

**LAPORAN PRAKTIKUM
ALGORITMA PEMROGRAMAN 2**

MODUL 2

RIVIEW STRUKTUR KONTROL



Oleh:

IQBAL BAWANI

2311102130

S1IF-11-07

S1 TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Dalam Bahasa pemrograman go ada 2 komponen dalam program utama yaitu package main dan func main (). Package main penanda bahwa file ini berisi program utama, sedangkan func main berisikan kode program utama.

Tipe data

| 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 ⁹ ..10 ⁹ 64 bit: -10 ¹⁹ ..10 ¹⁹ bergantung platform 0..255 0..4294967295 0..(2 ⁶⁴ -1) |
| real | float32 float64 | 32bit: -3.4E+38 .. 3.4E+38 64bit: -1.7E+308 .. 1.7E+308 |
| boolean (atau logikal) | bool | false dan true |
| karakter | byte //uint8 rune //int32 | tabel ASCII/UTF-8 tabel UTF-16 |
| string | string | |

Operato Go lang

| Operator dalam Go | Tipe data terkait | Keterangan |
|-------------------|------------------------------|--|
| + | string integer dan real | konkatenasi 2 string operasi penjumlahan |
| - * / | integer dan real | operasi pengurangan, perkalian, dan pembagian |
| % | integer | operasi sisa pembagian integer (modulo) |
| & ^ &^ | integer | operasi per-bit AND, OR, XOR, AND-NOT |
| << >> | integer dan unsigned integer | operasi geser bit kiri/kanan sebanyak unsigned integer yang diberikan |
| < <= >= > == != | 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 boolean AND, OR, dan NOT |
| * & | variabel apasaja | mendapatkan data dari lokasi memori dan mendapatkan lokasi dari variabel |

Percabngan dan perulangan pada

Percabangan di bahasa Go (Golang) digunakan untuk mengontrol alur eksekusi program berdasarkan kondisi tertentu. Dengan menggunakan pernyataan seperti if, else if, dan else, Golang memungkinkan eksekusi blok kode yang berbeda sesuai dengan hasil evaluasi ekspresi logika. Selain itu, Go mendukung percabangan melalui switch yang efisien untuk memeriksa banyak kondisi tanpa perlu menuliskan banyak if. Struktur ini membantu membuat kode lebih bersih dan mudah dipahami, karena dapat mengurangi kompleksitas ketika bekerja dengan berbagai skenario logika.

II. GUIDED

Guided1 (Percobaan tipe data)

Source code:

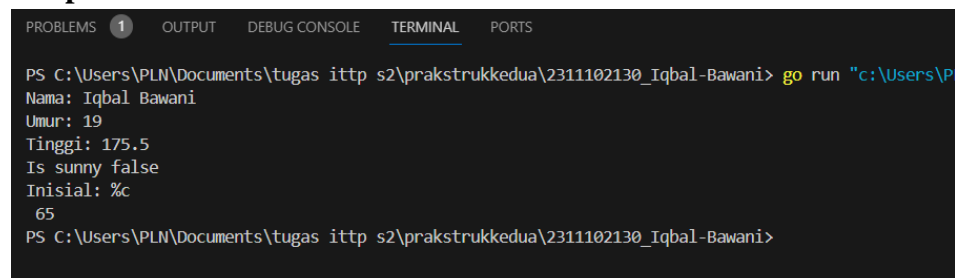
```
package main

import "fmt"

func main() {
    var nama string = "Iqbal Bawani"
    var umur int = 19
    var tinggi float64 = 175.5
    var isSunny bool = false
    var inisial rune = 'A'

    fmt.Println("Nama:", nama)
    fmt.Println("Umur:", umur)
    fmt.Println("Tinggi:", tinggi)
    fmt.Println("Is sunny", isSunny)
    fmt.Println("Inisial: %c\n", inisial)
}
```

Output =



Penjelasan =

Pada Go, tipe data `rune` digunakan untuk merepresentasikan karakter Unicode, yang merupakan alias dari `int32`. Ketika ingin mencetak `rune` sebagai karakter, kamu harus menggunakan format string khusus `%c` dalam fungsi `fmt.Printf`, karena fungsi `fmt.Println` akan mencetak nilai numerik dari `rune` tersebut, bukan karakternya. Dengan `%c`, nilai `rune` akan ditampilkan sebagai karakter yang sesuai dengan nilai Unicode-nya. Ini penting untuk memastikan bahwa karakter yang diinginkan tercetak dengan benar dalam output program.

