

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



Oleh:

Alden Audy Akbar

2311102309

IF-11-07

S1 TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

1. Struktur if

```
package main

import "fmt"

func main() {
    x := 10

    if x > 5 {
        fmt.Println("x lebih besar dari 5")
    } else {
        fmt.Println("x kurang dari atau sama dengan 5")
    }
}
```

2. Struktur for

```
package main

import "fmt"

func main() {
    // Contoh perulangan klasik
    for i := 0; i < 5; i++ {
        fmt.Println(i)
    }

    // Perulangan tanpa kondisi, berjalan terus-menerus
    count := 0
    for {
        fmt.Println("Infinite loop", count)
        count++
        if count == 3 {
            break
        }
    }

    // Looping array/slice dengan range
    nums := []int{2, 4, 6}
    for index, value := range nums {
```

```
    fmt.Printf("Index: %d, Value: %d\n", index, value)
  }
}
```

3. Struktur switch

```
package main

import "fmt"

func main() {
    day := "Monday"

    switch day {
    case "Monday":
        fmt.Println("Hari ini adalah Senin")
    case "Tuesday":
        fmt.Println("Hari ini adalah Selasa")
    default:
        fmt.Println("Hari ini bukan Senin atau Selasa")
    }

    // Switch tanpa kondisi, bisa digunakan seperti if-else
    num := 10
    switch {
    case num < 5:
        fmt.Println("Angka kurang dari 5")
    case num > 5:
        fmt.Println("Angka lebih dari 5")
    default:
        fmt.Println("Angka sama dengan 5")
    }
}
```

4. Struktur defer, panic, dan recover

- defer: Menunda eksekusi fungsi sampai fungsi sekitarnya selesai.
- panic: Menghentikan eksekusi normal program dan mulai memanggil defer-ed fungsi.
- recover: Memungkinkan pemulihan dari situasi panic.

```
package main

import "fmt"

func main() {
    defer fmt.Println("Ini akan dieksekusi terakhir")
    fmt.Println("Ini akan dieksekusi pertama")

    deferExample()
}

func deferExample() {
    defer fmt.Println("Selesai deferExample")
    fmt.Println("Memulai deferExample")

    // panic dan recover
    defer func() {
        if r := recover(); r != nil {
            fmt.Println("Recovered dari panic:", r)
        }
    }()

    panic("Terjadi panic!")
}
```

II. GUIDED

2.A

1. Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sebanyak yang diminta program.

```
package main

import "fmt"

func main() {
    var satu, dua, tiga string

    fmt.Print("Masukkan input string pertama: ")
    fmt.Scanln(&satu)

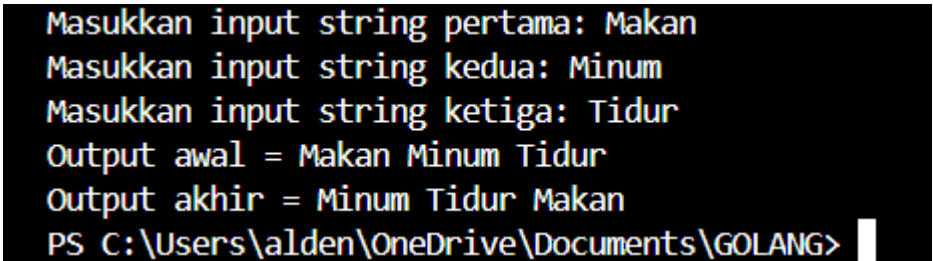
    fmt.Print("Masukkan input string kedua: ")
    fmt.Scanln(&dua)

    fmt.Print("Masukkan input string ketiga: ")
    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)
}
```



```
Masukkan input string pertama: Makan
Masukkan input string kedua: Minum
Masukkan input string ketiga: Tidur
Output awal = Makan Minum Tidur
Output akhir = Minum Tidur Makan
PS C:\Users\alden\OneDrive\Documents\GOLANG>
```

Penjelasan: Program ini menukar nilai dari tiga variabel string. Proses penukaran ini dilakukan dengan memanfaatkan variabel sementara temp

untuk menyimpan nilai sementara salah satu variabel sebelum dilakukan penukaran.

