

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

Raihan Ramadhan/2311102040

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

Golang adalah bahasa pemrograman sumber terbuka (opensource), Golang (Go) disusun pada bulan September 2007 oleh Robert Griesemer, Rob Pike, dan Ken Thompson, semua kegiatannya dilakukan di Google, dan diumumkan pada bulan November 2009. Golang (Go) dapat berjalan pada Unix-like, Sistem Linux, FreeBSD, OpenBSD, Mac OS X, Plan 9, dan Microsoft Windows.

Go memiliki beberapa tipe data dasar, yaitu:

- Integer: Bilangan bulat seperti int8, int16, int32, dan int64.
- Floating-Point Number: Bilangan desimal seperti float32 dan float64.
- Complex Number: Bilangan kompleks.
- Booleans: Nilai boolean (true dan false).
- String: Urutan karakter dengan panjang tertentu

Kelebihan Go

Go memiliki kelebihan dibanding bahasa lainnya, beberapa di antaranya:

- Mendukung konkurensi di level bahasa dengan pengaplikasian cukup mudah
- Mendukung pemrosesan data dengan banyak prosesor dalam waktu yang bersamaan (*parallel processing*)
- Memiliki *garbage collector*
- Proses kompilasi sangat cepat
- Bukan bahasa pemrograman yang hirarkial dan bukan *strict OOP*, memberikan kebebasan ke developer perihal bagaimana cara penulisan kode.
- Dependensi dan *tooling* yang disediakan terbilang lengkap.
- Dukungan komunitas sangat bagus. Banyak tools yang tersedia secara gratis dan *open source* yang bisa langsung dimanfaatkan.

Kekurangan Golang

- Kurangnya Framework Web yang Kuat
- Kurangnya Fitur OOP yang Mendalam
- Kurangnya Generics Hingga Versi 1.18
- Kurangnya Penerimaan di Beberapa Industri

II. GUIDED

1. Guided 1

Source Code

```
package main

import "fmt"

func main() {

    var a, b, c, d, e int
    var hasil int
    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 Hasil

```
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1> go run
"
20 30 40 50 60
Hasil Penjumlahan  20 30 40 50 60 adalah =  200
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1> █
```

Deskripsi Program

Program diatas merupakan program aritmatika penjumlahan, dimana terdapat 5 variabel yang bertipe data integer. Variabel tersebut nantinya akan menyimpan nilai input dan terdapat variable hasil bertipe data integer untuk menyimpan hasil penjumlahan.

2. Guided 2

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 (teks bergaris bawah 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() {
    // 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")
}
}

```

Screenshot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
"
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:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

```

Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
BERHASIL : false
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan urutan warna sebanyak lima kali. Urutan warna yang harus dimasukkan adalah "merah", "kuning", "hijau", dan "ungu". Jika salah memasukkan urutan di salah satu percobaan, program akan langsung menghentikan proses dan menganggap percobaan

tersebut gagal. Jika semua percobaan berhasil dan urutan warna selalu benar, program akan menampilkan "BERHASIL : true". Namun, jika ada kesalahan dalam urutan warna, program akan menampilkan "BERHASIL : false". Program ini memeriksa apakah pengguna dapat memasukkan urutan warna dengan benar dalam lima kesempatan.

3. Guided 3

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 < NAM <= 57.5	C
40 < NAM <= 50	D
NAM <= 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'.

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)
}

```

Screenshot Output

a.

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
"
Masukkan nilai : 80.1
Nilai Indeks untuk nilai 80.10 adalah A
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

b.

```

Masukkan nilai : 93.5
Nilai Indeks untuk nilai 93.50 adalah A
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

```

Masukkan nilai : 70.6
Nilai Indeks untuk nilai 70.60 adalah C
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

```

Masukkan nilai : 49.5
Nilai Indeks untuk nilai 49.50 adalah E
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

Deskripsi Program

Program diatas merupakan program untuk menghitung nilai mata kuliah mahasiswa dengan mengkatagorikannya sesuai dengan soal.terdapt variable nam bertipe data float untuk menyimpan nilai numerik yang dimasukkan pengguna dan terdapat variable nmk bertipe data string yang digunakan untuk menyimpan hasil konversi nilai numerik menjadi nilai huruf.

III. UNGUIDED

4. Unguided 1

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):

N: <u>3</u>	N : <u>0</u>
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
    var pita []string

    // Membaca input jumlah bunga
    fmt.Print("Masukkan jumlah bunga: ")
    fmt.Scan(&N)

    reader := bufio.NewReader(os.Stdin)

    // Membersihkan newline yang tertinggal setelah input N
    reader.ReadString('\n')

    bungaKe := 1

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

        // Jika input adalah "SELESAI", hentikan proses input
        if strings.ToUpper(bunga) == "SELESAI" {
            break
        }

        // Tambahkan bunga ke pita
        pita = append(pita, bunga)
        bungaKe++
    }

    // Tampilkan pita (gabungan bunga dengan separator " - ")
    fmt.Println("Pita:", strings.Join(pita, " - "))

    // Tampilkan jumlah bunga dalam pita
    fmt.Printf("Jumlah bunga dalam pita: %d\n", len(pita))
}

