

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

Annasya Maulafidatu Zahra / 2311102246

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

Bahasa pemrograman GO memiliki struktur atau kerangka pemrograman yang wajib ada pada setiap programnya, yakni package main dan func main (). Selain kedua kerangka tadi, bahasa go juga memiliki beberapa struktur lain seperti fungsi, percabangan, perulangan, dan penyimpanan data, seperti array, slice, dan masi banyak lagi.

Bahasa GO atau yang sering disebut Golang ini, memiliki beberapa tipe data yang memiliki fungsi tersendiri, diantaranya terdapat interger untuk bilangan bulat, string untuk kalimat, char untuk karakter dan ASCII, float untuk decimal, dan bool untuk logika. Selain itu bahasa ini juga memiliki berbagai operator atau operasi perhitungan, seperti:

- 1) Operator penjumlahan (+)
- 2) Operator pengurangan (-)
- 3) Operator pembagian (/ dan %)
- 4) Operator perkalian (*)
- 5) Operator logika (&& dan ||)
- 6) Operator pangkat (^)
- 7) Operator perbandingan (==, <, >, !)

Bahasa Go memiliki beberapa perintah dasar seperti print untuk mencetak dan scan untuk menerima input. Apabila menambahkan \n pada kedua perintah tersebut maka akan terbentuk new line saat proses mencetak output maupun menerima input. Bahasa ini juga memiliki kondisi perulangan baik itu while-loop maupun for, dan percabangan, seperti if else dan switch case.

II. GUIDED

1. Guided 1

Soal Studi Case

Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berurutan 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 yang 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 E:\alpro2> go run "e:\alpro2\modul 2\guided1.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 E:\alpro2> go run "e:\alpro2\modul 2\guided1.go"
Percobaan 1: merah kuning ungu hijau
Percobaan 2: merah ungu kuning hijau
Percobaan 3: merah kuning hijau ungu
Percobaan 4: kuning merah hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
PS E:\alpro2> 

```

Deskripsi Program

Program di atas dibuat untuk memeriksa apakah urutan warna yang diberikan oleh pengguna sesuai dengan urutan warna yang telah ditentukan. Urutan warna yang benar adalah "merah", "kuning", "hijau", dan "ungu". Program ini memberikan kesempatan kepada pengguna untuk mencoba memasukkan urutan warna tersebut sebanyak lima kali. Setiap input dari

pengguna diproses dan dibandingkan dengan urutan warna yang benar, jika pada salah satu percobaan urutan yang dimasukkan tidak sesuai, maka program akan menandainya sebagai gagal.

Cara kerja program dimulai dengan membaca input pengguna menggunakan ``bufio.NewReader`` dan memisahnya berdasarkan spasi untuk mendapatkan masing-masing warna yang dimasukkan. Setelah itu, program membandingkan setiap elemen warna dengan urutan yang benar. Jika semua warna sesuai pada setiap percobaan, program menampilkan "BERHASIL: true". Namun, jika ada kesalahan dalam urutan warna, meskipun hanya satu percobaan, program akan menampilkan "BERHASIL: false". Program menggunakan sebuah variabel ``success`` untuk menentukan apakah percobaan berhasil atau gagal.

2. Guided 2

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
$72.5 < NAM \leq 80$	AB
$65 < NAM \leq 72.5$	B
$57.5 < NAM \leq 65$	BC
$50 < NAM \leq 57.5$	C
$40 < NAM \leq 50$	D
$NAM \leq 40$	E

Buatlah program yang menerima inputan nilai dan mencetak output NMK nya!

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 = "AB"

    } else if nam > 65{

        nmk = "B"

    } else if nam > 57.5{

        nmk = "BC"

    } else if nam > 50{

        nmk = "C"
```

```

    }else if nam > 40{

        nmk = "D"

    }else{

        nmk = "E"

    }

    //menampilkan hasil

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