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).

```
package main

//2311102309
import "fmt"

func main() {
    var tahun int

    fmt.Print("Tahun: ")
    fmt.Scanln(&tahun)

    if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {
        fmt.Println("Kabisat: true")
    } else {
        fmt.Println("Kabisat: false")
    }
}
```

```
Tahun: 2077
Kabisat: false
PS C:\Users\alden\OneDrive\Documents\GOLANG>
Tahun: 2016
Kabisat: true
PS C:\Users\alden\OneDrive\Documents\GOLANG>
```

Penjelasan: Program ini bekerja dengan cara:

Meminta input tahun: Pengguna diminta untuk memasukkan tahun yang ingin dicek.

Memeriksa kondisi tahun kabisat: Program akan mengecek apakah tahun yang dimasukkan memenuhi syarat tahun kabisat, yaitu:

Habis dibagi 400 (misal, 2000, 2400)

Atau habis dibagi 4 tetapi tidak habis dibagi 100 (misal, 2024, 2028)
Memberikan output: Program akan menampilkan hasil apakah tahun tersebut adalah tahun kabisat atau bukan. Jika memenuhi salah satu syarat di atas, maka akan ditampilkan "Kabisat: true", jika tidak, maka "Kabisat: false".

3. Buat program Bola yang menerima input jari-jari suatu bola (bilangan bulat). Tampilkan Volume dan Luas kulit bola.

```
package main

// 2311102309
import (
    "fmt"

    "math"
)

func main() {
    var jariJari float64

    fmt.Print("Jari-jari: ")
    fmt.Scanln(&jariJari)

    volume := (4.0 / 3.0) * math.Pi * math.Pow(jariJari, 3)
    luasKulit := 4 * math.Pi * math.Pow(jariJari, 2)

    fmt.Printf("Bola dengan jari-jari %.2f memiliki Volume %.3f dan\nluas Kulit %.3f\n", jariJari, volume, luasKulit)
}
```

```
Jari-jari: 10
Bola dengan jari-jari 10.00 memiliki volume 4188.790 dan luas Kulit 1256.637
PS C:\Users\alden\OneDrive\Documents\GOLANG>
```

Penjelasan: Program meminta pengguna untuk memasukkan nilai jari-jari bola.

Nilai jari-jari yang dimasukkan disimpan dalam variabel jari Jari.

Program menghitung volume dan luas permukaan bola menggunakan rumus yang sudah ditentukan.

Hasil perhitungan kemudian ditampilkan ke layar

4. Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam Fahrenheit. Lanjutkan program di atas, sehingga temperatur dinyatakan juga dalam derajat Reamur dan Kelvin.

```
package main

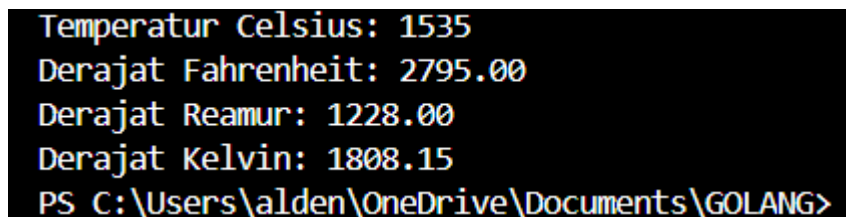
//2311102309
import "fmt"

func main() {
    var celsius float64

    fmt.Print("Temperatur Celsius: ")
    fmt.Scanln(&celsius)

    fahrenheit := (celsius * 9.0 / 5.0) + 32
    reamur := celsius * 4.0 / 5.0
    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)
}
```



```
Temperatur Celsius: 1535
Derajat Fahrenheit: 2795.00
Derajat Reamur: 1228.00
Derajat Kelvin: 1808.15
PS C:\Users\alden\OneDrive\Documents\GOLANG>
```

Penjelasan: Program meminta pengguna untuk memasukkan suhu dalam derajat Celsius.

Nilai suhu Celsius yang dimasukkan disimpan dalam variabel celsius.

Program menghitung suhu dalam Fahrenheit, Reamur, dan Kelvin menggunakan rumus konversi yang sesuai.

Hasil perhitungan kemudian ditampilkan ke layar.