```

Screenshoot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
go"
Masukkan jumlah bunga: 3
Bunga 1: Mawar
Bunga 2: Melati
Bunga 3: Anggrek
Pita: Mawar - Melati - Anggrek
Jumlah bunga dalam pita: 3
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
go"
Masukkan jumlah bunga: 3
Bunga 1: selesai
Pita:
Jumlah bunga dalam pita: 0
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

Deskripsi Program

Program ini meminta kamu untuk memasukkan nama-nama bunga dan menggabungkannya dalam satu rangkaian, dipisahkan oleh tanda strip. Ketika sudah selesai, cukup ketik selesai untuk mengakhiri prosesnya. Setelah itu, program akan menampilkan daftar bunga yang telah dimasukan dengan tampilan yang teratur. Program ini juga akan menghitung dan memberi tahu jumlah bunga yang ada dalam daftar tersebut. Intinya, program ini memudahkan kamu untuk membuat daftar nama bunga dengan cara yang sederhana.

5. Unguided 2

Source Code

```
package main

import (
    "fmt"
    "math"
)

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

    for {
        fmt.Print("\nMasukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): ")

        // Menggunakan Scanln untuk membaca input dengan spasi
        _, err := fmt.Scanln(&beratKiri, &beratKanan)
        if err != nil {
            fmt.Println("Input tidak valid. Silakan masukkan dua angka.")
            continue
        }

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

        // Tampilkan apakah motor Pak Andi akan oleng
        fmt.Printf("Sepeda motor Pak Andi akan oleng: %v\n", oleng)

        // Hitung total berat
        totalBerat = beratKiri + beratKanan

        // Jika total berat melebihi 150 kg atau salah satu berat negatif, hentikan program
        if totalBerat >= 150 || beratKiri < 0 || beratKanan < 0 {
            fmt.Println("Proses selesai.")
            break
        }
    }
}
```

Screenshoot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1> go run "d:\Kuliah\
go"

Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 5 10
Sepeda motor Pak Andi akan oleng: false

Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 55.6 70.2
Sepeda motor Pak Andi akan oleng: true

Masukkan berat belanjaan di kedua kantong (pisahkan dengan spasi): 70 90
Sepeda motor Pak Andi akan oleng: true
Proses selesai.

```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan berat barang belanjaan yang ada di dua kantong. Setelah berat tersebut dimasukkan, program akan menghitung perbedaan berat antara kedua kantong. Jika perbedaan beratnya lebih dari 9 kg, sepeda motor Pak Andi akan menjadi tidak stabil. Program ini akan terus meminta input sampai total berat kedua kantong melebihi 150 kg atau salah satu beratnya menjadi negatif. Setelah kondisi tersebut terpenuhi, program akan berhenti dan memberi tahu pengguna bahwa proses telah selesai.

6. Unguided 3

Diberikan sebuah persamaan sebagai berikut ini.

INFORMATIKA IQB

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

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 = <u>100</u> Nilai $f(K)$ = 1.0000061880

Source Code

```

package main

import (
    "fmt"
)

func hitungF(k float64) float64 {
    // Menghitung f(K) sesuai dengan persamaan yang diberikan
    atas := (4*k + 2) * (4*k + 2)
    bawah := (4*k + 1) * (4*k + 3)

```

