

LAPORAN PRAKTIKUM
Algoritma Pemrograman
MODUL II
REVIEW STRUKTUR KONTROL



Oleh:

Ilhan Sahal Mansiz

2311102029

IF-11-02

S1 TEKNIK INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

Struktur kontrol dalam bahasa pemrograman Go merupakan komponen penting yang menentukan alur eksekusi program berdasarkan kondisi tertentu atau perulangan. Struktur kontrol bersyarat seperti ``if-else`` dan ``switch`` memungkinkan program untuk memilih blok kode yang akan dijalankan berdasarkan hasil evaluasi kondisi. Selain itu, Go juga menyediakan struktur perulangan sederhana melalui ``for``, yang bisa digunakan untuk mengulang eksekusi blok kode hingga kondisi tertentu terpenuhi. Dengan hanya menggunakan ``for``, Go menyederhanakan pengelolaan perulangan tanpa memerlukan ``while`` atau ``do-while``.

Selain kontrol alur dasar, Go memiliki mekanisme untuk memindahkan atau menghentikan eksekusi blok kode dengan perintah seperti ``break``, ``continue``, dan ``goto``. Struktur ini digunakan untuk mengontrol aliran eksekusi secara lebih eksplisit. Misalnya, ``break`` digunakan untuk keluar dari perulangan, sedangkan ``continue`` memungkinkan program melewati iterasi tertentu dan melanjutkan ke iterasi berikutnya. Meskipun ``goto`` jarang digunakan karena dapat membingungkan alur program, Go tetap menyediakannya untuk kasus-kasus tertentu.

Selain itu, Go mendukung pemrograman konkuren melalui goroutine dan kanal. Goroutine memungkinkan eksekusi fungsi secara paralel, yang menjadikan Go unggul dalam menangani banyak tugas secara bersamaan. Kanal (``channels``) digunakan sebagai alat komunikasi antara goroutine, memastikan sinkronisasi yang aman. Dengan memanfaatkan kontrol seperti ``defer``, ``panic``, dan ``recover``, Go juga memungkinkan penanganan kondisi khusus atau kesalahan yang mungkin terjadi selama eksekusi program, sehingga membantu menjaga kestabilan alur kontrol program.

II. GUIDED

1. Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang niversity 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.Printf("Output awal = %s %s %s\n", satu, dua, tiga)
    temp = satu
    satu = dua
    dua = tiga
    tiga = temp
    fmt.Printf("Output akhir = %s %s %s\n", satu, dua, tiga)
}
```

Screenshot Program :

```
Masukan input string: a
Masukan input string: b
Masukan input string: c
Output awal = a b c
Output akhir = b c a
```

Deksripsi Program :

Program ini mengambil tiga input string dari pengguna dan mencetaknya sebagai "Output awal". Setelah itu, program menukar nilai dari variabel `satu`, `dua`, dan `tiga` menggunakan variabel sementara `temp`. Penukaran dilakukan dengan menggeser nilai: `satu` diisi dengan nilai `dua`, `dua` diisi dengan nilai `tiga`, dan `tiga` diisi dengan nilai awal `satu`. Setelah penukaran, program mencetak hasil baru sebagai "Output akhir" yang menunjukkan perubahan posisi nilai variabel.

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).

Source Code :

```
package main

import "fmt"

func isLeapYear(year int) bool {
    if year%400 == 0 {
        return true
    }
    if year%100 == 0 {
        return false
    }
    if year%4 == 0 {
        return true
    }
    return false
}

func main() {
    var year int
    fmt.Print("Masukkan tahun: ")
    fmt.Scan(&year)
    fmt.Println("Tahun :", year)
    if isLeapYear(year) {
        fmt.Println("Kabisat : true")
    } else {
        fmt.Println("Kabisat : false")
    }
}
```

Screenshot Program :