Guided 2 (Tahun Kabisat)

Source code :

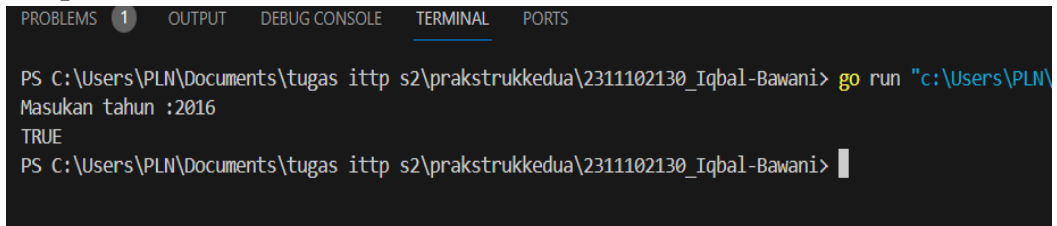
```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Masukan tahun :")
    fmt.Scan(&tahun)

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

Output =



```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PLN\Documents\tugas http s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas http s2\prakstrukkedua\2311102130_Iqbal-Bawani"
Masukan tahun :2016
TRUE
PS C:\Users\PLN\Documents\tugas http s2\prakstrukkedua\2311102130_Iqbal-Bawani>
```

Penjelasan =

Program di atas berfungsi untuk menentukan apakah suatu tahun merupakan tahun kabisat. Pengguna diminta untuk memasukkan nilai tahun, kemudian program memeriksa kondisinya. Tahun kabisat terjadi jika tahun tersebut habis dibagi 4 ****dan**** tidak habis dibagi 100, ****kecuali**** jika tahun tersebut juga habis dibagi 400. Kondisi ini diterapkan menggunakan operator logika `&&` (dan) serta `||` (atau). Jika syarat tersebut terpenuhi, program akan mencetak "TRUE", menandakan bahwa tahun tersebut adalah tahun kabisat. Jika tidak, program mencetak "false", yang berarti bukan tahun kabisat.

Guided 3

Source code

```
package main

import "fmt"
```

```

func main() {
    var temperaturCelcius float64

    fmt.Println("Masukkan temperatur dalam Celcius:")
    fmt.Scanln(&temperaturCelcius)

    temperaturFahrenheit := (temperaturCelcius * 9 / 5) + 32
    temperaturKelvin := temperaturCelcius + 273.15
    temperaturReamur := temperaturCelcius * 4 / 5

    fmt.Printf("Temperatur dalam Fahrenheit: %.2f\n", temperaturFahrenheit)
    fmt.Printf("Temperatur dalam Kelvin: %.2f\n", temperaturKelvin)
    fmt.Printf("Temperatur dalam Reamur: %.2f\n", temperaturReamur)
}

```

Output =

```

PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqba
Masukkan temperatur dalam Celcius:
50
Temperatur dalam Fahrenheit: 122.00
Temperatur dalam Kelvin: 323.15
Temperatur dalam Reamur: 40.00
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqba

```

Penjelasan =

Program Go ini berfungsi untuk mengonversi suhu dari derajat Celcius ke beberapa skala suhu lainnya, yaitu Fahrenheit, Kelvin, dan Reamur. Pertama, program meminta pengguna untuk memasukkan nilai temperatur dalam derajat Celcius dan menyimpannya dalam variabel `temperaturCelcius`. Setelah menerima input, program menghitung konversi suhu menggunakan rumus yang sesuai untuk setiap skala: suhu dalam Fahrenheit dihitung dengan rumus $(F = C * 9 / 5) + 32$, suhu dalam Kelvin menggunakan rumus $(K = C + 273.15)$, dan suhu dalam Reamur dihitung dengan rumus $(R = C * 4 / 5)$. Akhirnya, program mencetak hasil konversi dengan format dua desimal untuk memberikan informasi yang jelas dan mudah dipahami kepada pengguna. Dengan demikian, program ini menyediakan cara yang sederhana dan efektif untuk melakukan konversi suhu di antara berbagai skala yang umum digunakan.

III. UNGUIDED

Uguided 2b ke 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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**false** berarti **berhasil** adalah input/read):

