

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**Afif Rijal Azzami / 2311102235**

**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

### A. Struktur Program Go

Dalam kerangka program yang ditulis dalam bahasa pemrograman Go, program utama selalu mempunyai dua komponen berikut:

- **package main** merupakan penanda bahwa file ini berisi program utama.
- **func main()** berisi kode utama dari sebuah program Go.

#### 1. Kompilasi

Beberapa bahasa pemrograman dirancang untuk diimplementasikan sebagai interpreter dan lainnya sebagai kompilator. Interpreter akan membaca setiap baris instruksi dan kemudian langsung mengeksekusinya, dengan hanya sedikit pemeriksaan apakah penulisan keseluruhan program sudah benar atau belum. Kompilator akan memeriksa keseluruhan program sumber dan kemudian mengubahnya menjadi program eksekutabel, sehingga konsistensi penulisan Formatics 100 (seperti penggunaan tipe data) sudah diperiksa sebelum eksekusi. Selain itu karena program dibuat menjadi eksekutabel lebih dahulu, proses optimasi dapat dilakukan sehingga program menjadi sangat efisien.

### B. Tipe data

#### 1. Data dan Variabel

Tipe data yang umum tersedia adalah integer, real, boolean, karakter, dan string.

Notasi tipe dasar	Tipe dalam Go	Keterangan
integer	int int8 int32 //rune int64 uint uint8 //byte uint32 uint64	bergantung platform 8 bit: -128..127 32 bit: -10 <sup>9</sup> ..10 <sup>9</sup> 64 bit: -10 <sup>19</sup> ..10 <sup>19</sup> bergantung platform 0..255 0..4294967295 0..(2 <sup>64</sup> -1)
real	float32 float64	32bit: -3.4E+38 .. 3.4E+38 64bit: -1.7E+308 .. 1.7E+308
boolean (atau logikal)	<b>bool</b>	<b>false</b> dan <b>true</b>
karakter	byte	tabel tabel
string	string	

Bahasa Go menganut kesesuaian tipe data yang ketat. Tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi, bahkan tipe data masih yang sejenis, misalnya masih sama- sama integer (Int dan Int32).

## 2. Operator dalam Golang

Operator aritmatika adalah operator yang digunakan untuk operasi yang sifatnya perhitungan. Go mendukung beberapa operator aritmatika standar.

		Keterangan
+	string integer dan real	konkatenasi 2 string penjumlahan
- * /	integer dan real	operasi pengurangan, perkalian, dan pembagian
%	integer	operasi sisa pembagian integer (modulo)
&   ^ &^	integer	operasi <b>per-bit</b> AND OR XOR AND-NOT
<< >>	integer dan unsigned integer	operasi geser bit unsigned integer yang
< <= >= > == !=	selain boolean	komparasi menghasilkan nilai boolean komparasi karakter sesuai dengan posisi karakter tersebut dalam tabel ASCII/UTF-16 komparasi string sesuai dengan operasi karakter per karakter, dimulai dari karakter paling kiri (awal)
&&    !	boolean	operasi <b>boolean</b> AND, OR, dan NOT
* &	variabel apasaja	mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel

## II. GUIDED 1

### 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, " ")
```

### Screenshot Output



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\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: kuning merah ungu hijau
BERHASIL: false
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>
```

### Deskripsi Program

Sintaks tersebut adalah program untuk mengecek urutan warna yang kita inputkan pada program dengan lima kali perulangan, jika urutan warna yang di-inputkan benar maka program akan menampilkan true, dan jika urutan warna ada yang salah disalahsatu perulang maka program akan menampilkan false.

### III. GUIDED 2

#### Sourcecode

```
package main

import "fmt"

func main(){

    var a, b, c, d, e int

    var hasil int

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

    hasil = a+b+c+d+e

    fmt.Println("Hasil Penjumlahan adalah =", hasil)

}
```

#### Screenshoot Output



#### Deskripsi Program

Sintaks tersebut adalah program perjumlahan sederhana dengan mendeklarasikan lima variabel yang bertipe data integer dan memasukan rumus  $a+b+c+d+e$  yang disimpan pada variabel hasil, pada terminal user akan memasukan lima bilangan dan hasilnya akan ditampilkan.