```
Masukkan tahun: 2016  
Tahun : 2016  
Kabisat : true
```

```
Masukkan tahun: 2018  
Tahun : 2018  
Kabisat : false
```

Deskripsi Program :

Program di atas adalah program dalam bahasa Go yang digunakan untuk memeriksa apakah suatu tahun merupakan tahun kabisat. Program ini meminta input berupa sebuah bilangan bulat yang mewakili tahun, kemudian menggunakan fungsi `isLeapYear` untuk menentukan apakah tahun tersebut kabisat atau bukan. Algoritma pengecekan mengikuti aturan: jika tahun habis dibagi 400, maka tahun kabisat; jika habis dibagi 100 tetapi tidak habis dibagi 400, maka bukan kabisat; jika habis dibagi 4 dan tidak habis dibagi 100, maka kabisat. Program kemudian mencetak hasil dalam format "Tahun : [input]" dan "Kabisat : [true/false]" tergantung hasil pengecekannya.

3. Buat program **Bola** yang menerima input jari-jari suatu bola (bilangan bulat). Tampilkan Volume dan Luas kulit bola. $volumebola = \frac{4}{3}\pi r^3$ dan $luasbola = 4\pi r^2$ ($\pi \approx 3.1415926535$).

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

Jejari = 5

Bola dengan jejari 5 memiliki volume 523.5988 dan luas kulit 314.1593

Source Code :

```

package main

import (
    "fmt"
    "math"
)

func main() {
    var r float64
    const pi = 3.1415926535

    fmt.Print("Jejari = ")
    fmt.Scan(&r)

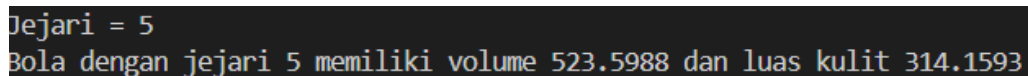
    volume := (4.0 / 3.0) * pi * math.Pow(r, 3)

    luas := 4 * pi * math.Pow(r, 2)

    fmt.Printf("Bola dengan jejari %.0f memiliki volume %.4f dan luas kulit %.4f\n", r, volume, luas)
}

```

Screenshot Program :



```

Jejari = 5
Bola dengan jejari 5 memiliki volume 523.5988 dan luas kulit 314.1593

```

Deskripsi Program :

Program ini menerima input berupa jari-jari bola dari pengguna, lalu menghitung volume dan luas permukaan bola berdasarkan nilai jari-jari tersebut. Program menggunakan nilai pi yang telah ditetapkan untuk perhitungan, kemudian menampilkan hasil berupa volume dan luas permukaan bola. Hasil perhitungan ditampilkan dengan format yang jelas dan menggunakan empat angka di belakang koma untuk ketepatan nilai volume dan luasnya.

4.

Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperatur 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: **50**

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: **50**

Derajat Reamur: 40

Derajat Fahrenheit: 122

Derajat Kelvin: 323

Source Code :

```
package main

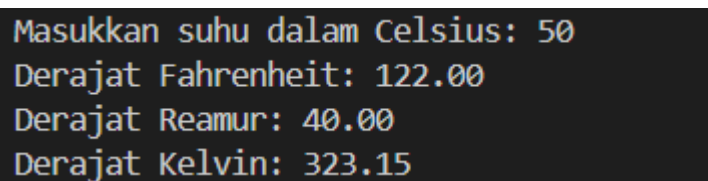
import "fmt"

func main() {
    var celsius float64
    fmt.Print("Masukkan suhu dalam Celsius: ")
    fmt.Scanf("%f", &celsius)

    fahrenheit := (celsius * 9 / 5) + 32
    reamur := celsius * 4 / 5
    kelvin := celsius + 273.15

    fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)
    fmt.Printf("Derajat Reamur: %.2f\n", reamur)
    fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)
}
```

Screenshot Program :