```
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

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
```

Source code

package main

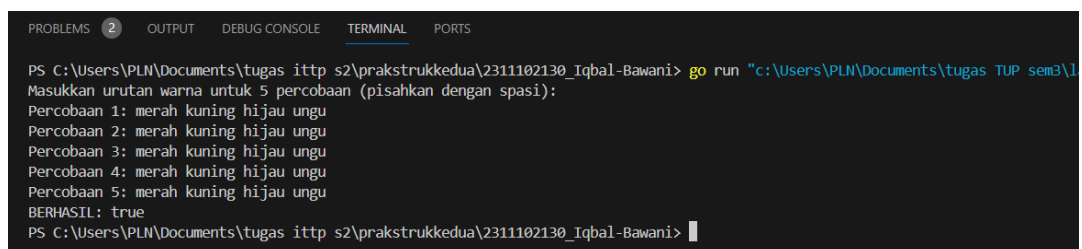
```
import (
    "bufio"
    "fmt"
    "os"
    "strings"
)

func main() {
    warna := []string{"merah", "kuning", "hijau", "ungu"}
    var isCorrect = true
    scanner := bufio.NewScanner(os.Stdin)
```

```
    fmt.Println("Masukkan urutan warna untuk 5 percobaan (pisahkan dengan spasi):")
```

```
    for i := 1; i <= 5; i++ {  
        fmt.Printf("Percobaan %d: ", i)  
        scanner.Scan()  
        input := strings.Fields(scanner.Text())  
        if len(input) != len(warna) {  
            isCorrect = false  
            break  
        }  
        for j := 0; j < len(warna); j++ {  
            if input[j] != warna[j] {  
                isCorrect = false  
                break  
            }  
        }  
        if !isCorrect {  
            break  
        }  
    }  
  
    if isCorrect {  
        fmt.Println("BERHASIL: true")  
    } else {  
        fmt.Println("BERHASIL: false")  
    }  
}
```

Output =



```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas TUP sem3\1  
Masukkan urutan warna untuk 5 percobaan (pisahkan dengan spasi):  
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\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>
```

Penjelasan =

Program Go di atas memeriksa apakah urutan warna yang dimasukkan oleh pengguna selama 5 percobaan sesuai dengan urutan warna yang sudah ditentukan, yaitu `["merah", "kuning", "hijau", "ungu"]`. Pertama, program mendeklarasikan array `warna` yang berisi urutan warna yang benar, dan variabel `isCorrect` yang awalnya diatur ke `true`. Kemudian, menggunakan fungsi `bufio.NewScanner` untuk membaca input dari pengguna, program meminta pengguna untuk memasukkan urutan warna

untuk setiap percobaan satu per satu, dengan warna yang dipisahkan oleh spasi. Untuk setiap percobaan, program memeriksa apakah jumlah warna yang dimasukkan sesuai dengan jumlah yang diharapkan (yaitu 4 warna). Jika tidak sesuai, variabel `isCorrect` diatur ke `false` dan perulangan dihentikan. Selanjutnya, program membandingkan setiap warna yang dimasukkan dengan warna yang benar. Jika ditemukan perbedaan, variabel `isCorrect` juga diubah menjadi `false`, dan perulangan segera dihentikan. Setelah semua percobaan selesai, jika seluruh urutan warna benar di setiap percobaan, program mencetak `BERHASIL: true`; jika ada satu percobaan yang salah, maka program mencetak `BERHASIL: false`. Program ini menggunakan logika sederhana dengan perulangan dan perbandingan untuk memvalidasi urutan input pengguna sesuai dengan urutan yang sudah ditentukan.

Unguided 2b ke 2

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

23 | Modul Praktikum Algoritma dan Pemrograman 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.

(Petunjuk: gunakan operasi penggabungan string dengan operator "+").

Tampilkan isi pita setelah proses input selesai.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

| | |
|--------------------------------|--------|
| N: <u>3</u> | N : 3 |
| Bunga 1: <u>Kertas</u> | Pita : |
| Bunga 2: <u>Mawar</u> | |
| Bunga 3: <u>Tulip</u> | |
| Pita: Kertas – Mawar – Tulip – | |

Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan 'SELESAI'. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

| | |
|--------------------------------|-------------------------|
| Bunga 1: <u>Kertas</u> | Bunga 1: <u>SELESAI</u> |
| Bunga 2: <u>Mawar</u> | Pita : |
| Bunga 3: <u>Tulip</u> | Bunga: 0 |
| Bunga 4: <u>SELESAI</u> | |
| Pita: Kertas – Mawar – Tulip – | |
| Bunga: 3 | |

Source code