## IV. GUIDED 3

### 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



```
PROBLEMS 60 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\guided3.go"
Masukkan Nilai: 85
Nilai Indeks untuk nilai 85.00 adalah A
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> █
```

## Deskripsi Program

Sintaks tersebut adalah program untuk mengonversikan nilai ke dalam indeks huruf dengan pengondisian if else, kondisi pertama jika nilai lebih dari 80 maka program akan menampilkan A, kondisi kedua jika nilai lebih dari 72,5 maka program akan menampilkan B, kondisi ketiga jika nilai lebih dari 65 maka program akan menampilkan C, kondisi keempat jika nilai lebih dari 50 maka program akan menampilkan D, kondisi kelima jika nilai lebih dari 40 maka program akan menampilkan E, dan terakhir jika selain termasuk kondisi di atas maka program akan menampilkan F.



## V. UNGUIDED 1

### Sourcecode

```
package main

import "fmt"

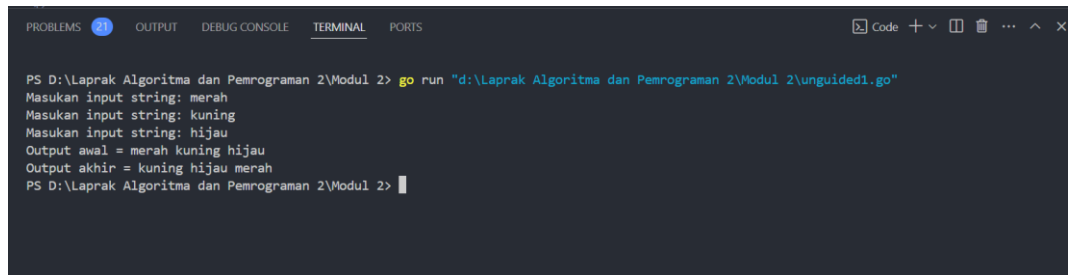
func main() {

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 Output



```
PROBLEMS 21 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\unguided1.go"
Masukan input string: merah
Masukan input string: kuning
Masukan input string: hijau
Output awal = merah kuning hijau
Output akhir = kuning hijau merah
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> |
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menukarkan posisi pada string yg di-inputkan oleh user, pertama kita menyimpan string kesatu di variabel temp untuk menyimpan nilai sementara, kedua kita menukarkan posisi string kedua menjadi posisi kesatu (satu = dua), ketiga kita menukarkan posisi string ketiga menjadi kedua (dua = tiga), dan terakhir kita mengembalikan string pertama yang disimpan sementara di posisi ketiga (tiga = temp), dan pada terminal akan memunculkan output awal sebelum ditukar dan output akhir setelah ditukar.

## VI. UNGUIDED 2

### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    for {
        var tahun int

        fmt.Print("Tahun: ")

        fmt.Scanln(&tahun)
```

```
if Cek_Kabisat(tahun) {  
  
    fmt.Println("Kabisat: true")  
  
    } else {  
  
        fmt.Println("Kabisat: false")  
  
    }  
  
}  
}
```

```
func Cek_Kabisat(tahun int) bool {  
  
    // Tahun kabisat habis dibagi 400  
  
    if tahun%400 == 0 {  
  
        return true  
  
    }  
  
  
    // Tahun kabisat habis dibagi 4 tetapi tidak habis dibagi  
100  
  
    if tahun%4 == 0 && tahun%100 != 0 {  
  
        return true  
  
    }  
  
  
    // Selain kondisi di atas, bukan tahun kabisat  
  
    return false  
  
}
```

## Screenshoot Output



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided2_cek_tahun kabisat.go"
Tahun: 2024
Kabisat: true
Tahun: 2023
Kabisat: false
Tahun: 2022
Kabisat: false
Tahun: 2021
Kabisat: false
Tahun: 
```

## Deskripsi Program

Sintaks tersebut adalah program untuk mengecek apakah tahun yang di-input user adalah kabisat atau bukan, pertama kita membuat perulangan yang tidak ada batasnya agar user bisa mengecek tahun sampe puas, untuk mengecek tahun kabisat kita membuat func yang dinamai cek\_kabisat, pada func tersebut berisi pengecekan menggunakan if, kondisi pertama jika tahun habis dibagi 400 maka program akan mengembalikan nilai true, kondisi kedua jika tahun habis dibagi 4 maka program akan mengembalikan nilai true, dan jika diluar kondisi tersebut program akan mengembalikan nilai false.