```

Screenshoot Output

```

Nilai Indeks untuk nilai 87.00 adalah A
PS E:\alpro2> go run "e:\alpro2\guided2.go"
Masukkan Nilai: 77go run "e:\alpro2\guided2.go"
Nilai Indeks untuk nilai 77.00 adalah AB
PS E:\alpro2> go run "e:\alpro2\guided2.go"
# command-line-arguments
.\guided2.go:32:2: syntax error: unexpected literal 7 after top level declaration
PS E:\alpro2> go run "e:\alpro2\guided2.go"
Masukkan Nilai: 87
Nilai Indeks untuk nilai 87.00 adalah A
PS E:\alpro2> go run "e:\alpro2\guided2.go"
Masukkan Nilai: 77
Nilai Indeks untuk nilai 77.00 adalah AB
PS E:\alpro2> go run "e:\alpro2\guided2.go"
Masukkan Nilai: 67
Nilai Indeks untuk nilai 67.00 adalah B
PS E:\alpro2> go run "e:\alpro2\guided2.go"
Masukkan Nilai: 57
Nilai Indeks untuk nilai 57.00 adalah C
PS E:\alpro2> go run "e:\alpro2\guided2.go"
Masukkan Nilai: 47
Nilai Indeks untuk nilai 47.00 adalah D
PS E:\alpro2> go run "e:\alpro2\guided2.go"
Masukkan Nilai: 37
Nilai Indeks untuk nilai 37.00 adalah E
PS E:\alpro2>

```


Deskripsi Program

Program di atas dibuat untuk mengonversi nilai numerik menjadi nilai huruf (indeks) berdasarkan rentang tertentu. Pengguna diminta untuk memasukkan nilai berupa angka desimal, yang kemudian akan diproses oleh program untuk menentukan nilai huruf yang sesuai. Nilai huruf tersebut mengikuti standar umum seperti A, AB, B, BC, C, D, dan E, tergantung pada berapa besar nilai yang dimasukkan

Cara kerja program dimulai dengan mendeklarasikan dua variabel, `nam` untuk menyimpan nilai dan `nmk` untuk menyimpan nilai huruf. Setelah pengguna memasukkan nilai, selanjutnya program memeriksa nilai tersebut dengan beberapa kondisi menggunakan pernyataan `if-else`. Jika nilai lebih dari 80, maka nilai hurufnya adalah "A", jika lebih dari 72.5 menjadi "AB", dan seterusnya hingga di bawah 40 yang mendapatkan "E". Terakhir, program menampilkan hasil konversi nilai numerik menjadi huruf dengan menampilkan nilai desimal dua angka di belakang koma beserta nilai huruf yang sesuai.

3. Guided 3

Soal Studi Case

Buatlah program yang digunakan untuk menghitung hasil penjumlahan 5 bilangan bulat. Masukan terdiri dari lima bilangan bulat a, b, c, d, dan e. Keluaran berupa bilangan hasil penjumlahan lima bilangan bulat a,b,c,d, dan e.

Sourcecode

```
package main

import (
    "fmt"
)

func main(){
    var a,b,c,d,e int
    var hasil int
```

```
fmt.Print("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