```
package main

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

func main() {
    var N int
    fmt.Print("Masukkan jumlah bunga yang akan dimasukkan: ")
    fmt.Scan(&N)

    bunga := []string{ }
    reader := bufio.NewReader(os.Stdin)

    for i := 1; i <= N; i++ {
        fmt.Printf("Bunga %d: ", i)

        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

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

        if input != "" {
            bunga = append(bunga, input)
        } else {
            fmt.Println("Nama bunga tidak boleh kosong!")
            i--
        }
    }

    result := strings.Join(bunga, " - ")
    fmt.Println("Pita : " + result)
    fmt.Printf("Bunga : %d\n", len(bunga))
}
```

Output =

```
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tu
Masukkan jumlah bunga yang akan dimasukkan: 4 mawar
Bunga 1: Bunga 2: melati
Bunga 3: kertas
Bunga 4: tulip
Pita : mawar - melati - kertas - tulip
Bunga : 4
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>
```

Penjelasan =

Program Go ini meminta pengguna memasukkan sejumlah nama bunga, sesuai dengan angka N yang dimasukkan di awal. Setiap nama bunga akan disimpan dalam array dan dipisahkan dengan tanda " – ". Jika pengguna mengetik "SELESAI", program akan berhenti sebelum N input tercapai. Program juga menangani input kosong dengan memberikan peringatan dan meminta ulang input bunga tanpa menghitung iterasi tersebut. Setelah semua bunga dimasukkan atau pengguna mengetik "SELESAI", program menampilkan daftar bunga yang telah diinput dan jumlah total bunga yang berhasil dimasukkan

Unguided 2b ke 3

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

ian 24 | Modul Praktikum Algoritma dan Pemrograman 2

```
Masukan berat belanjaan di kedua kantong: 5.5 1.0
Masukan berat belanjaan di kedua kantong: 7.1 8.5
Masukan berat belanjaan di kedua kantong: 2 6
Masukan berat belanjaan di kedua kantong: 9 5.8
Proses selesai.
```

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

```
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
```

Source code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKiri, beratKanan float64
    var totalBerat float64
    Iqbalbawani := true

    for Iqbalbawani {
        fmt.Print("Masukkan berat belanjaan di kedua kantong : ")
        fmt.Scan(&beratKiri, &beratKanan)

        if beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Berat kantong tidak boleh negatif!")
            break
        }

        totalBerat = beratKiri + beratKanan
        selisih := math.Abs(beratKiri - beratKanan)

        if selisih > 9 {
            fmt.Printf("Sepeda motor Pak Andi akan oleng: true (Selisih: %.1f)\n", selisih)
            Iqbalbawani = false
            break
        } else {
            fmt.Printf("Sepeda motor Pak Andi akan oleng: false (Selisih: %.1f)\n", selisih)
        }

        if totalBerat > 150 {
            fmt.Printf("Total berat belanjaan melebihi 150 kg: %.1f kg\n", totalBerat)
            Iqbalbawani = false
            break
        }

        if beratKiri > 9 || beratKanan > 9 {
            fmt.Printf("Salah satu kantong melebihi 9 kg! Kantong Kiri: %.1f, Kantong Kanan: %.1f\n", beratKiri, beratKanan)
            Iqbalbawani = false
        }
    }
}
```

```

        break
    }
}

fmt.Println("Proses selesai.")
}