## VII. UNGUIDED 3

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jariJari int

    fmt.Print("Masukan jari-jari = ")
```

```

fmt.Scanln(&jariJari)

    volume := hitungVolume(jariJari)

    luas := hitungLuas(jariJari)

    fmt.Printf("Bola dengan jari-jari %d memiliki volume %.4f
dan luas kulit %.4f\n",

        jariJari, volume, luas)
}

func hitungVolume(jariJari int) float64 {


    return (4 / 3) * math.Pi * math.Pow(float64(jariJari), 3)
}

func hitungLuas(jariJari int) float64 {

    return 4 * math.Pi * math.Pow(float64(jariJari), 2)
}

```

## Screenshoot Output



```

PROBLEMS 33 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided3_hitungVolluasBola.go"
Masukan jari-jari = 15
Bola dengan jari-jari 15 memiliki volume 14137.1669 dan luas kulit 2827.4334
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>

```

## Deskripsi Program

Sintaks tersebut adalah program untuk menghitung volume dan luas pada bangun ruang bola yang jari-jarinya ditentukan oleh user, untuk menghitung volume dan luas kita membuat func hitungVolume dan hitungLuas, di func hitungVolume kita memasukan rumus volume bola yaitu 4 dibagi 3 dikalikan dengan nilai konstanta Pi (`math.Pi`) dan dikalikan pangkat 3 dari nilai jari-jari [ `(math.Pow(float64(jariJari),3)` ], dan pada func hitungLuas kita memasukan rumus luas bola yaitu 4 dikali nilai konstanta Pi (`math.Pi`) dan dikalikan pangkat 2 dari nilai jari-jari [ `(math.Pow(float64(jariJari),2)` ], dalam menghitung volume dan luas kenapa menggunakan float64 karna tipe data dari nilai konstanta Pi adalah float64.

## VIII. UNGUIDED 4

### Sourcecode

```
package main

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

func main() {
    var n int
    var bunga string
    var pita []string
```

```
// Meminta input jumlah bunga dari user

fmt.Print("N: ")

fmt.Scanln(&n)


// Meminta input nama bunga sebanyak N kali
// dan menyimpannya ke dalam slice pita
scanner := bufio.NewScanner(os.Stdin)

for i := 1; i <= n; i++ {

    fmt.Printf("Bunga %d: ", i)

    scanner.Scan()

    bunga = scanner.Text()


    // Jika user menginputkan "SELESAI", hentikan loop
    if bunga == "SELESAI" {

        break

    }


    pita = append(pita, bunga)

}


// Menggabungkan semua nama bunga di dalam pita
// dengan separator " - "

hasilPita := strings.Join(pita, " - ")
```

```
// Menampilkan isi pita dan jumlah bunga

fmt.Println("Pita:", hasilPita)

fmt.Println("Bunga:", len(pita))

}
```

## Screenshoot Output



```
PROBLEMS 89 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided5.go"
N: 5
Bunga 1: melati
Bunga 2: anggrek
Bunga 3: mawar
Bunga 4: kemboja
Bunga 5: SELESAI
Pita: melati - anggrek - mawar - kemboja
Bunga: 4
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> |
```

## Deskripsi Program

Sintaks tersebut adalah program dimana user akan memasukan nilai N untuk menjalankan perulangan, dan user akan meng-input nama bunga yang nantinya terminal akan menampilkan urutan nama-nama bunga tersebut yang dipisahkan spasi dan tanda stirp ( - ) dan jika user menginput kata SELESAI maka program akan berhenti.



## XI. UNGUIDED 5

### Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKiri, beratKanan float64
    var totalBerat float64
    var selesai bool

    for !selesai {
        // Meminta input berat belanjaan di kedua kantong

        fmt.Print("Masukan berat belanjaan di kedua kantong (kiri kanan): ")

        fmt.Scan(&beratKiri, &beratKanan)

        // Mengecek apakah ada kantong yang beratnya negatif

        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Berat tidak boleh negatif. Proses selesai.")
            break
        }
    }
}
```

```
// Menghitung total berat belanjaan

    totalBerat += beratKiri + beratKanan

    // Mengecek apakah total berat melebihi 150 kg

    if totalBerat > 150 {

        fmt.Println("Total berat melebihi 150 kg. Proses
selesai.")

        break

    }

    // Menghitung selisih berat antara kantong kiri dan kanan

    selisih := math.Abs(beratKiri - beratKanan)

    // Menampilkan hasil apakah selisih lebih dari 9 kg atau
tidak

    if selisih > 9 {

        fmt.Println("Sepeda motor Pak Andi akan oleng:", true)

    } else {

        fmt.Println("Sepeda motor Pak Andi akan oleng:",
false)

    }

    // Menampilkan total berat belanjaan

    fmt.Println("Total berat belanjaan:", totalBerat)
```

```
// Jika user memasukkan angka yang menyebabkan salah satu
kondisi, hentikan loop

    if beratKiri == 0 && beratKanan == 0 {

        selesai = true

        fmt.Println("Proses selesai.")

    }

}

}
```