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)

```
package main

// 2311102309
import (
    "bufio"
    "fmt"
    "os"
)

func main() {
    var angka [5]int
    var karakter [3]byte
    reader := bufio.NewReader(os.Stdin)

    fmt.Println("Masukkan 5 angka (32-127):")
    for i := 0; i < 5; i++ {
        fmt.Scanf("%d", &angka[i])
    }

    reader.ReadString('\n')

    fmt.Println("Masukkan 3 karakter:")
    for i := 0; i < 3; i++ {
        char, _ := reader.ReadByte()
        karakter[i] = char
    }
    fmt.Print("Karakter dari angka: ")
    for _, v := range angka {
        if v >= 32 && v <= 127 {
            fmt.Printf("%c", v)
        }
    }
}
```

```

    } else {
        fmt.Print("?")
    }
}
fmt.Println()
fmt.Print("Karakter setelahnya: ")
for _, v := range karakter {
    fmt.Printf("%c", v+1)
}
fmt.Println()
}

```

```

Masukkan 5 angka (32-127):
74 69 76 69 75
Masukkan 3 karakter:
KNK
Karakter dari angka: JELEK
Karakter setelahnya: LOL
PS C:\Users\alden\OneDrive\Documents\GOLANG>

```

Penjelasan:

Meminta input: Program meminta pengguna untuk memasukkan 5 angka dan 3 karakter.

Membaca input: Input yang dimasukkan pengguna disimpan dalam array.

Konversi angka ke karakter: Setiap angka yang dimasukkan dianggap sebagai kode ASCII dan langsung dikonversi menjadi karakter. Jika angka di luar rentang kode ASCII yang dapat dicetak (32-127), akan dicetak tanda tanya.

Konversi karakter ke karakter selanjutnya: Setiap karakter yang dimasukkan akan dinaikkan nilainya satu tingkat dalam tabel ASCII, sehingga menghasilkan karakter berikutnya.

Mencetak hasil: Hasil konversi dari angka ke karakter dan karakter ke karakter selanjutnya akan ditampilkan di layar.

III. UNGUIDED

2B

1. 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.

```
package main

// 2311102309

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

func cekPercobaan(warnaPercobaan [][]string) bool {
    warnaldeal := []string{"merah", "kuning", "hijau", "ungu"}
    for _, percobaan := range warnaPercobaan {
        if !isSame(percobaan, warnaldeal) {
            return false
        }
    }
    return true
}

func isSame(a, b []string) bool {
```

```
    if len(a) != len(b) {
        return false
    }
    for i := 0; i < len(a); i++ {
        if a[i] != b[i] {
            return false
        }
    }
    return true
}

func main() {
    var warnaPercobaan [][]string
    scanner := bufio.NewScanner(os.Stdin)

    fmt.Println("Masukkan 5 percobaan (masukkan 4 warna per percobaan):")
    for i := 0; i < 5; i++ {
        fmt.Printf("Percobaan %d: ", i+1)
        scanner.Scan()
        line := scanner.Text()

        percobaan := strings.Fields(line)

        if len(percobaan) != 4 {
            fmt.Println("Input tidak valid, masukkan tepat 4 warna.")
            i--
        }
    }
}
```

```

        continue
    }

    warnaPercobaan = append(warnaPercobaan, percobaan)
}

if cekPercobaan(warnaPercobaan) {
    fmt.Println("True")
} else {
    fmt.Println("False.")
}
}

```

```

Masukkan 5 percobaan (masukkan 4 warna per percobaan):
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
True
PS C:\Users\alden\OneDrive\Documents\GOLANG>

```

Penjelasan: Program meminta pengguna untuk menginput 5 percobaan. Setiap percobaan harus terdiri dari 4 warna yang dimasukkan dalam satu baris, dipisahkan oleh spasi. Input ini dibaca menggunakan `bufio.Scanner`, yang menangkap setiap baris input. Jika jumlah warna dalam percobaan tidak tepat 4, pengguna akan diminta mengulangi input untuk percobaan tersebut. Setiap percobaan dibandingkan dengan array warna ideal ["merah", "kuning", "hijau", "ungu"] menggunakan fungsi `cekPercobaan`. Fungsi `isSame` memeriksa apakah array percobaan sama dengan array warna ideal. Jika ada satu percobaan yang tidak sesuai, hasilnya adalah `False`. Jika semua percobaan sesuai dengan warna ideal, program mencetak `True`. Jika ada satu percobaan yang tidak sesuai, program mencetak `False`. `cekPercobaan(warnaPercobaan [][]string)`: Memeriksa setiap

percobaan apakah sesuai dengan warna ideal. `isSame(a, b []string)`:

Membandingkan dua slice string dan memastikan semua elemennya sama.