```

Output =

```

PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Docu
Masukkan berat belanjaan di kedua kantong : 5 10
Sepeda motor Pak Andi akan oleng: false (Selisih: 5.0)
Salah satu kantong melebihi 9 kg! Kantong Kiri: 5.0, Kantong Kanan: 10.0
Proses selesai.
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> 

```

Penjelasan =

Program Go di atas mensimulasikan pengecekan keseimbangan berat belanja yang dibawa oleh Pak Andi menggunakan dua kantong di sepeda motornya. Program meminta pengguna untuk memasukkan dua nilai, yaitu berat belanjaan di kantong kiri dan kanan, dalam satuan kilogram. Program memverifikasi tiga kondisi: apakah selisih berat antara kedua kantong lebih dari 9 kg, apakah total berat dari kedua kantong melebihi 150 kg, atau apakah salah satu kantong memiliki berat lebih dari 9 kg. Jika salah satu kondisi ini terpenuhi, program akan menampilkan pesan yang sesuai, misalnya apakah sepeda motor akan oleng jika selisih berat terlalu besar, dan program akan berhenti. Variabel Iqbalbawani berfungsi sebagai kontrol untuk perulangan, dan selama nilainya tetap true, program akan terus meminta input berat dari pengguna. Program juga menangani input negatif, dengan menampilkan pesan kesalahan jika berat yang dimasukkan kurang dari 0, dan menghentikan eksekusi dalam kasus tersebut.

Unguided 2b ke 4

Diberikan sebuah persamaan sebagai berikut ini.

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

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):

```
Nilai K = 100
Nilai f(K) = 1.0000061880
```

$\sqrt{2}$ merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihiperir dengan rumus berikut:

25 | Modul Praktikum Algoritma dan Pemrograman 2

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

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):

| | |
|---|--|
| 1 | Nilai K = <u>10</u> Nilai akar 2 = 1.4062058441 |
| 2 | Nilai K = <u>100</u> Nilai akar 2 = 1.4133387072 |
| 3 | Nilai K = <u>1000</u> Nilai akar 2 = 1.4141252651 |

Source code
package main

```
import (  
    "fmt"
```

```

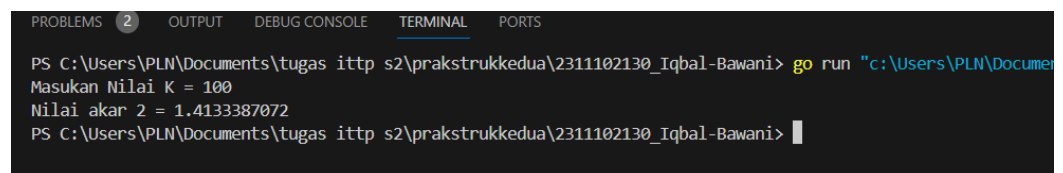
)

func main() {
    var K int
    fmt.Print("Masukan Nilai K = ")
    fmt.Scan(&K)

    Iqbalbawani := 1.0
    for k := 0; k <= K; k++ {
        k := float64(k)
        hasil := (4*k + 2) * (4*k + 2) / ((4*k + 1) * (4*k + 3))
        Iqbalbawani *= hasil
    }
    fmt.Printf("Nilai akar 2 = %.10f\n", Iqbalbawani)
}

```

Output =



```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PLN\Documents\tugas_1\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas_1\prakstrukkedua\2311102130_Iqbal-Bawani.go"
Masukan Nilai K = 100
Nilai akar 2 = 1.4133387072
PS C:\Users\PLN\Documents\tugas_1\prakstrukkedua\2311102130_Iqbal-Bawani>

