

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



Oleh:

HANAH NUR AZIZAH

2311102312

IF-11-07

**PROGRAM STUDY S1 INFORMATIKA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM PURWOKERTO
2024**

I. DASAR TEORI

2.1 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) Koding, Kompilasi, dan Eksekusi Go

Koding

- Tidak berbeda dengan penulisan program sumber dalam bahasa lain, program Go harus dibuat menggunakan penyunting teks dan disimpan dalam format teks, bukan dalam format dokumen (doc, docx, atau lainnya).
- Setiap program go disimpan dalam file teks dengan ekstensi *.go, dengan nama bebas. Sebaiknya nama file adalah nama untuk program tersebut.
- Setiap satu program lengkap Go disimpan dalam satu folder tersendiri. Nama folder merupakan nama program tersebut. Karena itu secara prinsip, satu program Go dapat dipecah dalam beberapa file dengan ekstensi *.go selama disimpan dalam folder yang sama.

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 gga konsistensi penulisan (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.

Go diimplementasikan sebagai kompilator. Berikut adalah contoh sesi yang biasa dilakukan saat mengkompilasi dan mengeksekusi program dalam bahasa Go:

- Panggil shell atau terminal (program/utiliti cmd.exe di Windows)
- Masuk ke dalam (cd) folder program (normalnya ada di C:\Users\go\src\ atau yang sejenis)
- Kemudian panggil perintah go build atau go build file.go untuk mengkompilasi file.go
- Jika gagal, akan muncul pesan eror yang sesuai, pelajari dengan baik pesan tersebut, perbaiki teks program sumber, kemudian ulangi proses build-nya.
- Jika berhasil maka pada folder tersebut akan dibuat program dengan nama yang sama dan diakhiri dengan.exe (untuk Windows)
- Panggil program eksekutabel tersebut dari terminal yang sama. Jangan memanggil program tersebut dengan mengklik eksekutabel tersebut dari folder karena program kalian hanya berbasis teks, bukan/belum dirancang dengan tampilan Windows.

Catatan

Semua proses terkait bahasa Go dilakukan melalui utilitas go. Beberapa opsi dengan utilitas go:

- **go build:** mengkompilasi program sumber yang ada dalam folder menjadi sebuah program.
- **go build file.go:** mengkompilasi program sumber file.go saja.
- **go fmt:** membaca semua program sumber dalam folder dan mereformat penulisannya agar sesuai dengan standar penulisan program sumber Go.
- **go clean:** membersihkan file-file dalam folder sehingga tersisa program sumber nya saja.

2.2 Tipe Data dan Instruksi Dasar

1) Data dan Variabel

Variabel adalah nama dari suatu lokasi di memori, yang data dengan tipe tertentu dapat disimpan

2) Intruksi Dasar

3) Konstanta Simbolik

Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut.

II. GUIDED

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

Source Code

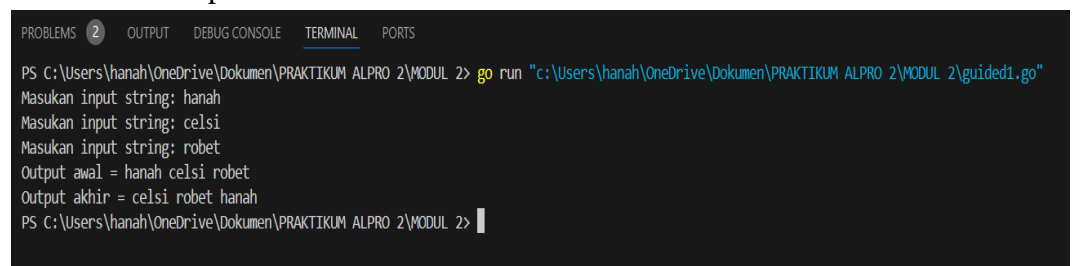
```
package main
import "fmt"

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

    temp = satu
    satu = dua
    dua = tiga
    tiga = temp

    fmt.Println("Output akhir = " + satu + " " + dua + " " +
tiga)
}
```