```
Masukkan suhu dalam Celsius: 50
Derajat Fahrenheit: 122.00
Derajat Reamur: 40.00
Derajat Kelvin: 323.15
```

Program di atas adalah program dalam bahasa Go yang meminta pengguna untuk memasukkan suhu dalam satuan Celsius, kemudian mengonversinya ke tiga satuan suhu lainnya, yaitu Fahrenheit, Reamur, dan Kelvin. Setelah pengguna memberikan input suhu dalam Celsius, program menghitung nilai Fahrenheit, Reamur, dan Kelvin, lalu menampilkan hasil konversinya dalam format yang rapi dengan dua angka di belakang koma. Program ini berguna untuk membantu memahami perbandingan antara berbagai satuan suhu dalam satu waktu.

5.

Tipe karakter sebenarnya hanya apa yang tampak dalam tampilan. Di dalamnya tersimpan dalam bentuk biner 8 bit (byte) atau 32 bit (rune) saja.

Buat program ASCII yang akan membaca 5 buah data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII)

Masukan terdiri dari dua baris. Baris pertama berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

Keluaran juga terdiri dari dua baris. Baris pertama berisi 5 buah representasi karakter dari data yang diberikan, yang berdampingan satu dengan lain, tanpa dipisahkan spasi. Baris kedua berisi 3 buah karakter (juga tidak dipisahkan oleh spasi).

| No. | Masukan | Keluaran |
|-----|--------------------------|--------------|
| 1 | 66 97 103 117 115 SNO | Bagus TOP |

Catatan: Gunakan `fmt.Scanf("%c", &var)` untuk pembacaan satu karakter dan `fmt.Printf("%c", var)` untuk penulisan satu karakter.

Source Code :

```

package main

import (
    "fmt"
)

func main() {
    // Deklarasi variabel untuk menampung input
    var a, b, c, d, e int
    var char1, char2, char3 byte

    // Membaca input untuk 5 data integer
    fmt.Scanf("%d %d %d %d %d", &a, &b, &c, &d, &e)

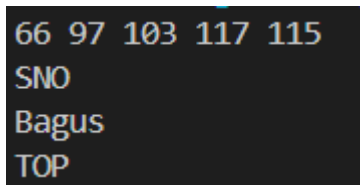
    // Membaca input untuk 3 data karakter, mengabaikan newline
    fmt.Scanf("\n%c%c%c", &char1, &char2, &char3)

    // Mencetak hasil konversi ASCII dari integer
    fmt.Printf("%c%c%c%c%c\n", a, b, c, d, e)

    // Mencetak karakter yang di=berikan
    fmt.Printf("%c%c%c\n", char1+1, char2+1, char3+1)
}

```

Screenshot Program :



```

66 97 103 117 115
SNO
Bagus
TOP

```

Deskripsi Program :

Program di atas adalah program dalam bahasa Go yang menerima input berupa lima bilangan bulat dan tiga karakter. Lima bilangan bulat tersebut diinterpretasikan sebagai kode ASCII dan dikonversi menjadi karakter yang sesuai, kemudian dicetak sebagai sebuah kata tanpa spasi. Setelah itu, program membaca tiga karakter tambahan dan mencetak karakter tersebut setelah masing-masing karakternya dinaikkan satu nilai dalam tabel ASCII (misalnya, jika input karakter adalah 'A',

output akan menjadi 'B'). Program ini berguna untuk mengonversi angka menjadi karakter ASCII dan memodifikasi karakter input dengan menambah nilai ASCII-nya.

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 Telkom University urutan warna lainnya.

Source Code :

```
package main

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

func cekPercobaan(percobaan [][]string) bool {
    susunanBenar := []string{"merah", "kuning", "hijau", "ungu"}

    for _, p := range percobaan {
        for i, warna := range p {
            if warna != susunanBenar[i] {
                return false
            }
        }
    }
    return true
}

func main() {
    var percobaan [][]string
    scanner := bufio.NewScanner(os.Stdin)
    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        scanner.Scan()
        input := scanner.Text()
        warna := strings.Split(input, " ")
        percobaan = append(percobaan, warna)
    }

    hasil := cekPercobaan(percobaan)
    fmt.Println("BERHASIL:", hasil)
}
```

Screenshot Program :