```

Penjelasan =

Program Go yang disediakan menghitung nilai perkiraan dari akar kuadrat dua menggunakan metode deret matematis, yang beroperasi berdasarkan rumus yang menggabungkan fungsi polinomial. Program ini dimulai dengan mengimpor package `fmt` untuk menangani input dan output. Di dalam fungsi `main`, pengguna diminta untuk memasukkan nilai integer `K`, yang berfungsi sebagai batas atas dalam perhitungan deret. Variabel `iqbalbawani` diinisialisasi dengan nilai 1.0 dan berfungsi sebagai akumulator untuk hasil akhir. Selanjutnya, program menggunakan loop yang berjalan dari 0 hingga nilai `K`, di mana setiap iterasi menghitung nilai berdasarkan rumus yang melibatkan perhitungan polinomial pada `k`, yang dikonversi menjadi tipe `float64` untuk menjaga presisi. Rumus tersebut, $(4k + 2) * (4k + 2) / ((4k + 1) * (4k + 3))$ menghasilkan faktor yang secara bertahap dikalikan dengan `iqbalbawani`, sehingga mendekati nilai akar kuadrat dua. Setelah loop selesai, hasil akhir ditampilkan dengan format presisi sepuluh angka desimal. Program ini secara efektif menunjukkan bagaimana metode numerik dapat digunakan untuk mendekati nilai yang tidak dapat dihitung secara tepat, dengan memberikan pengguna fleksibilitas untuk menentukan tingkat ketelitian dengan

mengubah nilai `K`, di mana semakin besar `K`, semakin akurat hasil yang diperoleh.

Unguided 2c ke1

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**teks bergaris bawah** adalah input/read):

| | |
|---|--|
| 1 | Contoh #1 Berat parcel (gram): <u>8500</u> Detail berat: 8 kg + 500 gr Detail biaya: Rp. 80000 + Rp. 2500 Total biaya: Rp. 82500 |
| 2 | Contoh #2 Berat parcel (gram): <u>9250</u> Detail berat: 9 kg + 250 gr Detail biaya: Rp. 90000 + Rp. 3750 Total biaya: Rp. 93750 |
| 3 | Contoh #3 Berat parcel (gram): <u>11750</u> Detail berat: 11 kg + 750 gr Detail biaya: Rp. 110000 + Rp. 3750 Total biaya: Rp. 110000 |

Source code

```
package main
```

```
import "fmt"
```

```
func main() {
```

```
    var berat, kilogram, gram, biayakirim, tambahanbiaya, totalbiaya int
```

```
    fmt.Print("Berat parcel (gram): ")
```



```

    fmt.Scan(&berat)

    kilogram = berat / 1000
    gram = berat % 1000

    if kilogram > 10 {
        tambahanbiaya = 0
    } else if gram >= 500 {
        tambahanbiaya = gram * 5
    } else {
        tambahanbiaya = gram * 15
    }

    biayakirim = kilogram * 10000
    totalbiaya = biayakirim + tambahanbiaya

    fmt.Printf("Detail berat: %d kg + %d gr\n", kilogram, gram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayakirim,
    tambahanbiaya)
    fmt.Printf("Total biaya: Rp. %d\n", totalbiaya)
}

```

Output =

Penjelasan =

Kode Go yang disediakan berfungsi untuk menghitung biaya pengiriman berdasarkan berat parcel yang dimasukkan pengguna dalam gram. Pada awal program, variabel `berat`, `kilogram`, `gram`, `biayakirim`, `tambahanbiaya`, dan `totalbiaya` dideklarasikan sebagai integer. Pengguna diminta untuk memasukkan berat parcel, yang kemudian diproses untuk menghitung nilai dalam kilogram dan gram melalui operasi pembagian dan modulus. Selanjutnya, program mengevaluasi biaya tambahan berdasarkan berat, dengan ketentuan bahwa jika berat dalam kilogram melebihi 10, tidak ada tambahan biaya yang dikenakan; jika berat dalam gram mencapai atau melebihi 500, tambahan biaya ditetapkan sebesar Rp 5 per gram; dan untuk berat di bawah 500 gram, biaya tambahan menjadi Rp 15 per gram. Biaya pengiriman dasar dihitung dengan mengalikan berat dalam kilogram dengan tarif Rp 10.000 per kilogram. Setelah perhitungan biaya selesai, program mencetak detail berat dalam format kilogram dan gram, serta memaparkan rincian biaya kirim dan tambahan. Terakhir, total biaya yang harus dibayar dihitung dan ditampilkan. Struktur kode ini memberikan cara yang jelas dan sistematis untuk mengkalkulasi biaya pengiriman, sehingga pengguna dapat

dengan mudah memahami rincian biaya yang terlibat dalam pengiriman parcel berdasarkan beratnya.