Screenshot Output



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\guided1.go"
Masukan input string: hanah
Masukan input string: celsi
Masukan input string: robet
Output awal = hanah celsi robet
Output akhir = celsi robet hanah
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> |
```

Penjelasan

Program di atas adalah program Go yang meminta pengguna untuk memasukkan tiga string secara berurutan. Setelah menerima input, program mencetak ketiga string tersebut dalam urutan awal. Kemudian, program melakukan pertukaran nilai dari ketiga string secara melingkar: string pertama disimpan sementara di variabel temp, string pertama diisi oleh string kedua, string kedua diisi oleh string ketiga, dan string ketiga diisi oleh nilai awal dari string pertama. Akhirnya, program mencetak urutan baru dari ketiga string tersebut setelah pertukaran dilakukan.

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

| | |
|---|--------------------------------------|
| 1 | Tahun: 2016 Kabisat: true |
| 2 | Tahun: 2000 Kabisat: true |
| 3 | Tahun: 2018 Kabisat: false |

Source Code

```
package main
import "fmt"
func main() {
    var tahun int
    fmt.Print("Tahun: ")
    fmt.Scanln(&tahun)
    kabisat := cekKabisat(tahun)
    fmt.Println("Kabisat: ", kabisat)
}
func cekKabisat(tahun int) bool {
    if tahun%400 == 0 {
        return true
    } else if tahun%4 == 0 {
        return true
    } else if tahun%100 == 0 {
        return false
    }
    return false
}
```

Screenshot Output

```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\guided2.go"
Tahun: 2026
Kabisat: false
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> 
```

Penjelasan

Program di atas adalah program Go yang menentukan apakah suatu tahun adalah tahun kabisat atau tidak. Pertama, program meminta input tahun dari pengguna dan menyimpannya dalam variabel tahun.

Kemudian, program memanggil fungsi cekKabisat() yang menerima tahun tersebut sebagai argumen dan mengembalikan nilai boolean (true jika kabisat, false jika bukan). Fungsi ini menggunakan aturan tahun kabisat: jika tahun habis dibagi 400, maka kabisat; jika habis dibagi 100, bukan kabisat; jika habis dibagi 4, maka kabisat. Hasilnya dicetak dengan label "Kabisat: ".

3. Dibaca nilai temperature dalam derajat celcius. Nyatakan temperature tersebut dalam Fahrenheit.

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

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

| |
|-------------------------------|
| Temperatur Celsius: <u>50</u> |
| Derajat Fahrenheit: 122 |

Lanjutkan program di atas, sehingga temperatur dinyatakan juga dalam derajat Reamur dan Kelvin.

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

| |
|-------------------------------|
| Temperatur Celsius: <u>50</u> |
| Derajat Reamur: 40 |
| Derajat Fahrenheit: 122 |
| Derajat Kelvin: 323 |

Source Code

```
package main
import (
    "fmt"
)
func main() {
    var celsius float64
    fmt.Print("Masukkan suhu dalam derajat Celsius: ")
    fmt.Scan(&celsius)
    fahrenheit := (celsius * 9 / 5) + 32
```

```

    reamur := celsius * 4 / 5
    kelvin := celsius + 273.15
    fmt.Printf("Temperatur Celsius: %.2f\n", celsius)
    fmt.Printf("Derajat Reamur: %.2f\n", reamur)
    fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)
    fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\guided3.go"
Masukkan suhu dalam derajat Celsius: 18
Temperatur Celsius: 18.00
Derajat Reamur: 14.40
Derajat Fahrenheit: 64.40
Derajat Kelvin: 291.15
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2>

```

Penjelasan

Program di atas adalah program Go yang mengonversi suhu dari derajat Celsius ke tiga skala suhu lainnya: Fahrenheit, Reamur, dan Kelvin. Program pertama-tama meminta pengguna memasukkan suhu dalam Celsius, lalu menghitung suhu dalam Fahrenheit menggunakan rumus $(\text{celsius} * 9 / 5) + 32$, dalam Reamur dengan rumus $\text{celsius} * 4 / 5$, dan dalam Kelvin dengan rumus $\text{celsius} + 273.15$. Setelah perhitungan, hasil konversi dari suhu tersebut ditampilkan dengan format desimal dua angka di belakang koma.

III. UNGUIDED

1. Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berturut-turut adalah 'merah', 'kuning', 'hijau', dan 'ungu' selama 5 kali percobaan berulang.
Buatlah sebuah program yang menerima input berupa warna dari ke 4 gelas reaksi sebanyak 5 kali percobaan. Kemudian program akan menampilkan **true** apabila urutan warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan **false** untuk urutan warna lainnya.

| | | | | |
|-----------------|--------------|---------------|--------------|--------------|
| Percobaan 1: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| Percobaan 2: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| Percobaan 3: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| Percobaan 4: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| Percobaan 5: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| BERHASIL: true | | | | |
| | | | | |
| Percobaan 1: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| Percobaan 2: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| Percobaan 3: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| Percobaan 4: | <u>ungu</u> | <u>kuning</u> | <u>hijau</u> | <u>merah</u> |
| Percobaan 5: | <u>merah</u> | <u>kuning</u> | <u>hijau</u> | <u>ungu</u> |
| BERHASIL: false | | | | |

Source Code

