

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR CONTROL**



Disusun Oleh :

Ghilbran Alfaries Pryma / 2311102267

IF 11 05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Apa itu golang? , Golang adalah bahasa pemrograman modern yang dikembangkan oleh Google, dirancang untuk efisiensi, kemudahan, dan keamanan. Dengan sintaks yang bersih dan mirip dengan bahasa C, Golang memungkinkan pengembang membuat aplikasi yang skalabel dan berkinerja tinggi. Fitur-fitur unggulannya meliputi manajemen memori otomatis (garbage collection), dukungan untuk konkurensi melalui goroutine dan channel, serta sistem pengetikan statis yang kuat. Golang sering digunakan untuk membangun aplikasi web, layanan backend, infrastruktur cloud, dan alat-alat DevOps. Kepopulerannya terus meningkat karena kemampuannya dalam menangani beban kerja yang berat dan kompleks dengan cara yang elegan dan efisien.

Golang bisa untuk apa saja?

- If-Else: Kondisional pada umumnya biasanya menggunakan `if ... else`. Dan hematnya code golang itu tidak perlu ada kurung tutup dan kurung buka yang biasanya program yang lain harus menggunakan itu. Contoh program kondisional seperti ini². Switch: Alternatif ``if`` untuk memeriksa banyak kondisi. Tidak memerlukan ``break``, dan mendukung `**expressionless switch**`.
 - For Loop: Perulangan pada golang hanya operasi `for` saja tidak punya `foreach`, `while`, `do`, `until` tetapi dengan menggunakan `for` ini semua operasi dalam aritmatika ataupun algoritma sudah bisa dilakukan. Misalkan contoh dasar kita akan mengeluarkan bilangan/angka dari 0-1. Maka code-nya seperti ini
 - Switch Kondisional program juga bisa menggunakan `switch` yang mana lebih mudah untuk satuan pengecekan. Bila lebih complex suatu kondisi program lebih bisa dipahami menggunakan `if else` saja
- Dan masih banyak lagi yang bisa dilakukan dengan golang.

II. GUIDED I

1.Sourcecode

```
package main
import "fmt"
func main() {
    nama := "Ghilbran Alfaries Pryma"
    fmt.Print(nama)
}
```

Screenshoot Output

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided1Modul2>
go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided1Modul2\main.go"
Ghilbran Alfaries Pryma
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided1Modul2>
```

Deskripsi Program

Program ini menunjukkan bagaimana cara mendeklarasikan variabel dan mencetak nilai ke konsol

GUIDED II

Source Code

```
package main

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

func main() {
    //urutan warna yang benar
    correctOrder := []string{"merah", "kuning", "hijau", "ungu"}

    //membaca input untuk 5 percobaan
    reader := bufio.NewReader(os.Stdin)
    success := true

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)

        //membaca input dari pengguna
        input, _ := reader.ReadString('\n')
```

```

input = strings.TrimSpace(input)

//memisahkan input berdasarkan spasi
colors := strings.Split(input, " ")

//mengecek apakah urutan warna sesuai
for j := 0; j < 4; j++ {
    if colors[j] != correctOrder[j] {
        success = false
        break
    }
}

//jika ada percobaan yang tidak sesuai, keluar dari loop
if !success {
    break
}

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

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2\main.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2\main.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
BERHASIL: false
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2>

```

Deskripsi program

Program yang dibuat dalam bahasa Go ini dirancang untuk memeriksa urutan warna yang dimasukkan oleh pengguna selama lima percobaan. Dengan menggunakan struktur kontrol dasar seperti loop dan kondisi, program ini mengimplementasikan logika untuk membandingkan input pengguna dengan urutan warna yang benar, yaitu "merah", "kuning", "hijau", dan "ungu".

Meskipun program berfungsi untuk tujuan yang dimaksud, terdapat beberapa aspek yang perlu diperbaiki, seperti pengaturan ulang status `success` untuk setiap percobaan dan pengecekan panjang input untuk memastikan bahwa pengguna memberikan tepat empat warna.

Dengan melakukan perbaikan ini, program akan lebih robust dan memberikan umpan balik yang lebih baik kepada pengguna jika input tidak sesuai dengan yang diharapkan. Secara keseluruhan, program ini memberikan contoh dasar tentang penggunaan input/output, manipulasi string, dan kontrol alur dalam pemrograman

GUIDED III

Source Code