Unguided 2c ke2

Source code (soal)

```
package main
import "fmt"
func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)
    if nam > 80 {
        nam = "A"
    }
    if nam > 72.5 {
        nam = "AB"
    }
    if nam > 65 {
        nam = "B"
    }
    if nam > 57.5 {
        nam = "BC"
    }
    if nam > 50 {
        nam = "C"
    }
    if nam > 40 {
        nam = "D"
    } else if nam <= 40 {
        nam = "E"
    }
    fmt.Println("Nilai mata kuliah: ", nmk)
}
```

Pertanyaan =

2. Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

| NAM | NMK |
|----------------------|-----|
| $NAM > 80$ | A |
| $72.5 < NAM \leq 80$ | AB |

Halaman 29 | Modul Praktikum Algoritma dan Pemrograman 2

| | |
|----------------------|----|
| $65 < NAM \leq 72.5$ | B |
| $57.5 < NAM \leq 65$ | BC |
| $50 < NAM \leq 57.5$ | C |
| $40 < NAM \leq 50$ | D |
| $NAM \leq 40$ | E |

Program berikut menerima input sebuah bilangan riil yang menyatakan NAM. Program menghitung NMK dan menampilkannya.

Jawablah pertanyaan-pertanyaan berikut:

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

Output =

a.)

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run
Masukkan nilai: 80.1
Nilai Indeks untuk nilai 80.10 adalah A
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> []
```

b) kesalahan kode diatas yaitu tidak ada sama dengn sehingga misal nilai 80 tidak masuk ke 'A'. jika saja ≥ 80 nam = 'A', maka 80 termasuk ke A

c)source code yang sudah di perbaiki

```
package main
```

```

import (
    "fmt"
)

func main() {
    var nam float64
    var nmk string

    fmt.Print("Masukkan nilai: ")
    fmt.Scan(&nam)

    if nam >= 80 {
        nmk = "A"
    } else if nam >= 70 {
        nmk = "B"
    } else if nam >= 60 {
        nmk = "C"
    } else if nam >= 45 {
        nmk = "D"
    } else if nam >= 40 {
        nmk = "E"
    } else {
        nmk = "F"
    }

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

```

```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani\main.go"
Masukkan nilai: 93
Nilai Indeks untuk nilai 93.00 adalah A
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>

```

```

PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani\main.go"
Masukkan nilai: 49.5
Nilai Indeks untuk nilai 49.50 adalah D
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>

```

```

PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani\main.go"
Masukkan nilai: 70.6
Nilai Indeks untuk nilai 70.60 adalah B
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>

```

Penjelasan =

Program Go yang disediakan dirancang untuk menerima input nilai dari pengguna dalam bentuk angka desimal dan kemudian menentukan indeks nilai berdasarkan rentang yang telah ditentukan. Setelah mengimpor package `fmt` untuk menangani input dan output, program memulai dengan mendeklarasikan dua variabel: `nam`, yang berjenis `float64` untuk menyimpan nilai yang dimasukkan, dan `nmk`, yang berjenis `string` untuk menyimpan indeks nilai yang sesuai. Pengguna diminta untuk memasukkan nilai, dan setelah itu, program menggunakan struktur kondisi `if`-`else if` untuk mengevaluasi nilai tersebut. Setiap kondisi memeriksa apakah nilai berada dalam rentang tertentu, seperti 80 atau lebih untuk indeks "A", 70 hingga 79 untuk "B", dan seterusnya hingga nilai di bawah 40 yang menghasilkan indeks "F". Setelah menentukan indeks, program mencetak hasilnya dalam format yang menunjukkan nilai yang dimasukkan dan indeks yang diberikan. Dengan struktur yang jelas dan sederhana, program ini memberikan cara yang efektif untuk memberikan umpan balik kepada pengguna tentang kinerja mereka berdasarkan nilai yang dimasukkan.

Unguided 2c ke3

3. 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!

Perhatikan contoh sesi interaksi program seperti di bawah ini (**taba bergaris bawah** adalah input/read):

| | |
|---|-----------------------------------|
| Bilangan: 12 Faktor: 1 2 3 4 6 12 | Bilangan: 7 Faktor: 1 7 |
|---|-----------------------------------|

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.

Perhatikan contoh sesi interaksi program seperti di bawah ini (**taba bergaris bawah** adalah input/read):

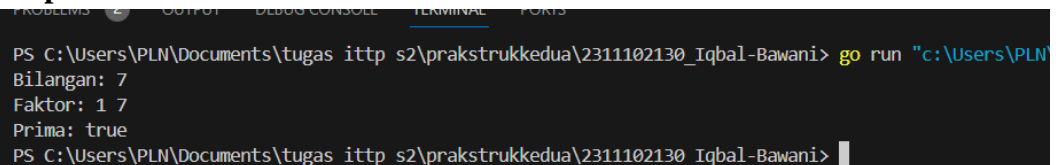
| | |
|---|--|
| Bilangan: 12 Faktor: 1 2 3 4 6 12 Prima: false | Bilangan: 7 Faktor: 1 7 Prima: true |
|---|--|

Source code