Pengguna diminta memasukkan 5 baris warna. Jika semua baris sesuai dengan warna ideal, hasilnya adalah True, jika tidak, hasilnya False.

2. 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.

```
package main
// 2311102309
import (
    "fmt"
    "strings"
)

func main() {
    var bunga string
    var pita string
    for {
        fmt.Print("Masukkan nama bunga: ")
        fmt.Scanln(&bunga)
        if bunga == "SELESAI" {
            break
        }

        pita += bunga + " "
    }
    fmt.Println("Pita:", pita)
    fmt.Println("Jumlah bunga:", len(strings.Split(pita, " ")))
}
```

```
Masukkan nama bunga: Kertas
Masukkan nama bunga: Mawar
Masukkan nama bunga: Tulip
Masukkan nama bunga: SELESAI
Pita: Kertas Mawar Tulip
Jumlah bunga: 4
PS C:\Users\alden\OneDrive\Documents\GOLANG>
```

Penjelasan: Program meminta pengguna untuk menginput nama bunga satu per satu. Proses input akan terus berlangsung hingga pengguna mengetikkan "SELESAI". Setiap nama bunga yang dimasukkan akan ditambahkan ke dalam variabel pita, dipisahkan oleh spasi. Setelah keluar dari loop, program mencetak pita (gabungan semua nama bunga) dan menghitung jumlah bunga yang dimasukkan menggunakan `strings.Split()` yang memisahkan kata berdasarkan spasi. `fmt.Scanln(&bunga)`: Menerima input bunga dari pengguna. `strings.Split(pita, " ")`: Memisahkan string pita menjadi slice kata-kata untuk menghitung jumlah bunga. Jika pengguna memasukkan "Mawar", "Melati", "Anggrek", lalu "SELESAI", hasilnya akan mencetak pita dan jumlah bunga.

3. 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.

```
package main

//2311102309
import (
    "fmt"
    "math"
)

func isSepedaMotorOleng(berat1, berat2 float64) bool {
    return math.Abs(berat1-berat2) >= 9
}

func main() {
    var berat1, berat2 float64
```

```

fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
fmt.Scan(&berat1, &berat2)

if berat1 < 0 || berat2 < 0 {
    fmt.Println("Berat tidak boleh negatif.")
    return
}

if berat1+berat2 > 150 {
    fmt.Println("Total berat melebihi 150 kg. Proses selesai.")
    return
}
if isSepedaMotorOlang(berat1, berat2) {
    fmt.Println("Sepeda motor Pak Andi akan oleng: true")
} else {
    fmt.Println("Sepeda motor Pak Andi akan oleng: false")
}
}

```

Masukkan berat belanjaan di kedua kantong: 59.5 98.7
Total berat melebihi 150 kg. Proses selesai.

Penjelasan: Program meminta pengguna memasukkan berat belanjaan untuk dua kantong.

- Jika ada berat negatif, program menampilkan pesan bahwa berat tidak boleh negatif dan menghentikan proses.

- Jika total berat lebih dari 150 kg, program menampilkan pesan bahwa berat melebihi batas dan menghentikan proses. Cek Sepeda Motor Olang: Program menghitung selisih berat antara dua kantong. Jika selisihnya ≥ 9 kg, sepeda motor akan oleng dan mencetak hasil "true". Jika tidak, mencetak "false". isSepedaMotorOlang: Mengembalikan true jika selisih berat dua kantong ≥ 9 kg, menggunakan fungsi math.Abs untuk menghitung selisih absolut.