```
PS E:\alpro2> go run "e:\alpro2\guided3.go"
Inputkan angka: 6 7 8 9 10
Hasil Penjumlahan 6 7 8 9 10 adalah 40
PS E:\alpro2> 
```

Deskripsi Program

Program di atas dibuat untuk menjumlahkan lima angka yang dimasukkan oleh pengguna. Program meminta input lima buah angka secara bersamaan, kemudian menghitung hasil penjumlahan dari angka-angka tersebut. Setelah itu, program akan menampilkan hasilnya dalam bentuk output yang menunjukkan angka-angka yang dijumlahkan dan hasil akhirnya. Program ini sederhana dan efektif untuk menghitung penjumlahan dari lima bilangan bulat.

Cara kerja program dimulai dengan mendeklarasikan lima variabel `a`, `b`, `c`, `d`, dan `e` untuk menyimpan input angka dari pengguna, serta variabel `hasil` untuk menyimpan hasil penjumlahan. Pengguna diminta untuk memasukkan lima angka sekaligus melalui fungsi `fmt.Scanln`. Setelah semua angka dimasukkan, program melakukan penjumlahan sederhana dengan menambahkan nilai dari setiap variabel dan menyimpan hasilnya ke dalam variabel `hasil`. Terakhir, program menampilkan hasil penjumlahan beserta angka-angka yang dimasukkan sebelumnya dengan format yang jelas.

I. UNGUIDED

1. Unguided 1

Soal Studi Case

Suatu pita (string) berisi kumpulan nama – nama bunga yang dipisahkan oleh spasi dan '-', contoh pita diilustrasikan seperti berikut ini.

Pita: mawar – melati – tulip – teratai – kamboja – anggrek

Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita. Lalu modifikasilah program tersebut, dimana proses input akan berhenti apabila user mengetikkan 'SELESAI'. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita.

Sourcecode

```
package main

import "fmt"

func main(){

    Bunga := [] string{"Mawar", "Melati", "Tulip",
"Teratai", "Kamboja", "Anggrek"}

    var banyak int

    fmt.Print("Masukkan Bilangan: ")

    fmt.Scan (&banyak)

    fmt.Println(" ")

    if banyak == 0{

        fmt.Println("Program tidak menerima input 0!")

    }else{

        for a:= 1; a <= banyak; a++){

            bagi := a%6

            if bagi == 1{
```

```

        fmt.Println ("Bunga ", a ,": " + Bunga[0])
    }else if bagi == 2{
        fmt.Println ("Bunga ", a ,": " + Bunga[1])
    }else if bagi == 3{
        fmt.Println ("Bunga ", a ,": " + Bunga[2])
    }else if bagi == 4{
        fmt.Println ("Bunga ", a ,": " + Bunga[3])
    }else if bagi == 5{
        fmt.Println ("Bunga ", a ,": " + Bunga[4])
    }else if bagi == 0{
        fmt.Println ("Bunga ", a ,": " + Bunga[5])
    }
}

fmt.Print("Pita : ")

for a:= 1; a <= banyak; a++){

    sisa := a%6

    if sisa == 1{

        fmt.Print (Bunga[0] + " - ")

    }else if sisa == 2{

        fmt.Print (Bunga[1] + " - ")

    }else if sisa == 3{

        fmt.Print (Bunga[2] + " - ")

    }else if sisa == 4{

        fmt.Print (Bunga[3] + " - ")

    }else if sisa == 5{

        fmt.Print (Bunga[4] + " - ")
    }
}

```

```
        }else if sisa == 0{
            fmt.Print (Bunga[5] + " - ")
        }
    }
}

//Modifikasi program

fmt.Println("\n")

fmt.Println("\n Input Bunga Pita")

for ke:=1; ke>0; ke++){

    var Bunga string

    fmt.Print("Bunga ", ke, ": ")

    fmt.Scan(&Bunga)

    if Bunga == "SELESAI"{

        break

    }

}

}
```

Screenshoot Output

```

PS E:\alpro2> go run "e:\alpro2\unguided1.go"
Masukkan Bilangan: 7

Bunga 1 : Mawar
Bunga 2 : Melati
Bunga 3 : Tulip
Bunga 4 : Teratai
Bunga 5 : Kamboja
Bunga 6 : Anggrek
Bunga 7 : Mawar
Pita : Mawar - Melati - Tulip - Teratai - Kamboja - Anggrek - Mawar -