```
package main

import (
    "fmt"
)

func main() {
    var Iqbalbawani int
    fmt.Print("Bilangan: ")
    fmt.Scanln(&Iqbalbawani)
    fmt.Print("Faktor: ")
    for i := 1; i <= Iqbalbawani; i++ {
        if Iqbalbawani%i == 0 {
            fmt.Print(i, " ")
        }
    }
    fmt.Println()
    prima := true
    if Iqbalbawani <= 1 {
        prima = false
    } else {
        for i := 2; i*i <= Iqbalbawani; i++ {
            if Iqbalbawani%i == 0 {
                prima = false
                break
            }
        }
    }
    fmt.Println("Prima:", prima)
}
```

Output =

```
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani> go run "c:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Users\PLN\Documents\tugas ittp s2\prakstrukkedua\2311102130_Iqbal-Bawani>
```

Penjelasan =

Program Go ini bertujuan untuk menentukan faktor dari suatu bilangan yang dimasukkan oleh pengguna serta untuk mengecek apakah bilangan tersebut merupakan bilangan prima. Pertama, program meminta pengguna untuk memasukkan sebuah bilangan dengan menggunakan

``fmt.Scanln(&Iqbalbawani)``, yang disimpan dalam variabel bertipe ``int`` bernama ``Iqbalbawani``. Setelah menerima input, program menampilkan faktor-faktor dari bilangan tersebut dengan perulangan ``for`` yang memeriksa setiap angka dari 1 hingga ``Iqbalbawani``. Dalam setiap iterasi, program menggunakan kondisi ``if Iqbalbawani%i == 0`` untuk menentukan apakah angka ``i`` adalah faktor dari ``Iqbalbawani``, dan jika kondisi tersebut terpenuhi, angka tersebut dicetak sebagai faktor.

Selanjutnya, program memeriksa apakah bilangan yang dimasukkan adalah bilangan prima. Program menetapkan variabel ``prima`` ke ``true`` dan memeriksa apakah ``Iqbalbawani`` kurang dari atau sama dengan 1; jika iya, maka bilangan tersebut tidak dapat dianggap sebagai bilangan prima, sehingga ``prima`` diubah menjadi ``false``. Jika bilangan lebih besar dari 1, perulangan kedua dimulai dari 2 dan berlangsung hingga ``i*i`` kurang dari atau sama dengan ``Iqbalbawani``, menggunakan logika yang efisien untuk memeriksa faktor hingga akar kuadrat bilangan tersebut. Jika ditemukan bahwa ``Iqbalbawani`` dapat dibagi oleh angka lain, variabel ``prima`` diubah menjadi ``false`` dan perulangan dihentikan. Akhirnya, program mencetak hasil pengecekan apakah bilangan tersebut adalah bilangan prima dengan menampilkan nilai dari variabel ``prima``. Program ini dengan demikian memberikan informasi yang komprehensif mengenai faktor dan status primalitas dari bilangan yang dimasukkan pengguna.

DAFTAR PUSTAKA

Modul praktikum 2 riview struktur kontrol