```
package main

import (
    "fmt"
)

func main(){
    var a,b,c,d,e int
    var hasil int
    fmt.Print("Inputkan angka: ")
    fmt.Scanln(&a, &b, &c, &d, &e)

    hasil = a+b+c+d+e
    fmt.Println("Hasil Penjumlahan ",a,b,c,d,e, "adalah ",hasil)
}
```

Screenshot Output

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided3Modul2>
go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided3Modul2\main.go"
Inputkan angka: 1 2 3 4 5
Hasil Penjumlahan 1 2 3 4 5 adalah 15
```

Deskripsi Program

Program ini cara melakukan input dari pengguna, menyimpan nilai ke dalam variabel, serta melakukan operasi aritmetika sederhana dalam bahasa pemrograman Go.

Beberapa perbaikan yang bisa dipertimbangkan untuk meningkatkan program ini:

- **Validasi Input:** Menambahkan pengecekan untuk memastikan bahwa input yang dimasukkan adalah angka.
- **Penggunaan Slice:** Menggunakan slice untuk mengelola angka yang dimasukkan, yang dapat membuat kode lebih fleksibel jika jumlah angka yang dimasukkan berubah.
- **Output yang Lebih Jelas:** Menggunakan format string untuk output yang lebih terstruktur.

Dengan tambahan tersebut, program ini dapat menjadi alat yang lebih robust dan user-friendly.

GUIDED IV

Source Code

```
package main

import "fmt"

func main() {
    var nam float32
    var nmk string

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

    // Logika penentuan nilai huruf berdasarkan nilai numerik
```

```

if nam > 80 {
    nmk = "A"
} else if nam > 72.5 {
    nmk = "B"
} else if nam > 65 {
    nmk = "C"
} else if nam > 50 {
    nmk = "D"
} else if nam > 40 {
    nmk = "E"
} else {
    nmk = "F"
}

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

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided4Modul2>
go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided4Modul2\main.go"
Masukkan nilai : 90
Nilai Indeks untuk nilai 90.00 adalah A

```

Deskripsi Program

Program ini untuk memahami penggunaan input/output, struktur kontrol `if-else`, dan konversi tipe data dalam bahasa pemrograman Go.

Beberapa perbaikan yang bisa dipertimbangkan untuk meningkatkan program ini:

- **Validasi Input:** Menambahkan pengecekan untuk memastikan bahwa input yang dimasukkan adalah angka valid.
- **Penggunaan Fungsi:** Memecah logika penilaian menjadi fungsi terpisah untuk meningkatkan modularitas dan keterbacaan kode.
- **Pesan Error:** Menampilkan pesan error jika nilai yang dimasukkan tidak sesuai dengan rentang yang diharapkan.

Dengan tambahan tersebut, program ini bisa menjadi alat yang lebih robust dan user-friendly.

III. UNGUIDED

1. Soal Studi Case

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 berturutan 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 (teks bergaris bawah adalah input/read):

Source Code

```
package main

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

func main() {
    // Urutan warna yang benar
    correctOrder := []string{"merah", "kuning", "hijau", "ungu"}

    // Membaca input untuk 5 percobaan
    reader := bufio.NewReader(os.Stdin)
    success := true

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)

        // Membaca input dari pengguna
        input, _ := reader.ReadString('\n')
```



```
input = strings.TrimSpace(input)

// Memisahkan input berdasarkan spasi
colors := strings.Split(input, " ")

// Mengecek apakah urutan warna sesuai
for j := 0; j < 4; j++ {
    if colors[j] != correctOrder[j] {
        success = false
        break
    }
}

// Jika ada percobaan yang tidak sesuai, keluar dari loop
if !success {
    break
}

// Menampilkan hasil
if success {
    fmt.Println("BERHASIL : true")
} else {
    fmt.Println("BERHASIL : false")
}
}
```