Input Bunga Pita
Bunga 1: Tulip
Bunga 2: Teratai
Bunga 3: SELESAI
PS E:\alpro2>

```

Deskripsi Program

Program di atas dibuat untuk menampilkan deretan nama bunga berdasarkan input angka dari pengguna, dan kemudian menampilkan pita bunga dengan pola yang berulang. Pengguna diminta untuk memasukkan sebuah bilangan, yang kemudian akan digunakan sebagai jumlah iterasi untuk menampilkan nama-nama bunga dari daftar yang sudah ditentukan. Selain itu, di bagian akhir program, ada tambahan fitur di mana pengguna bisa memasukkan nama bunga untuk pita bunga hingga mengetikkan kata "SELESAI" sebagai tanda berhenti.

Cara kerjanya dimulai dengan mendeklarasikan array `Bunga` yang berisi beberapa nama bunga seperti "Mawar", "Melati", "Tulip", dan lainnya. Setelah itu, pengguna diminta memasukkan bilangan dengan fungsi `fmt.Scan`. Jika pengguna memasukkan angka 0, program menolak dan menampilkan pesan bahwa 0 tidak diterima. Jika bilangan lebih dari 0, program masuk ke perulangan `for` yang mencetak nama bunga sesuai urutan, di mana nama bunga dipilih berdasarkan hasil operasi modulo dari angka iterasi. Lalu, program juga mencetak pita bunga dengan pola yang sama. Bagian terakhir dari program memberikan kesempatan bagi pengguna untuk terus memasukkan nama bunga satu per satu hingga mengetikkan "SELESAI" sebagai tanda akhir.

2. Unguided 2
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 9kg atau lebih.

Lalu modifikasilah 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 150kg atau salah satu kantong beratnya negatif.

Sourcecode

Sebelum Dimodifikasi

```
package main

import "fmt"

func main(){

    var kiri float32

    var kanan float32

    for{

        fmt.Print("Masukkan Beban: ")

        fmt.Scan(&kiri, &kanan)

        if kanan >= 9.0 || kiri >= 9.0{

            break

        }

    }

}
```

```
}
```

Sesudah Dimodifikasi

```
package main

import "fmt"

func main(){

    var kiri float32

    var kanan float32

    for{

        fmt.Print("Masukkan Beban: ")

        fmt.Scan(&kiri, &kanan)

        var oleng bool = false

        total := kiri + kanan

        if (kiri - kanan) >= 9 || (kanan - kiri) >= 9{

            oleng = true

        }

        if total > 150 || kanan <0 || kiri <0{

            break

        }

        fmt.Println("Sepeda Motor Pak Andi akan Oleng : ",

oleng)

    }

}
```

Screenshoot Output


```

PS E:\alpro2> go run "e:\alpro2\tempCodeRunnerFile.go"
Masukkan Beban: 8 6
Masukkan Beban: 2 3
Masukkan Beban: 7 8
Masukkan Beban: 9 10
PS E:\alpro2> go run "e:\alpro2\unguided2_modif.go"
Masukkan Beban: 9 10
Sepeda Motor Pak Andi akan Olenng : false
Masukkan Beban: 12 34
Sepeda Motor Pak Andi akan Olenng : true
Masukkan Beban: 23 89
Sepeda Motor Pak Andi akan Olenng : true
Masukkan Beban: 67 85
PS E:\alpro2>

```

Deskripsi Program

Program di atas dibuat untuk menunjukkan keseimbangan beban pada sepeda motor Pak Andi berdasarkan dua sisi, yaitu sisi kiri dan sisi kanan. Pengguna diminta untuk memasukkan dua nilai beban, satu untuk beban di sisi kiri dan satu untuk sisi kanan. Program ini akan menentukan apakah sepeda motor akan oleng (tidak seimbang) atau tidak, tergantung pada perbedaan beban antara kedua sisi. Jika perbedaan beban melebihi 9 satuan atau total beban melebihi 150, program akan keluar dari loop.

Cara kerja program dimulai dengan meminta input dua angka untuk beban kiri dan kanan. Program kemudian menjumlahkan kedua beban dan memeriksa apakah perbedaan antara beban kiri dan kanan lebih besar atau sama dengan 9. Jika ya, maka variabel `oleng` akan bernilai `true`, menandakan sepeda motor oleng. Program juga mengecek jika total beban lebih dari 150 atau salah satu beban negatif, maka loop akan berhenti. Setelah setiap perulangan, program akan menampilkan apakah sepeda motor oleng atau tidak berdasarkan hasil perhitungan.

3. Unguided 3

Soal Studi Case

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

Buatlah sebuah program yang menerima input sebuah bilangan k, kemudian menghitung dan menampilkan nilai f(k) sesuai dengan persamaan yang ada di atas.

Sourcecode