## Screenshoot Output

```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided6.go"
Masukan berat belanja di kedua kantong (kiri kanan): 56 32
Sepeda motor Pak Andi akan oleng: true
Total berat belanja: 88
Masukan berat belanja di kedua kantong (kiri kanan): 25 25
Sepeda motor Pak Andi akan oleng: false
Total berat belanja: 138
Masukan berat belanja di kedua kantong (kiri kanan): 90 75
Total berat melebihi 150 kg. Proses selesai.
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>
```

```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided6.go"
Masukan berat belanja di kedua kantong (kiri kanan): 9 -3
Berat tidak boleh negatif. Proses selesai.
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>
```

## Deskripsi Program

Sintaks di atas adalah program untuk mengecek kantong motor apakah selisih berat kantong kanan sama yang kiri  $< 9$  dengan berat maksimal yang bisa diangkut  $< 150$ , jika selisih berat pada kedua kantong  $> 9$  maka program akan menampilkan motor oleng, pertama kita membuat perulangan for yang dimana perulangan akan terus berjalan hingga total berat  $> 150$  dan salah satu kantong beratnya negatif, untuk mengecek selisih berat kantong kita membuat rumus selisih yaitu berat kanan dikurangi berat kiri, dan untuk mengecek apakah motor itu oleng atau tidak kita membuat kondisi menggunakan if dimana jika selisih  $< 9$  maka motor tidak oleng, dan elsenya yaitu jika diluar dari kondisi yang sudah dibuat (selisih  $< 9$ ) maka motor akan oleng.

## X. UNGUIDED 6

### Sourcecode

```
package main

import (
    "fmt"
)

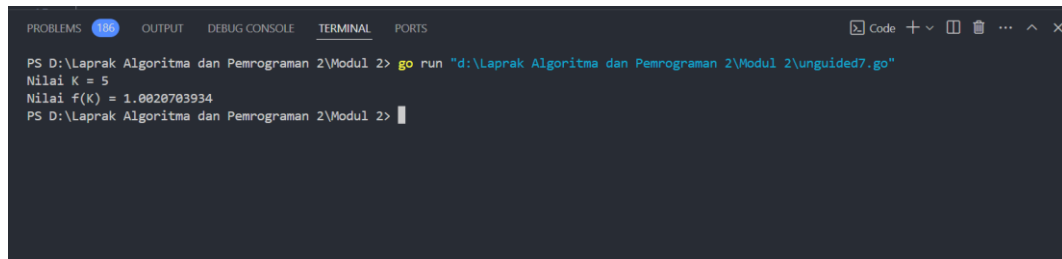
func main() {
    var k float64

    // Meminta input nilai K dari user
    fmt.Print("Nilai K = ")
    fmt.Scanln(&k)

    // Menghitung nilai f(K) sesuai dengan rumus
    //  $f(k) = ((4k + 2)^2) / ((4k + 1)(4k + 3))$ 
    atas := (4*k + 2) * (4*k + 2) //  $(4k + 2)^2$ 
    bawah := (4*k + 1) * (4*k + 3) //  $(4k + 1)(4k + 3)$ 
    fk := atas / bawah

    // Menampilkan hasil f(K)
    fmt.Printf("Nilai f(K) = %.10f\n", fk)
}
```