Screenshoot Output

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2\main.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2\main.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
BERHASIL: false
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Guided2Modul2> |
```

Deskripsi Program

Program memasukkan urutan warna dalam lima percobaan, lalu mengecek apakah urutan warna yang diberikan oleh pengguna sudah sesuai dengan urutan warna yang benar. Program akan menampilkan hasil apakah seluruh percobaan berhasil atau tidak.

Program menetapkan urutan warna yang benar di dalam array `correctOrder` yang terdiri dari warna-warna: "merah", "kuning", "hijau", dan "ungu".

Program membaca input pengguna sebanyak 5 kali (5 percobaan).

Setiap input dipisahkan berdasarkan spasi menggunakan fungsi `strings.Split()`, sehingga menghasilkan array `colors` yang berisi daftar warna yang diinputkan oleh pengguna.

Program mengecek apakah setiap warna yang diinputkan oleh pengguna sesuai dengan urutan warna di `correctOrder`.

Jika urutan warna tidak sesuai pada salah satu percobaan, program akan keluar dari perulangan dan menandai percobaan sebagai gagal.

Jika semua percobaan berhasil (input pengguna selalu sesuai dengan urutan warna yang benar), program akan menampilkan "BERHASIL : true".

Jika ada satu percobaan yang gagal, program akan menampilkan "BERHASIL : false" dan berhenti mengecek sisa percobaan.

2. Soal Studi Case

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.

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

Source Code

```
package main

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

func main() {
    reader := bufio.NewReader(os.Stdin)

    // Input jumlah bunga
    fmt.Print("Masukkan bunga: ")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)

    // Konversi input ke integer
    var N int
    fmt.Sscanf(input, "%d", &N)

    pita := ""

    // Input nama-nama bunga
    for i := 1; i <= N; i++ {
        fmt.Printf("Bunga %d: ", i)
        bunga, _ := reader.ReadString('\n')
        bunga = strings.TrimSpace(bunga) // Hapus karakter newline

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

    // Output pita
    fmt.Println("Pita:", pita)
}

// pita := ""
// banyakBunga := 0
```

```
// reader := bufio.NewReader(os.Stdin)

// for {
//     banyakBunga++
//     fmt.Printf("Bunga %d: ", banyakBunga)
//     bunga, _ := reader.ReadString('\n')
//     bunga = strings.TrimSpace(bunga)

//     if bunga == "SELESAI" {
//         banyakBunga-- // Kurangi jumlah bunga karena "SELESAI" bukan bunga
//         break
//     }

//     if banyakBunga == 1 {
//         pita += bunga
//     } else {
//         pita += " - " + bunga
//     }
// }

// if pita != "" {
//     fmt.Println("Pita:", pita)
// } else {
//     fmt.Println("Pita:")
// }

// fmt.Println("Bunga:", banyakBunga)
// }
```

Screenshoot Output

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2B)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2B)Modul2\Unguided2(2B)Modul2.go"
Masukkan bunga: 2
Bunga 1: tulip
Bunga 2: mawar
Pita: tulip mawar
```

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):

SESUDAH DIUBAH

Sourcecode

```
package main

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

func main() {
    // reader := bufio.NewReader(os.Stdin)

    // // Input jumlah bunga
    // fmt.Print("Masukkan bunga: ")
    // input, _ := reader.ReadString('\n')
    // input = strings.TrimSpace(input)

    // // Konversi input ke integer
    // var N int
    // fmt.Sscanf(input, "%d", &N)

    // pita := ""

    // // Input nama-nama bunga
    // for i := 1; i <= N; i++ {
    //     fmt.Printf("Bunga %d: ", i)
    //     bunga, _ := reader.ReadString('\n')
    //     bunga = strings.TrimSpace(bunga) // Hapus karakter newline

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

    // // Output pita
    // fmt.Println("Pita:", pita)
    // }

    pita := ""
    banyakBunga := 0
```

```

reader := bufio.NewReader(os.Stdin)

for {
    banyakBunga++
    fmt.Printf("Bunga %d: ", banyakBunga)
    bunga, _ := reader.ReadString('\n')
    bunga = strings.TrimSpace(bunga)

    if bunga == "SELESAI" {
        banyakBunga-- // Kurangi jumlah bunga karena "SELESAI" bukan bunga
        break
    }

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

if pita != "" {
    fmt.Println("Pita:", pita)
} else {
    fmt.Println("Pita:")
}

fmt.Println("Bunga:", banyakBunga)
}

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2B)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2B)Modul2\Unguided2(2B)Modul2.go"
Bunga 1: mawar
Bunga 2: melati
Bunga 3: tulip
Bunga 4: SELESAI
Pita: mawar - melati - tulip
Bunga: 3