```

package main

import "fmt"

func main(){

    var k float32

    fmt.Print("Masukkan Nilai k: ")

    fmt.Scan(&k)

    atas := (4*k + 2) * (4*k + 2)

    bawah := (4*k +1) *( 4*k + 3)

    var fungsi float32 = atas / bawah

    fmt.Print("Nilai fungsi f dari k: ", fungsi)

}

```

Screenshoot Output

```

PS E:\alpro2> go run "e:\alpro2\tempCodeRunnerFile.go"
Masukkan Nilai k: 100
Nilai fungsi f dari k: 1.0000062
PS E:\alpro2> 98
98
PS E:\alpro2> go run "e:\alpro2\tempCodeRunnerFile.go"
Masukkan Nilai k: 98
Nilai fungsi f dari k: 1.0000064
PS E:\alpro2> █

```

Deskripsi Program

Program di atas dibuat untuk menghitung suatu fungsi matematika berdasarkan nilai `k` yang dimasukkan oleh pengguna. Jadi, pengguna hanya perlu memasukkan nilai `k`, lalu program akan melakukan perhitungan dan menampilkan hasilnya.

Cara kerjanya simpel, setelah pengguna memasukkan nilai `k`, program menghitung dua hal: bagian atas dan bagian. Setelah itu, hasil dari bagian atas dibagi dengan hasil dari bagian bawah, dan program akan menampilkan hasil akhir perhitungan tersebut. Singkatnya, program ini meminta angka `k`, menghitung dengan rumus, lalu menampilkan hasil akhirnya.

4. Unguided 4

Soal Studi Case

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

- 1) Parcel harus dihitung per kilogramnya, dengan harga Rp 10.000/kg
- 2) Sisa gram yang tidak mencapai 1 kg, akan dihitung harganya per gram sesuai dengan ketentuan:
 - a. Berat < 500 gr, maka harga per gramnya adalah 15
 - b. Berat > 500 gr, maka harga per gramnya adalah 5
 - c. Gratis, apabila berat kilogramnya > 10 kg

Sourcecode

```
package main

import "fmt"

func main(){

    var parcel, kg, gr, hargakg, hargagr, total uint32

    fmt.Print("Berat Parcel(gram): ")

    fmt.Scan(&parcel)

    gr = parcel % 1000

    kg = (parcel - gr) / 1000

    fmt.Println("Detail Berat: ", kg , "Kg + ", gr, "gr")

    hargakg = kg * 10000

    if kg <= 10{

        if gr < 500{

            hargagr = gr * 15

            total = hargakg + hargagr
```

```
        fmt.Println("Detail Biaya: ", hargakg, " + ",  
hargagr)  
  
        fmt.Println("Total Biaya: ", total)  
    }else if gr >= 500{  
        hargagr = gr * 5  
  
        total = hargakg + hargagr  
  
        fmt.Println("Detail Biaya: ", hargakg, " + ",  
hargagr)  
  
        fmt.Println("Total Biaya: ", total)  
    }  
    }else if kg > 10{  
        hargagr = 0  
  
        total = hargakg + hargagr  
  
        fmt.Println("Detail Biaya: ", hargakg, " + ",  
hargagr)  
  