```
package main
import (
    "bufio"
    "fmt"
    "os"
    "strings"
)
//HANAH NUR AZIZAH (2311102312)
func main() {

    urutanWarna := []string{"merah", "kuning", "hijau",
"ungu"}
    reader := bufio.NewReader(os.Stdin)
    success := true
    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)
        colors := strings.Split(input, " ")

        for j := 0; j < 4; j++ {
            if colors[j] != urutanWarna[j] {
                success = false
                break
            }
        }
    }
    if !success {
        break
    }
}
```



```

    if success {
        fmt.Println("BERHASIL : true")
    } else {
        fmt.Println("BERHASIL : false")
    }
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\unguided2B1.go"
Percobaan 1: merah, kuning, hijau, ungu
BERHASIL : false
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2>

```

Penjelasan

Program di atas merupakan program Go yang dibuat untuk memvalidasi urutan warna pada sebuah percobaan kimia. Program ini membandingkan urutan warna yang diinputkan pengguna dengan urutan warna yang sudah ditentukan (merah, kuning, hijau, ungu). Pengguna akan diminta untuk memasukkan urutan warna sebanyak 5 kali percobaan. Program akan memeriksa setiap urutan warna yang dimasukkan dan memberikan hasil akhir berupa "true" jika semua urutan benar atau "false" jika ada satu saja urutan yang salah. Sederhananya, program ini berfungsi sebagai penilai otomatis untuk sebuah eksperimen sederhana dengan kriteria tertentu.

2. Unguided 2B no. 2

Source Code

```

package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)
//HANAH NUR AZIZAH (2311102312)
func main() {

    reader := bufio.NewReader(os.Stdin)

    fmt.Print("N: ")
    var N int
    for {
        input, err := reader.ReadString('\n')
        if err != nil {
            fmt.Printf("Error: %v", err)
            return
        }
    }
}

```

```

    }
    N, err = strconv.Atoi(strings.TrimSpace(input))
    if err != nil || N <= 0 {
        fmt.Println("Masukkan bilangan bulat positif")
    } else {
        break
    }
}

var pita string
var count int

for i := 1; i <= N; i++ {
    fmt.Printf("Bunga %d: ", i)
    input, err := reader.ReadString('\n')
    if err != nil {
        fmt.Printf("Error: %v", err)
        return
    }
    input = strings.TrimSpace(input)

    if strings.ToUpper(input) == "SELESAI" {
        break
    }

    if pita == "" {
        pita = input
    } else {
        pita = pita + " - " + input
    }
    count++
}

fmt.Println("Pita:", pita)
fmt.Printf("Bunga: %d\n", count)
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "
c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\unguided2B2.
go"
N: 2
Bunga 1: matahari
Bunga 2: tulip
Pita: matahari - tulip
Bunga: 2
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2>

```

Penjelasan

Program di atas merupakan program Go yang meminta pengguna untuk memasukkan sejumlah nama bunga, di mana jumlah bunga ditentukan oleh input awal berupa bilangan bulat positif N. Program menggunakan `bufio.NewReader` untuk membaca input dari pengguna dan mengonversi input tersebut menjadi integer. Setelah itu, program meminta pengguna memasukkan nama bunga satu per satu sebanyak N kali, atau pengguna dapat mengetik "SELESAI" untuk menghentikan

input lebih awal. Nama-nama bunga yang dimasukkan digabungkan menjadi satu string yang dipisahkan dengan tanda "-". Pada akhirnya, program menampilkan string yang berisi daftar bunga yang telah dimasukkan dan jumlah total bunga.

3. Unguided 2B no. 3

Source Code

```
package main
import (
    "fmt"
)
//HANAH NUR AZIZAH 2311102312
func main() {
    var beratKantongKiri, beratKantongKanan float64

    for {
        fmt.Print("Masukkan berat belanjaan di kedua kantong (misal: 50 60): ")
        fmt.Scan(&beratKantongKiri, &beratKantongKanan)

