fmt.Printf("Prima: %v\n", isPrime)
}
```

Screenshoot Output



```
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\UnguidedC3.go"
Masukkan bilangan bulat (b > 1): 7
Faktor: 1 7
Prima: true
PS C:\ALpro Sem3> go run "c:\ALpro Sem3\UnguidedC3.go"
Masukkan bilangan bulat (b > 1): 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\ALpro Sem3> █
```

Deskripsi Program

Program diatas bertujuan untuk menerima input bilangan bulat dari user, menampilkan faktor-faktornya, dan menentukan apakah bilangan tersebut merupakan bilangan prima. Algoritma pertama-tama memastikan bahwa bilangan yang diinput lebih besar dari 1. Program kemudian menghitung faktor-faktor bilangan tersebut dengan membagi bilangan secara berurutan dari 1 hingga bilangan itu sendiri dan mencatat berapa kali pembagian tersebut menghasilkan hasil tanpa sisa. Jika jumlah faktor yang ditemukan adalah 2 (yaitu 1 dan bilangan itu sendiri), program menyimpulkan bahwa bilangan tersebut adalah bilangan prima dan menampilkan hasilnya dalam bentuk true atau false. Output akhir menampilkan daftar faktor dan informasi tentang apakah angka tersebut prima atau tidak.