```

Deskripsi Program

Program meminta pengguna untuk memasukkan nama-nama bunga satu per satu dan menyusun nama-nama tersebut menjadi sebuah pita (rantai string) yang dihubungkan dengan tanda "-" antara setiap bunga. Pengguna dapat menginputkan nama bunga secara bertahap, dan proses input dihentikan ketika pengguna mengetikkan kata "SELESAI".

Variabel:

- pita: string yang menyimpan semua nama bunga.
- banyakBunga: penghitung jumlah bunga yang diinput.
- Perulangan: Program meminta pengguna memasukkan nama bunga.
- Jika pengguna mengetikkan "SELESAI", perulangan berhenti.
- Setiap bunga ditambahkan ke pita, dipisahkan oleh " - " jika lebih dari satu bunga.

Output:

Setelah input selesai, program menampilkan pita (rantai nama bunga) dan jumlah total bunga.

3. Soal Studi Case

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):

Source Code

```
package main

import (
    "fmt"
    // "math"
)

func main() {
    var beratKiri, beratKanan float64

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

        // Cek apakah salah satu kantong mencapai 9 kg atau lebih
        if beratKiri >= 9 || beratKanan >= 9 {
            break
        }
    }

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

```
// var beratKiri, beratKanan float64

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

//     // Cek apakah berat salah satu kantong negatif atau total lebih dari 150 kg
//     if beratKiri < 0 || beratKanan < 0 || (beratKiri+beratKanan) > 150 {
//         break
//     }

//     // Hitung selisih berat kantong kiri dan kanan
//     selisih := math.Abs(beratKiri - beratKanan)
//     oleng := selisih >= 9

//     fmt.Println("Sepeda motor pak Andi akan oleng:", oleng)
// }

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

Screenshoot Program

```
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided3(2B)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided3(2B)Modul2\Unguided3(2B)Modul2.go"
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.
```

Deskripsi Program

Program di atas adalah sebuah simulasi yang meminta pengguna untuk memasukkan berat belanjaan di dua kantong secara berulang. Program memiliki dua versi, tetapi masing-masing versi memiliki kondisi penghentian yang berbeda.

Program terus meminta pengguna memasukkan berat belanjaan di dua kantong (kantong kiri dan kantong kanan). Jika salah satu kantong memiliki berat 9 kg atau lebih, program akan berhenti. Setelah kondisi tersebut tercapai, program mencetak pesan "Proses selesai."

Source Code

```
package main

import (
    "fmt"
    "math"
)

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

    // for {
    //     fmt.Print("Masukan berat belanja di kedua kantong: ")
    //     fmt.Scan(&beratKiri, &beratKanan)

    //     // Cek apakah salah satu kantong mencapai 9 kg atau lebih
    //     if beratKiri >= 9 || beratKanan >= 9 {
    //         break
    //     }
    // }

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

    var beratKiri, beratKanan float64

    for {
        fmt.Print("Masukan berat belanja di kedua kantong: ")
        fmt.Scan(&beratKiri, &beratKanan)

        // Cek apakah berat salah satu kantong negatif atau total lebih dari 150 kg
        if beratKiri < 0 || beratKanan < 0 || (beratKiri+beratKanan) > 150 {
            break
        }

        // Hitung selisih berat kantong kiri dan kanan
        selisih := math.Abs(beratKiri - beratKanan)
        oleng := selisih >= 9

        fmt.Println("Sepeda motor pak Andi akan oleng:", oleng)
    }

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



Screenshoot Output