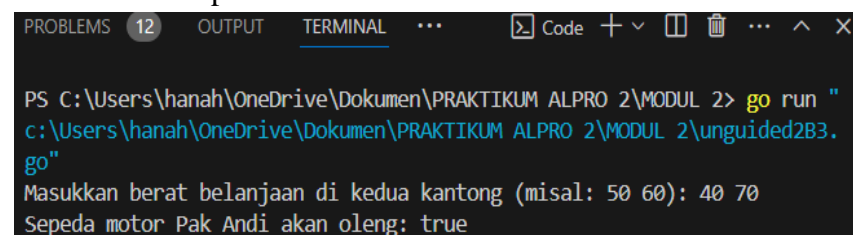
        if beratKantongKiri < 0 || beratKantongKanan < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        totalBerat := beratKantongKiri + beratKantongKanan
        if totalBerat > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        selisihBerat := beratKantongKiri - beratKantongKanan
        if selisihBerat < 0 {
            selisihBerat = -selisihBerat
        }

        if selisihBerat >= 9 {
            fmt.Println("Sepeda motor Pak Andi akan oleng: true")
        } else {
            fmt.Println("Sepeda motor Pak Andi akan oleng: false")
        }
    }
}
```

Screenshot Output



```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "
c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\unguided2B3.
go"
Masukkan berat belanjaan di kedua kantong (misal: 50 60): 40 70
Sepeda motor Pak Andi akan oleng: true
```

Penjelasan

Program di atas adalah program Go yang menghitung apakah sepeda motor akan oleng berdasarkan berat barang di dua kantong belanjaan. Pengguna diminta untuk memasukkan berat masing-masing kantong secara berulang. Program akan berhenti jika salah satu kantong memiliki berat negatif atau jika total berat kedua kantong melebihi 150 kg. Program kemudian menghitung selisih berat antara kedua kantong, dan jika selisihnya 9 kg atau lebih, sepeda motor dianggap akan oleng. Hasilnya ditampilkan sebagai "true" jika oleng dan "false" jika tidak oleng.

4. Unguided 2B no. 4

Source Code

```
package main

import (
    "fmt"
)

//HANAH NUR AZIZAH 2311102312
func hitungAkar2(iterasi int) float64 {
    hasilPerkalian := 1.0

    for indeks := 0; indeks <= iterasi; indeks++ {
        pembilang := (4*float64(indeks) + 2) * (4*float64(indeks) + 2)
        penyebut := (4*float64(indeks) + 1) * (4*float64(indeks) + 3)
        hasilPerkalian *= pembilang / penyebut
    }
    return hasilPerkalian
}

func main() {
    var iterasi int
    fmt.Print("Masukkan jumlah iterasi (K): ")
    fmt.Scan(&iterasi)
    hasilAkar2 := hitungAkar2(iterasi)
    fmt.Printf("Nilai hampiran akar 2 = %.10f\n", hasilAkar2)
}
```

Screenshot Output

```
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "
c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\unguided2B4.
go"
Masukkan jumlah iterasi (K): 100
Nilai hampiran akar 2 = 1.4133387072
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> █
```

Penjelasan

Program di atas merupakan program Go yang menghitung nilai akar 2 menggunakan metode iterasi. Pengguna diminta untuk memasukkan jumlah iterasi (K), dan program kemudian melakukan perhitungan berdasarkan rumus rekursif yang melibatkan pembilang $(4k + 2)^2$ dan penyebut $(4k + 1)(4k + 3)$ pada setiap iterasi. Hasil perkalian dari semua iterasi tersebut mendekati nilai akar 2, yang kemudian ditampilkan dengan presisi hingga 10 angka desimal. Semakin banyak iterasi, semakin akurat hasil hampiran akar 2 yang diperoleh.

5. Unguide 2C no. 1

Source Code

```
package main

import (
    "fmt"
)