```
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:\Users\ACER\Downloads\Semester 3\Labrak Alpro smstr3\Modul2> go
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
Percobaan 5: merah kuning hijau ungu
BERHASIL: false
PS C:\Users\ACER\Downloads\Semester 3\Labrak Alpro smstr3\Modul2> |
```

Deskripsi Program :

Program ini adalah aplikasi berbasis terminal yang memverifikasi urutan warna cairan dalam percobaan kimia. Program meminta pengguna untuk memasukkan urutan warna pada empat tabung reaksi selama 5 percobaan berturut-turut. Setiap percobaan dicatat sebagai input berupa string warna yang dipisahkan oleh spasi (misalnya "merah kuning hijau ungu"). Setelah semua input dimasukkan, program memeriksa apakah urutan warna untuk semua percobaan sesuai dengan urutan yang benar yaitu "merah", "kuning", "hijau", dan "ungu". Jika semua percobaan mengikuti urutan tersebut, program akan mengeluarkan hasil "BERHASIL: true", jika tidak maka "BERHASIL: false".

2. Suatu pita (string) berisi kumpulan nama-nama bunga yang dipisahkan oleh spasi dan '-', contoh pita diilustrasikan seperti berikut ini. Pita: mawar - melati - tulip - teratai - kamboja - anggrek. Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita. (Petunjuk: gunakan operasi penggabungan string dengan operator "+"). Tampilkan isi pita setelah proses input selesai.

Source Code :


```
package main

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

func main() {
    var n int
    var pita string
    scanner := bufio.NewScanner(os.Stdin)

    fmt.Print("N : ")
    fmt.Scan(&n)

    scanner.Scan()

    for i := 1; i <= n; i++ {
        fmt.Print("Bunga ", i, " : ")
        scanner.Scan()
        bunga := scanner.Text()

        if i == 1 {
            pita = bunga
        } else {
            pita += " - " + bunga
        }
    }

    fmt.Println("Pita :", pita)
}
```

Screenshot Program :

```
N : 0
Pita :
PS C:\Users\ACER\Downloads\Semester
N : 3
Bunga 1 : Kertas
Bunga 2 : Mawar
Bunga 3 : Tulip
Pita : Kertas - Mawar - Tulip
```

Deskripsi Program :

Program ini adalah aplikasi berbasis terminal yang meminta input dari pengguna untuk mengisi nama bunga sejumlah N kali, di mana N merupakan bilangan bulat yang diinputkan terlebih dahulu oleh pengguna. Setelah menerima jumlah bunga, program kemudian meminta pengguna untuk memasukkan nama bunga satu per satu. Setiap nama bunga yang diinputkan akan digabungkan menjadi satu string yang dipisahkan oleh tanda " - ". Setelah semua bunga dimasukkan, program akan menampilkan hasil akhir berupa pita string yang berisi daftar nama bunga yang telah dimasukkan pengguna.

3. Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri-kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong sisi tidak lebih dari 9 kg. Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9 kg atau lebih. 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.

Source Code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    for {
        var beratKiri, beratKanan float64

        fmt.Print("Masukkan berat belanja di kedua kantong (kiri kanan): ")
        fmt.Scan(&beratKiri, &beratKanan)

        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Salah satu kantong berisi berat negatif. Program dihentikan.")
            break
        }

        if beratKiri+beratKanan > 150 {
            fmt.Println("Total berat kedua kantong melebihi 150 kg. Program dihentikan.")
            break
        }

        selisih := math.Abs(beratKiri - beratKanan)

        if selisih >= 9 {
            fmt.Println("True, selisih berat kedua kantong lebih dari atau sama dengan 9 kg.")
        } else {
            fmt.Println("False, selisih berat kedua kantong kurang dari 9 kg.")
        }

        fmt.Println("Proses selesai.")
    }
}
```

| |
|--|
| |
|--|

Screenshot Program :

```
Masukkan berat belanjaan di kedua kantong (kiri kanan): 5 10
False, selisih berat kedua kantong kurang dari 9 kg.
Proses selesai.
Masukkan berat belanjaan di kedua kantong (kiri kanan): 55.6 70.2
True, selisih berat kedua kantong lebih dari atau sama dengan 9 kg.
Proses selesai.
Masukkan berat belanjaan di kedua kantong (kiri kanan): 72.3 66.9
False, selisih berat kedua kantong kurang dari 9 kg.
Proses selesai.
Masukkan berat belanjaan di kedua kantong (kiri kanan): 59.5 98.7
Total berat kedua kantong melebihi 150 kg. Program dihentikan.
```

Deskripsi Program :

Program di atas adalah aplikasi berbasis Go yang digunakan untuk membantu Pak Andi menghitung apakah selisih berat antara dua kantong belanjaan di sepeda motornya cukup besar untuk menyebabkan motor oleng. Program akan terus meminta input berat untuk kedua kantong sampai salah satu kantong memiliki berat negatif atau total berat kedua kantong melebihi 150 kg, yang menyebabkan program berhenti. Selisih berat antara kantong kiri dan kanan dihitung, dan jika selisih tersebut 9 kg atau lebih, program akan menampilkan "True" (motor oleng); jika kurang dari 9 kg, akan menampilkan "False".