```
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.
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided3(2B)Modul2> |
```

Deskripsi Program

Penjelasan untuk Versi Kedua

Program meminta pengguna memasukkan berat belanjaan di dua kantong secara berulang. Kondisi untuk menghentikan program:

Jika berat salah satu kantong negatif (input tidak valid). Jika total berat kedua kantong lebih dari 150 kg. Program menghitung selisih berat antara kantong kiri dan kanan.

Jika selisih berat antara kantong kiri dan kanan 9 kg atau lebih, program akan mencetak bahwa "Sepeda motor Pak Andi akan oleng" (berisiko kehilangan keseimbangan).

Setelah salah satu kondisi berhenti terpenuhi (berat negatif atau lebih dari 150 kg), program akan mencetak "Proses selesai."

4. Soal Studi Case

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

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):

Source Code

```
package main

import (
    "fmt"
)

func main() {
    var K float64
    fmt.Print("Nilai K = ")
    fmt.Scan(&K)

    // Menghitung nilai f(K)
    numer := (4*K + 2) * (4*K + 2) // (4k + 2)^2
    denom := (4*K + 1) * (4*K + 3) // (4k + 1)(4k + 3)
    fK := numer / denom           // f(k) = numer / denom

    // Menampilkan hasil dengan pembatasan 11 angka desimal
    fmt.Printf("Nilai f(K) = %.10f\n", fK)
}
```

Screenshoot Output

```
Nilai K = 100
Nilai f(K) = 1.0000061880
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided4(2B)Modul2> |
```

Deskripsi Program

1. Program meminta pengguna memasukkan nilai `K`.
2. Program menghitung pembilang $(4K + 2)^2$.
3. Program menghitung penyebut $(4K + 1)(4K + 3)$.
4. Program membagi pembilang dengan penyebut untuk mendapatkan nilai $f(K)$.
5. Program mencetak hasil $f(K)$ dengan 10 angka desimal.

5. Soal Studi Case

PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program BlayaPos 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 int // Berat dalam gram

    fmt.Print("Berat parsel (gram): ")
    fmt.Scan(&berat)

    // Menghitung berat total dalam kg dan sisa gram
    kg := berat / 1000    // Berat dalam kilogram (bulat)
    sisaGram := berat % 1000 // Sisa berat dalam gram (kurang dari 1kg)

    // Biaya dasar (Rp 10.000 per kg)
    biayaKirim := kg * 10000

    var biayaTambahan int
    if kg > 10 {
        // Jika berat lebih dari 10 kg, sisa gram gratis
        biayaTambahan = 0
    } else {
        // Jika sisa gram >= 500, biaya Rp 5 per gram, jika < 500, Rp 15 per gram
        if sisaGram >= 500 {
            biayaTambahan = sisaGram * 5
        } else {
            biayaTambahan = sisaGram * 15
        }
    }

    // Menampilkan hasil
    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, sisaGram)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKirim, biayaTambahan)
    fmt.Printf("Total biaya: Rp. %d\n", biayaKirim+biayaTambahan)
}

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided1(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided1(2C)Modul2\main.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided1(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided1(2C)Modul2\main.go"
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided1(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided1(2C)Modul2\main.go"
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided1(2C)Modul2>

```

Deskripsi Program

Program ini menghitung biaya pengiriman parcel berdasarkan berat yang diinputkan oleh pengguna dalam satuan gram. Ada biaya dasar untuk berat dalam kilogram, serta biaya tambahan untuk sisa berat yang kurang dari 1 kg. Jika berat parcel lebih dari 10 kg, biaya tambahan untuk sisa gram ditiadakan.

- Program menghitung biaya pengiriman berdasarkan berat kilogram dan gram.
- Jika berat lebih dari 10 kg, biaya tambahan untuk gram tidak dikenakan.
- Sisa gram di bawah 10 kg dikenakan biaya tambahan yang berbeda tergantung jumlah gramnya.