## Screenshoot Output



```
PROBLEMS 186 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided7.go"
Nilai K = 5
Nilai f(K) = 1.0020703934
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menghitung persamaan  $f(k)$  yang sudah ada dengan nilai  $k$  pada persamaan tersebut di-inputkan oleh user, pertama kita membuat tiga pengoperasian hitung, operasi pertama kita namai atas, yaitu untuk menghitung  $(4k + 2)^2$ , operasi kedua kita namai bawah, yaitu untuk menghitung  $(4k + 1)(4k + 3)$ , dan operasi terakhir kita namai hasil, yaitu untuk menghitung hasil operasi pertama dibagi hasil dari operasi kedua, hasil dari operasi ketiga adalah jawaban dari persamaan  $f(k)$  dan program akan memunculkan hasil dari persamaan  $f(k)$ .

## XI. UNGUIDED 7

### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var beratParsel int

    // Meminta input berat parsel dalam gram
    fmt.Print("Berat parsel (gram): ")
    fmt.Scanln(&beratParsel)

    // Menghitung berat dalam kilogram dan sisa gram
    beratKg := beratParsel / 1000
    sisaGram := beratParsel % 1000

    // Menghitung biaya pengiriman
    biaya := beratKg * 10000 // Rp. 10.000 per kg

    // Jika berat total lebih dari atau sama dengan 10 kg, sisa
    gram gratis
    if beratKg < 10 {
```

```
// Menghitung biaya tambahan berdasarkan sisa gram

    if sisaGram > 0 && sisaGram < 500 {

        biaya += sisaGram * 15 // Rp. 15 per gram jika kurang
dari 500 gram

    } else if sisaGram >= 500 {

        biaya += sisaGram * 5 // Rp. 5 per gram jika tidak
kurang dari 500 gram

    }

}

// Menampilkan detail biaya dan total

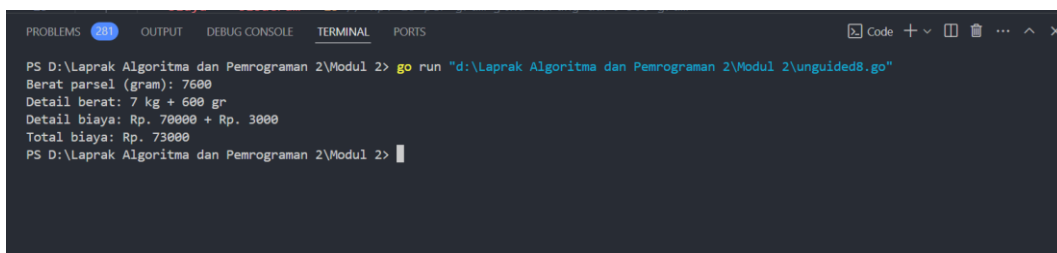
fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg,
sisaGram)

fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", beratKg*10000,
biaya-(beratKg*10000))

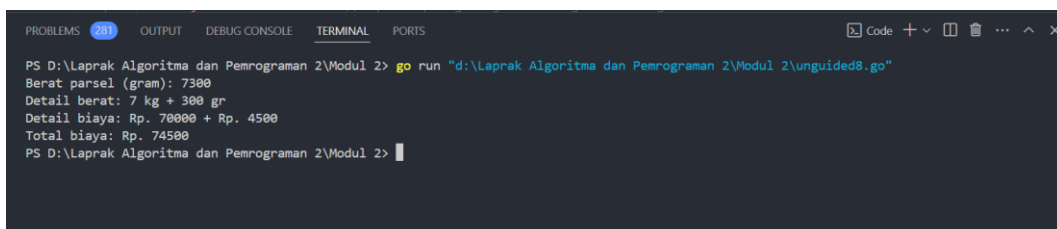
fmt.Printf("Total biaya: Rp. %d\n", biaya)

}
```