4.

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

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):

$$\sqrt{2} = \prod_{k=0}^{\infty} \frac{(4k+2)^2}{(4k+1)(4k+3)}$$

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):

Source Code :

```

package main

import (
    "fmt"
    "math"
)

func main(){
    // var k int

    // fmt.Print("Nilai K = ")
    // fmt.Scan(&k)

    // result := (math.Pow(float64(4*k+2),2) /
float64(4*k+1)*float64(4*k+3))
    // fmt.Printf("Nilai f(k) = %10f\n", result)

    // Modifikasi
    var k int
    fmt.Print("Nilai K = ")
    fmt.Scan(&k)
    result := 1.0
    for i := 0; i < k; i++ {
        rumus := (math.Pow(float64(4*i+2), 2)) / (float64(4*i+1) *
float64(4*i+3))

        if result * rumus > 0 {
            result *= rumus
        }
    }
    fmt.Printf("Nilai akar 2 = %.10f\n", result)
}

```

Screenshot Program :

```
Nilai K = 10
Nilai akar 2 = 1.4054086752
PS C:\Users\ACER\Downloads\Semes
Nilai K = 100
Nilai akar 2 = 1.4133299615
PS C:\Users\ACER\Downloads\Semes
Nilai K = 1000
Nilai akar 2 = 1.4141251768
```

Deskripsi Program :

Program ini meminta pengguna untuk memasukkan sebuah angka k, kemudian melakukan perhitungan berulang sebanyak k kali. Di setiap perulangan, program menghitung suatu nilai, dan jika hasilnya positif, nilai tersebut akan dikalikan dengan hasil perhitungan sebelumnya. Setelah semua perulangan selesai, program menampilkan hasil akhirnya dalam bentuk angka desimal yang sangat presisi dengan 10 angka di belakang koma.

5. PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program BiayaPos untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut! Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

Source Code :


```

package main

import (
    "fmt"
)

func hitungBiayaPengiriman(beratGram int) (int, int, int) {
    beratKg := beratGram / 1000
    sisaGram := beratGram % 1000

    biayaPerKg := 10000
    biayaTambahan := 0

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

    totalBiaya := (beratKg * biayaPerKg) + biayaTambahan
    return beratKg, sisaGram, totalBiaya
}

func main() {
    var beratParcel int
    fmt.Print("Masukkan berat parcel (dalam gram): ")
    fmt.Scan(&beratParcel)

    beratKg, sisaGram, biaya := hitungBiayaPengiriman(beratParcel)
    fmt.Printf("Berat parcel (gram): %d\n", beratParcel)
    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)
}

```

Screenshot Program :

```
Masukkan berat parcel (dalam gram): 8500
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\ACER\Downloads\Semester 3\Laprak
Masukkan berat parcel (dalam gram): 9250
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Users\ACER\Downloads\Semester 3\Laprak
Masukkan berat parcel (dalam gram): 11750
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
```

Deskripsi Program :

Program ini menghitung biaya pengiriman parcel berdasarkan berat yang dimasukkan pengguna dalam satuan gram. Setelah pengguna memasukkan berat, program akan mengonversi berat tersebut ke dalam kilogram dan gram, lalu menghitung biaya pengiriman berdasarkan tarif Rp. 10.000 per kilogram. Jika berat tidak lebih dari 10 kg, program juga menghitung biaya tambahan berdasarkan sisa gram dengan tarif yang berbeda, yaitu Rp. 5.000 jika sisa gram 500 atau lebih, dan Rp. 15.000 jika kurang dari 500 gram. Program kemudian menampilkan detail berat, rincian biaya, dan total biaya pengiriman.