5. Soal Studi Case

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 <= 80	AB
65 < NAM <= 72.5	B
57.5 < NAM <= 65	BC

$50 < \text{NAM} \leq 57.5$	C
$40 < \text{NAM} \leq 50$	D
$\text{NAM} \leq 40$	E

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

```
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"
    } else if nam >= 72.5 {
        nam = "AB"
    } else if nam >= 65 {
        nam = "B"
    } else if nam >= 57.5 {
        nam = "BC"
    } else if nam >= 50 {
        nam = "C"
    } else if nam >= 40 {
        nam = "D"
    } else {
        nam = "E"
    }
    fmt.Println("Nilai mata kuliah:", nmk)
}
```

- 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!

c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

Jawaban

a. Jika nilai nam diberikan adalah 80.1, program akan mengalami error **kompilasi** dan tidak akan berjalan sesuai spesifikasi. Hal ini terjadi karena pada bagian Program mencoba mengisi variabel nam (yang bertipe float64) dengan nilai berupa string seperti "A". Ini menyebabkan **incompatible type assignment**, karena variabel bertipe float64 tidak bisa menyimpan nilai bertipe string.

b. - Tipe data tidak sesuai: Variabel bertipe `float64` tidak bisa menyimpan nilai string.

- Variabel `nmk` tidak digunakan: Program seharusnya menggunakan variabel ini untuk menyimpan nilai huruf.

- Nilai tidak ditampilkan dengan benar: Output akan selalu kosong karena nilai huruf tidak pernah disimpan dengan benar.

Dengan memperbaiki tipe data dan memastikan logika `if-else` yang benar, program akan berfungsi sesuai dengan spesifikasi yang diinginkan.

c.

```
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"
    // } else if nam >= 72.5 {
    //     nam = "AB"
    // } else if nam >= 65 {
    //     nam = "B"
    // } else if nam >= 57.5 {
```



```

    //   nam = "BC"
    // } else if nam >= 50 {
    //   nam = "C"
    // } else if nam >= 40 {
    //   nam = "D"
    // } else {
    //   nam = "E"
    // }

    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)
}

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2C)Modul2\main.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2C)Modul2\main.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2C)Modul2\main.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided2(2C)Modul2> █

```

5. Soal Studi Case

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 (teks bergaris bawah adalah Input/read):

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

Source Code

```
package main

import (
    "fmt"
    "strconv"
)

func main() {
    for {
        var input string
        fmt.Print("Bilangan: ")
        fmt.Scanln(&input)

        if input == "Z" || input == "z" {
            break
        }

        b, err := strconv.Atoi(input)
        if err != nil || b <= 1 {
            fmt.Println("Masukkan bilangan bulat lebih besar dari 1!")
            continue
        }

        // Mencari dan mencetak faktor
        fmt.Print("Faktor: ")
        // jumlahFaktor := 0
    }
}
```

```

        for i := 1; i <= b; i++ {
            if b%i == 0 {
                fmt.Printf("%d ", i)
                // jumlahFaktor++
            }
        }
        // // Cek apakah bilangan prima
        // if jumlahFaktor == 2 {
        //     fmt.Println("Prima: true")
        // } else {
        //     fmt.Println("Prima: false")
        // }
        fmt.Println() // Cetak newline setelah semua faktor
        ditampilkan
    }
}