4. Buatlah sebuah program yang menerima input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai $f(K)$ sesuai persamaan di atas.

```
package main

// 2311102309
import (
    "fmt"
)

func hitungAkar2(K int) float64 {
    var akar2 float64 = 1.0
    for k := 0; k <= K; k++ {
        akar2 *= float64((4*k+2)*(4*k+2)) / float64((4*k+1)*(4*k+3))
    }
    return akar2
}

func main() {
    var K int
    var akar2 float64
    fmt.Print("Masukkan nilai K: ")
    fmt.Scanln(&K)
    akar2 = hitungAkar2(K)

    fmt.Printf("Nilai K = %d\n", K)
    fmt.Printf("Nilai akar 2 = %.10f\n", akar2)
}
```

```
Masukkan nilai K: 10
Nilai K = 10
Nilai akar 2 = 1.4062058441
PS C:\Users\alden\OneDrive\Documents\GOLANG> |
```

Penjelasan: Program mendefinisikan fungsi `hitungAkar2` yang memiliki tugas untuk menghitung hampiran nilai akar kuadrat dari 2. Fungsi ini menerima satu parameter, yaitu `K`, yang merupakan bilangan bulat. Nilai `K` akan menentukan tingkat akurasi dari hasil perhitungan. Di dalam fungsi ini, terdapat sebuah perulangan untuk mendapatkan hasil yang semakin akurat. Program membuat sebuah array yang bernama `contohInput`. Array ini berisi beberapa nilai bilangan bulat yang akan digunakan sebagai input untuk fungsi `hitungAkar2`. Nilai-nilai dalam array `contohInput` mewakili berbagai nilai `K` yang ingin kita uji untuk melihat bagaimana hasil perhitungan akar kuadratnya. Iterasi Melalui `contohInput`: Program akan melakukan perulangan untuk setiap nilai `K` yang ada dalam array `contohInput`. Pada setiap iterasi, nilai `K` yang sedang diproses akan diambil dari array tersebut.

Memanggil Fungsi `hitungAkar2`: Untuk setiap nilai `K` yang diambil dari array `contohInput`, program akan memanggil fungsi `hitungAkar2` dengan nilai `K` tersebut sebagai argumen. Fungsi `hitungAkar2` kemudian akan menghitung hampiran nilai akar kuadrat dari 2 berdasarkan nilai `K` yang diberikan. Setelah mendapatkan hasil perhitungan dari fungsi `hitungAkar2`, program akan mencetak hasil tersebut ke layar. Baris pertama menunjukkan nilai `K` yang sedang diproses. Baris kedua menunjukkan hasil perhitungan akar kuadrat, yaitu nilai hampiran akar kuadrat dari 2 untuk nilai `K` yang bersangkutan.

C2

1. PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program `BlayaPos` untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut!

```
package main

//2311102309
import "fmt"

func main() {
    var beratParsel, beratKg, beratGram int
    var biayaKg, biayaGram, totalBiaya int
    fmt.Print("Berat parsel (gram): ")
    fmt.Scanln(&beratParsel)

    beratKg = beratParsel / 1000
```

```

beratGram = beratParsel % 1000

biayaKg = beratKg * 10000
if beratGram >= 500 {
    biayaGram = beratGram * 5
} else {
    biayaGram = beratGram * 15
}

if beratKg > 10 {
    biayaGram = 0
}
totalBiaya = biayaKg + biayaGram

fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, beratGram)
fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKg, biayaGram)
fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}

```

```

Berat parsel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Users\alden\OneDrive\Documents\GOLANG>

```

Penjelasan: Program meminta input dari pengguna menggunakan `fmt.Scanln(&beratParsel)`. Menghitung Berat dalam Kilogram dan Gram:

Berat parsel dibagi 1000 untuk mendapatkan berat dalam kilogram (`beratKg = beratParsel / 1000`). Sisa hasil bagi dengan operasi modulus (%) digunakan untuk menghitung sisa berat dalam gram (`beratGram = beratParsel % 1000`). Biaya pengiriman dihitung berdasarkan berat dalam kilogram dan gram:

Biaya untuk kilogram: Setiap kilogram dikenakan biaya Rp 10.000 (`biayaKg = beratKg * 10000`). Jika berat gram ≥ 500 , setiap gram dikenakan biaya Rp 5 (`biayaGram = beratGram * 5`). Jika berat gram < 500 , setiap gram dikenakan biaya Rp 15 (`biayaGram = beratGram * 15`). Jika berat parsel lebih dari 10 kg, biaya untuk gram diabaikan (`biayaGram = 0`). Program menampilkan rincian berat dalam kilogram dan gram (`fmt.Printf("Detail berat: %d kg + %d gr\n", beratKg, beratGram)`).