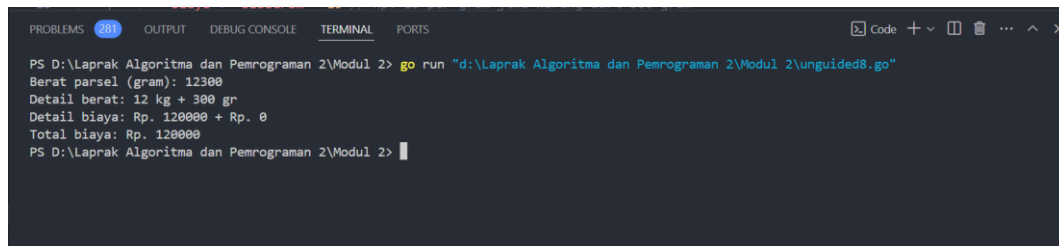
## Screenshoot Output



```
PROBLEMS 281 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided8.go"
Berat parcel (gram): 7500
Detail berat: 7 kg + 600 gr
Detail biaya: Rp. 70000 + Rp. 3000
Total biaya: Rp. 73000
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>
```



```
PROBLEMS 281 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\Modul 2\unguided8.go"
Berat parcel (gram): 7300
Detail berat: 7 kg + 300 gr
Detail biaya: Rp. 70000 + Rp. 4500
Total biaya: Rp. 74500
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>
```



```
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\unguided8.go"
Berat parcel (gram): 12300
Detail berat: 12 kg + 300 gr
Detail biaya: Rp. 120000 + Rp. 0
Total biaya: Rp. 120000
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>
```

## Deskripsi Program

Sintaks tersebut adalah program untuk menghitung ongkos kirim pada paket, setiap 1 kg paket dikenakan biaya 10 ribu, dan berat sisa gram pada paket terdapat dua kondisi, yaitu jika berat < 500 gram maka paket dikenakan biaya Rp 15 per gram, dan jika berat >= 500 gram maka paket dikenakan biaya Rp 15 per gram. Untuk menghitung biaya kita perlu memisahkan berat berdasarkan kg dengan cara berat paket dibagi 1000, dan untuk mencari sisa berat dengan cara berat paket hasil sisa bagi (%) 1000, untuk menentukan biaya per-kg maka kita menuliskan rumus, biaya = berat kg dikali 10 rb. Untuk menghitung biaya sisa gram kita menggunakan pengondisian if, untuk berat < 500 rumusnya, biaya (hasil biaya beratKg) ditambahkan hasil dari beratGram kali 15, dan untuk berat >=500 rumusnya, biaya (hasil biaya beratKg) ditambahkan hasil dari beratGram kali 5.

## XII. UNGUIDED 7

### Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var b int

    // Meminta input bilangan dari pengguna
    fmt.Print("Bilangan: ")
```



```
fmt.Scanln(&b)

// Memeriksa apakah bilangan valid (b > 1)
if b <= 1 {
    fmt.Println("Bilangan harus lebih besar dari 1.")
    return
}

// Mencari dan mencetak faktor bilangan
fmt.Print("Faktor: ")
for i := 1; i <= b; i++ {
    if b%i == 0 {
        fmt.Print(i, " ")
    }
}

fmt.Println()

// Memeriksa dan mencetak apakah bilangan prima
prima := true
for i := 2; i <= b/2; i++ {
    if b%i == 0 {
        prima = false
        break
    }
}

}
```

```

fmt.Print("Prima: ")

    if prima && b > 1 {

        fmt.Println("true")

    } else {

        fmt.Println("false")

    }

}

```

## Screenshoot Output

```

PROBLEMS 30% OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\unguided9.go"
Bilangan: 9
Faktor: 1 3 9
Prima: false
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>

```

```

PROBLEMS 30% OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2> go run "d:\Laprak Algoritma dan Pemrograman 2\unguided9.go"
Bilangan: 11
Faktor: 1 11
Prima: true
PS D:\Laprak Algoritma dan Pemrograman 2\Modul 2>

```

## Deskripsi Program

Sintaks tersebut adalah program untuk mengetahui faktor yang dimiliki oleh sebuah bilangan yang di-unput oleh user dan program akan menentukan apakah bilangan tersebut termasuk prima atau bukan, untuk mencari faktor pertama kita membuat perulangan yang dimana nilai  $i$  harus harus  $\leq$  bilangan yang di-input user ( $b$ ), kemudian kita menggunakan pengindisian dengan `if` jika  $b$  dibagi dengan  $i$  dan hasil dari sisa bagi  $= 0$  maka nilai  $i$  tersebut adalah faktor dari  $b$ . Untuk mengecek bilangan prima, pertama kita membuat perulangan dimana nilai  $i \leq b$  dibagi 2, jika nilai  $i$  terdapat bilangan yang sisa hasil bagi dengan  $b = 0$ , maka nilai tersebut bukan bilangan prima, karena bilangan prima adalah bilangan yang habis dibagi dengan 1 dan dirinya sendiri.