```

Screenshoot Output

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided3(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided3(2C)Modul2\main.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Bilangan: 7
Faktor: 1 7
Bilangan: 

```

Deskripsi Program

Program ini meminta input dari pengguna berupa bilangan bulat lebih besar dari 1, lalu menampilkan faktor-faktor dari bilangan tersebut. Jika pengguna memasukkan "Z" atau "z", program akan berhenti. Jika input tidak valid atau tidak memenuhi syarat (kurang dari atau sama dengan 1), program akan meminta input ulang.

1. **Fungsi Utama Program:** Program ini menerima input berupa bilangan bulat dari pengguna, lalu mencari dan menampilkan semua faktor bilangan tersebut. Program terus berjalan hingga pengguna memasukkan karakter "Z" atau "z" untuk keluar.
2. **Validasi Input:** Program memastikan bahwa input yang diberikan adalah bilangan bulat lebih besar dari 1. Jika input tidak valid atau berupa bilangan kurang dari atau sama dengan 1, pengguna diminta untuk memasukkan bilangan yang valid.

3. **Penggunaan Loop:** Program berjalan dalam loop hingga pengguna memilih untuk berhenti, memastikan pengguna dapat menghitung faktor dari beberapa bilangan secara berulang.
4. **Fleksibilitas Program:** Program fleksibel dengan memungkinkan input berupa string dan bilangan, serta memberikan pesan kesalahan yang jelas saat input tidak valid.
5. **Potensi Peningkatan:** Logika tambahan, seperti mengecek apakah bilangan prima, bisa diaktifkan untuk memperluas fungsionalitas program, tanpa mengubah alur utama.

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 (teks bergaris bawah adalah Input/read):Perhatikan contoh

Bilangan: <u>12</u>	Bilangan: <u>7</u>
Faktor: 1 2 3 4 6 12	Faktor: 1 7
Prima: false	Prima: true

Source Code

```
package main

import (
    "fmt"
    "strconv"
)

func main() {
    for {
        var input string
        fmt.Print("Bilangan: ")
        fmt.Scanln(&input)

        if input == "Z" || input == "z" {
            break
        }
    }
}
```

```

        b, err := strconv.Atoi(input)
        if err != nil || b <= 1 {
            fmt.Println("Masukkan bilangan bulat lebih besar
dari 1!")
            continue
        }

        // Mencari dan mencetak faktor
        fmt.Print("Faktor: ")
        jumlahFaktor := 0
        for i := 1; i <= b; i++ {
            if b%i == 0 {
                fmt.Printf("%d ", i)
                jumlahFaktor++
            }
        }
        // Cek apakah bilangan prima
        if jumlahFaktor == 2 {
            fmt.Println("\nPrima: true")
        } else {
            fmt.Println("\nPrima: false")
        }
        fmt.Println() // Cetak newline setelah semua faktor
ditampilkan
    }
}

```

Screenshoot Program

```

PS D:\TUGAS SEMESTER 3\Praktikum alpro 2\Unguided3(2C)Modul2> go run "d:\TUGAS SEMESTER 3\Prakti
kum alpro 2\Unguided3(2C)Modul2\main.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false

Bilangan: 7
Faktor: 1 7
Prima: true

```

Deskripsi Program

1. **Fungsi Utama:** Program ini meminta input berupa bilangan bulat dari pengguna, menampilkan faktor-faktor dari bilangan tersebut, dan menentukan apakah bilangan tersebut merupakan bilangan prima.

Program terus berjalan hingga pengguna memasukkan "Z" atau "z" untuk keluar.

2. **Validasi Input:** Program memastikan bahwa input yang diberikan adalah bilangan bulat lebih besar dari 1. Jika input tidak valid (misalnya huruf atau bilangan negatif), pengguna akan diberi pesan kesalahan dan diminta untuk memasukkan input yang benar.
3. **Pencarian Faktor dan Bilangan Prima:** Program menghitung faktor-faktor bilangan yang valid dan menggunakan jumlah faktor untuk memeriksa apakah bilangan tersebut prima (bilangan prima memiliki tepat 2 faktor: 1 dan bilangan itu sendiri).
4. **Looping Berkelanjutan:** Program berjalan dalam loop yang memungkinkan pengguna untuk memasukkan beberapa bilangan dan mendapatkan hasil faktor serta status prima hingga pengguna memilih untuk menghentikannya dengan memasukkan "Z" atau "z".

DAFTAR PUSTAKA

Coding Studio. (n.d.). *Golang adalah?* Coding Studio. Diakses pada 6 Oktober 2024, dari <https://codingstudio.id/blog/golang-adalah/>

Santekno. (n.d.). *Mengenal struktur control, perulangan, kondisi, dan map pada Golang.* Santekno. Diakses pada 6 Oktober 2024, dari <https://www.santekno.com/mengenal-struktur-control-perulangan-kondisi-dan-map-pada-golang/>