//HANAH NUR AZIZAH 2311102312
func hitungBiayaKirim(beratTotalGram int) int {
    jumlahKg := beratTotalGram / 1000
    sisaGram := beratTotalGram % 1000
    biayaPerKg := 10000
    totalBiayaKg := jumlahKg * biayaPerKg
    biayaTambahan := 0

    if jumlahKg >= 10 {
        biayaTambahan = 0
    } else {
        if sisaGram >= 500 {
            biayaTambahan = sisaGram * 5
        } else {
            biayaTambahan = sisaGram * 15
        }
    }

    return totalBiayaKg + biayaTambahan
}

func main() {
    var beratTotalGram int

    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&beratTotalGram)
    jumlahKg := beratTotalGram / 1000
    sisaGram := beratTotalGram % 1000
    biayaPerKg := 10000 * jumlahKg
    biayaTambahan := 0

    if jumlahKg >= 10 {
```

```

        biayaTambahan = 0
    } else {
        if sisaGram >= 500 {
            biayaTambahan = sisaGram * 5
        } else {
            biayaTambahan = sisaGram * 15
        }
    }

    totalBiaya := biayaPerKg + biayaTambahan

    fmt.Printf("Detail berat: %d kg + %d gr\n", jumlahKg, sisaGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaPerKg, biayaTambahan)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "c:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\unguided2C1.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> 

```

Penjelasan

Program di atas adalah program Go yang menghitung biaya pengiriman berdasarkan berat parcel yang dimasukkan pengguna dalam satuan gram. Program memisahkan berat menjadi kilogram dan sisa gram, kemudian menghitung biaya pengiriman dengan tarif Rp. 10.000 per kilogram. Jika berat parcel lebih dari atau sama dengan 10 kg, tidak ada biaya tambahan. Namun, jika berat kurang dari 10 kg, sisa gram dikenakan biaya tambahan: Rp. 5 per gram untuk sisa 500 gram atau lebih, dan Rp. 15 per gram jika kurang dari 500 gram. Program menampilkan detail berat dalam kg dan gram serta total biaya pengiriman.

6. Unguided 2C no. 2

Source Code

```

package main
import "fmt"
//HANAH NUR AZIZAH 2311102312
func main() {
    var nam float64
    var nmk string

    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scan(&nam)

    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"
    }

    fmt.Printf("Nilai mata kuliah: %s\n", nmk)
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "c:\Users\hanah\OneDrive\
Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\unguided2C2.go"
Nilai akhir mata kuliah: 93,5
Nilai mata kuliah: A
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2>

```

Penjelasan

Program di atas merupakan program Go yang meminta pengguna untuk memasukkan nilai akhir mata kuliah dalam bentuk numerik, kemudian menentukan dan menampilkan nilai huruf yang sesuai berdasarkan rentang nilai yang telah ditetapkan. Jika nilai lebih dari 80, pengguna mendapatkan nilai "A", dan seterusnya hingga nilai "E" untuk nilai yang kurang dari atau sama dengan 40. Program ini menggunakan serangkaian kondisi if-else untuk mengklasifikasikan nilai numerik ke dalam kategori huruf dan menampilkan hasilnya.

7. Unguided 2C no. 3

Source Code

```

package main
import (
    "fmt"
)
//HANAH NUR AZIZAH 2311102312
func main() {
    var b int

    fmt.Print("Masukkan bilangan bulat: ")
    fmt.Scan(&b)

    if b <= 0 {
        fmt.Println("Bilangan harus lebih besar dari 0.")
    }
}

```

```

        return
    }

    fmt.Printf("Bilangan: %d\nFaktor: ", b)
    factors := []int{}

    for i := 1; i <= b; i++ {
        if b%i == 0 {
            factors = append(factors, i)
            if i == b {
                fmt.Print(i)
            } else {
                fmt.Print(i, ", ")
            }
        }
    }

    isPrime := len(factors) == 2

    fmt.Printf("\nPrima: %t\n", isPrime)
}

```

Screenshot Output

```

PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2> go run "c:\Users\hanah\OneDrive\
Dokumen\PRAKTIKUM ALPRO 2\MODUL 2\unguided2C3.go"
Masukkan bilangan bulat: 12
Bilangan: 12
Faktor: 1, 2, 3, 4, 6, 12
Prima: false
PS C:\Users\hanah\OneDrive\Dokumen\PRAKTIKUM ALPRO 2\MODUL 2>

```

Penjelasan

Program di atas merupakan program Go yang meminta pengguna memasukkan bilangan bulat positif, lalu menampilkan faktor-faktor dari bilangan tersebut. Jika bilangan kurang dari atau sama dengan 0, program akan memberikan pesan kesalahan dan berhenti. Setelah menemukan semua faktor, program juga memeriksa apakah bilangan tersebut adalah bilangan prima dengan mengecek jumlah faktor yang dimiliki (bilangan prima hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri). Hasil faktor dan status apakah bilangan tersebut prima ditampilkan kepada pengguna.

DAFTAR PUSTAKA

- [1] Modul 2: Review Struktur Kontrol