```

    return atas / bawah
}

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

    fK := hitungF(k)
    fmt.Printf("Nilai f(K) = %.10f\n", fK)
}

```

Deskripsi Program

Program bertujuan untuk menghitung nilai $f(K)$ berdasarkan rumus yang menginputkan angka K oleh pengguna. Dalam fungsi `hitung F`, program melakukan perhitungan dengan K untuk menentukan nilai di bagian atas dan bawah, kemudian membagi hasil keduanya. Setelah perhitungan selesai, program akan menampilkan nilai $f(K)$ kepada pengguna. Saat dijalankan, program meminta pengguna untuk memasukkan nilai K .

7. Unguided 4

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 hitungBiaya(berat int) (int, string, string) {
    // Konversi berat dari gram ke kg dan sisa gram
    kg := berat / 1000
    sisaGram := berat % 1000

    // Biaya dasar per kilogram
    biayaKg := kg * 10000

    // Hitung biaya untuk sisa gram
    biayaSisaGram := 0
    if sisaGram > 0 {
        if sisaGram <= 500 {
            biayaSisaGram = sisaGram * 5
        } else {
            biayaSisaGram = sisaGram * 15
        }
    }

    // Jika total berat lebih dari 10 kg, biaya sisa gram tidak dikenakan
    if berat > 10000 {
        biayaSisaGram = 0
    }

    // Total biaya
    totalBiaya := biayaKg + biayaSisaGram

    // Format string untuk detail berat dan rincian biaya
    detailBerat := fmt.Sprintf("%d kg + %d gr", kg, sisaGram)
    detailBiaya := fmt.Sprintf("Rp. %d + Rp. %d", biayaKg, biayaSisaGram)
    return totalBiaya, detailBerat, detailBiaya
}

func main() {
    // Input berat parsel dalam gram
    var berat int

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

```

```

// Hitung biaya
totalBiaya, detailBerat, detailBiaya := hitungBiaya(berat)

// Tampilkan hasil
fmt.Println("Detail berat:", detailBerat)
fmt.Println("Rincian biaya:", detailBiaya)
fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}

```

Sreenshoot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
go"
Masukkan berat parsel (dalam gram): 1945
Detail berat: 1 kg + 945 gr
Rincian biaya: Rp. 10000 + Rp. 14175
Total biaya: Rp. 24175
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
nnerFile.go"
Masukkan berat parsel (dalam gram): 11945
Detail berat: 11 kg + 945 gr
Rincian biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

Deskripsi Program

Program ini adalah aplikasi sederhana yang menghitung biaya pengiriman berdasarkan berat parsel yang dimasukkan oleh pengguna. Pertama, berat parsel dikonversi dari gram ke kilogram dan sisa gram. Lalu, biaya dihitung berdasarkan Rp. 10.000 per kilogram, dengan tambahan biaya untuk sisa gram tergantung apakah berat sisanya lebih atau kurang dari 500 gram. Jika berat parsel lebih dari 10 kg, sisa gramnya tidak dikenakan biaya, program kemudian menampilkan rincian berat dan biaya.

8. Unguided 5

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: <u>12</u> Faktor: 1 2 3 4 6 12	Bilangan: <u>7</u> 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 (teks bergaris bawah adalah input/read):

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

Source Code

```
package main

import (
    "fmt"
    "strings"
)

// Fungsi untuk mencari faktor dari bilangan
func cariFaktor(b int) []int {
    var faktor []int
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            faktor = append(faktor, i)
        }
    }
    return faktor
}

// Fungsi untuk memeriksa apakah bilangan prima
func apakahPrima(b int) bool {
    if b < 2 {
        return false
    }
    for i := 2; i*i <= b; i++ {
        if b%i == 0 {
            return false
        }
    }
    return true
}
```



```

    return true
}

func main() {
    var b int

    // Meminta input dari user
    fmt.Print("Masukkan bilangan: ")
    fmt.Scan(&b)

    // Mencari faktor bilangan
    faktor := cariFaktor(b)

    // Mengubah slice faktor menjadi string dengan spasi sebagai pemisah
    faktorStr := strings.Trim(strings.Join(strings.Fields(fmt.Sprint(faktor)), " "), "[]")

    // Mencetak faktor tanpa tanda kurung
    fmt.Printf("Faktor: %s\n", faktorStr)

    // Menentukan apakah bilangan prima
    if apakahPrima(b) {
        fmt.Println("Prima: true")
    } else {
        fmt.Println("Prima: false")
    }
}

```

Screenshot Output

```

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
go"
Masukkan bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>
go"
Masukkan bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Kuliah\Matkul\SEMESTER 3\ALPRO PRAKTEK\Pertemuan 1>

```

Deskripsi Program

Program ini bertujuan untuk mencari faktor dari sebuah bilangan dan menentukan apakah bilangan tersebut termasuk bilangan prima. Pertama, program akan meminta pengguna memasukkan sebuah bilangan sebagai

input. Setelah itu, program akan menghitung dan menampilkan semua faktor dari bilangan tersebut. Jika bilangan tersebut hanya memiliki faktor 1 dan dirinya sendiri (bilangan prima), program akan menampilkan "Prima: true". Jika bukan bilangan prima, program akan menampilkan "Prima: false". Singkatnya, program ini mengecek faktor bilangan dan apakah bilangan tersebut prima atau tidak.