6. Jawablah pertanyaan-pertanyaan berikut:
- a. Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
 - b. Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
 - C. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Source Code :

```
package main

import (
    "fmt"
)

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&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.Println("Nilai mata kuliah:", nmk)
}
```

Soal a.

```
Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: A
```

Jika nam diberikan sebagai 80.1, keluaran dari program tersebut akan menjadi A karena nilai tersebut lebih besar dari 80 dan sesuai dengan kondisi `if nam > 80`.

eksekusi program tersebut sesuai dengan spesifikasi soal untuk kasus $nam = 80.1$. Nilai 80.1 seharusnya diberikan huruf A karena nilainya lebih besar dari 80.

Soal b.

1. Penggunaan banyak pernyataan if tanpa else :

- Di bagian pertama, tidak ada pernyataan else if yang membatasi agar hanya satu kondisi yang dieksekusi. Akibatnya, jika nilai nam berada di lebih dari satu rentang yang sesuai, beberapa kondisi bisa dieksekusi secara tidak terduga.
- Sebagai contoh, jika nam adalah 75, program bisa saja mengevaluasi lebih dari satu kondisi (misalnya, $if\ nam > 65$ juga bisa benar setelah $if\ nam > 72.5$).

2. Alur Program seharusnya menggunakan if else :

- Karena kondisi penilaian ini harus saling eksklusif (hanya satu rentang nilai yang berlaku untuk setiap nilai nam), pernyataan if harus diubah menjadi else if setelah kondisi pertama. Ini memastikan bahwa hanya satu blok kode yang dieksekusi.

Soal c.

Source Code :

```
package main

import (
    "fmt"
)

func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)

    // Struktur pengkondisian diperbaiki untuk menggunakan else if
    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.Println("Nilai mata kuliah:", nmk)
}
```

Screenshot Program :

```
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Users\ACER\Downloads\Semester
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Users\ACER\Downloads\Semester
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
```

Penjelasan :

- Input 93.5: Program akan mengevaluasi kondisi pertama $\text{nam} > 80$, sehingga akan memberikan keluaran A.
 - Input 70.6: Program akan mengevaluasi beberapa kondisi, dan kondisi $\text{nam} > 65$ yang akan terpenuhi, sehingga keluaran adalah B.
 - Input 49.5: Program akan mengevaluasi kondisi $\text{nam} > 40$, yang menghasilkan keluaran D.
7. 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! Bilangan bulat $b > 0$ merupakan bilangan prima p jika dan hanya jika memiliki persis dua faktor bilangan saja, yaitu 1 dan dirinya sendiri. Lanjutkan program sebelumnya. Setelah menerima masukan sebuah bilangan bulat $b > 0$. Program tersebut mencari dan menampilkan semua faktor bilangan tersebut. Kemudian, program menentukan apakah b merupakan bilangan prima.

Source Code :

```
package main

import "fmt"

func isPrime(number int) bool {
    if number < 2 {
        return false
    }
    for i := 2; i*i <= number; i++ {
        if number%i == 0 {
            return false
        }
    }
    return true
}

func main() {
    var number int
    fmt.Print("Masukkan Bilangan: ")
    fmt.Scanln(&number)

    fmt.Printf("Faktor dari %d adalah: ", number)
    for i := 1; i <= number; i++ {
        if number%i == 0 {
            fmt.Printf("%d ", i)
        }
    }
    fmt.Println()

    primeStatus := isPrime(number)
    fmt.Printf("Prima: %t\n", primeStatus)
}
```


Screenshot Program :

```
Masukkan Bilangan: 7
Faktor dari 7 adalah: 1 7
Prima: true
PS C:\Users\ACER\Downloads\Semester 1>
Masukkan Bilangan: 12
Faktor dari 12 adalah: 1 2 3 4 6 12
Prima: false
```

Deskripsi Program :

Program ini menerima input sebuah bilangan dari pengguna, kemudian menampilkan faktor-faktor dari bilangan tersebut dan memeriksa apakah bilangan tersebut merupakan bilangan prima. Proses ini dilakukan dengan dua langkah: pertama, program menghitung dan mencetak semua faktor dari bilangan; kedua, program memeriksa apakah bilangan tersebut hanya memiliki dua faktor (1 dan dirinya sendiri) untuk menentukan apakah itu bilangan prima. Hasil pemeriksaan bilangan prima akan ditampilkan dalam format boolean (`True` jika prima, `False` jika bukan).