Program juga menampilkan rincian biaya untuk kilogram dan gram, serta total biaya pengiriman (fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)).

2. 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! ujliah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Jawab:

A. Jika diberikan nilai nam sebesar 80.1, program akan mengalami masalah karena struktur if nya tidak lengkap. Program akan terus mengevaluasi kondisi if tanpa mencapai hasil . Beberapa kondisi if memiliki rentang nilai yang tumpang tindih, sehingga sulit untuk menentukan kondisi mana yang akan dipenuhi. Nilai nmk tidak selalu ditugaskan dengan benar pada setiap kondisi if. Fungsi fmt.Scanln() digunakan untuk membaca input dari pengguna, namun dalam kode ini digunakan secara tidak tepat.

B. Penggunaan if bersarang yang terlalu banyak membuat kode sulit dibaca. Beberapa kondisi if tidak mencakup semua kemungkinan nilai. Nilai nmk tidak selalu ditugaskan dengan benar pada setiap kondisi if. Fungsi fmt.Scanln() digunakan untuk membaca input dari pengguna, namun dalam kode ini digunakan secara tidak tepat.

C.

```
package main

//2311102309
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)
}

```

```

Nilai akhir mata kuliah: 93.5;
Nilai mata kuliah: A
PS C:\Users\alden\OneDrive\Documents\GOLANG>
Nilai akhir mata kuliah: 70.6;
Nilai mata kuliah: B
PS C:\Users\alden\OneDrive\Documents\GOLANG>
Nilai akhir mata kuliah: 49.5;
Nilai mata kuliah: D
PS C:\Users\alden\OneDrive\Documents\GOLANG>

```

Penjelasan: Digunakan struktur if-else if yang lebih efisien untuk memeriksa kondisi secara berurutan. Setiap kondisi if memeriksa rentang nilai yang spesifik dan tidak tumpang tindih dengan kondisi lainnya. Nilai nmk hanya ditugaskan sekali dalam setiap blok if atau else if. Fungsi fmt.Scanln() digunakan untuk membaca input nilai akhir mata kuliah dari pengguna dan menyimpannya ke dalam variabel nam.

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

```
package main

//2311102309
import "fmt"

func main() {
    var bilangan int

    fmt.Print("Bilangan: ")
    fmt.Scanln(&bilangan)
    fmt.Print("Faktor: ")
    var faktor []int
    for i := 1; i <= bilangan; i++ {
        if bilangan%i == 0 {
            faktor = append(faktor, i)
            fmt.Printf("%d ", i)
        }
    }
    fmt.Println()
    if len(faktor) == 2 {
        fmt.Println("Prima: true")
    } else {
        fmt.Println("Prima: false")
    }
}
```

```
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Users\alden\OneDrive\Documents\GOLANG>
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\alden\OneDrive\Documents\GOLANG>
```

Pengguna diminta untuk memasukkan sebuah bilangan integer menggunakan `fmt.Scanln(&bilangan)`. Program kemudian mengambil bilangan tersebut untuk mencari faktor-faktornya. Program melakukan perulangan dari 1 hingga bilangan yang dimasukkan oleh pengguna (`for i := 1; i <= bilangan; i++`). Setiap angka yang merupakan faktor dari bilangan tersebut (jika `bilangan % i == 0`), ditambahkan ke dalam slice faktor menggunakan `append(faktor, i)`, dan juga ditampilkan dengan `fmt.Printf("%d ", i)`.

Setelah semua faktor ditemukan, program memeriksa apakah bilangan tersebut adalah bilangan prima.

Bilangan prima hanya memiliki 2 faktor (1 dan bilangan itu sendiri), sehingga jika panjang slice faktor adalah 2 (`len(faktor) == 2`), maka program mencetak "Prima: true".

Jika panjang slice lebih dari 2, maka bilangan tersebut bukan prima, dan program mencetak "Prima: false". Program menampilkan semua faktor dari bilangan yang dimasukkan.

Kemudian, program menentukan dan menampilkan apakah bilangan tersebut adalah bilangan prima atau bukan, dengan mengacu pada jumlah faktornya.