        fmt.Println("Total Biaya: ", total)  
    }  
}
```

Screenshoot Output

```

PS E:\alpro2> go run "e:\alpro2\tempCodeRunnerFile.go"
Berat Parsel(gram): 8900
Detail Berat: 8 Kg + 900 gr
Detail Biaya: 80000 + 4500
Total Biaya: 84500
PS E:\alpro2> go run "e:\alpro2\tempCodeRunnerFile.go"
Berat Parsel(gram): 2300
Detail Berat: 2 Kg + 300 gr
Detail Biaya: 20000 + 4500
Total Biaya: 24500
PS E:\alpro2> go run "e:\alpro2\tempCodeRunnerFile.go"
Berat Parsel(gram): 20500
Detail Berat: 20 Kg + 500 gr
Detail Biaya: 200000 + 0
Total Biaya: 200000
PS E:\alpro2>

```

Deskripsi Program

Program di atas dirancang untuk menghitung biaya pengiriman berdasarkan berat parcel yang dimasukkan oleh pengguna dalam satuan gram. Pengguna diminta untuk memasukkan berat parcel, dan program akan memecah berat tersebut menjadi satuan kilogram dan gram. Selanjutnya, biaya dihitung berdasarkan berat tersebut, dengan aturan bahwa biaya untuk kilogram pertama hingga sepuluh dan gram akan berbeda tergantung pada jumlah gram yang dimiliki.

Cara kerja program dimulai dengan meminta input berat parcel dari pengguna dalam gram. Program kemudian memisahkan berat itu menjadi kilogram dan sisa gram. Setelah mengetahui berat dalam dua satuan tersebut, program menghitung biaya pengiriman. Akhirnya, program menampilkan detail biaya untuk kilogram dan gram, serta total biaya pengiriman yang harus dibayar pengguna.

5. Unguided 5

Soal Studi Case

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

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, modifikasilah program supaya dapat menentukan apakah b merupakan bilangan prima.

Sourcecode

```
package main

import "fmt"

func main(){
    var b, bagi, a int
    var banyak int = 0
    var faktor [] int
    var prima bool = false

    fmt.Print("Masukkan Bilangan Bulat: ")

    fmt.Scan (&b)

    for a= 1; a <= b; a++){
        bagi = b % a

        if bagi == 0{
            faktor = append(faktor, a)
            banyak++
        }
    }

    fmt.Println("Faktor dari bilangan ", b, " adalah ")

    for _, f := range faktor {
        fmt.Print(f, " ")
    }

    fmt.Println()
```

```

        if banyak == 2{
            prima = true
        }

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

```

Screenshoot Output

```

PS E:\alpro2> go run "e:\alpro2\unguided5.go"
Masukkan Bilangan Bulat: 64
Faktor dari bilangan 64 adalah
1 2 4 8 16 32 64
Prima: false
PS E:\alpro2> go run "e:\alpro2\unguided5.go"
Masukkan Bilangan Bulat: 23
Faktor dari bilangan 23 adalah
1 23
Prima: true
PS E:\alpro2>

```

Deskripsi Program

Program ini dibuat untuk mencari faktor dari sebuah bilangan bulat yang dimasukkan oleh pengguna dan juga menentukan apakah bilangan itu adalah bilangan prima. Jadi, ketika pengguna memasukkan angka, program akan mencari semua angka yang bisa membagi angka tersebut tanpa sisa. Jika bilangan hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri, maka program akan memberi tahu bahwa bilangan tersebut adalah bilangan prima.

Cara kerjanya sederhana. Program akan mulai dengan meminta pengguna untuk memasukkan angka, lalu akan mencari semua faktor dengan memeriksa setiap angka dari 1 hingga angka yang dimasukkan. Jika angka bisa membagi tanpa sisa, angka itu akan dicatat sebagai faktor. Setelah itu, program akan menampilkan semua faktor yang ditemukan dan memeriksa jumlahnya. Jika jumlah faktor hanya dua, program akan menampilkan bahwa angka tersebut adalah bilangan prima.

