

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II  
REVIEW STRUKTUR KONTROL**



**Disusun Oleh :**

**Aji Noto Sutrisno (2311102262)**

**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

Struktur kontrol dari Go memiliki beberapa fungsi yaitu kondisional (switch), perulangan (loop) , dan percabangan (if), yang merupakan alat penting untuk membuat program yang dapat membuat keputusan, mengulangi tindakan, dan mengontrol aliran eksekusi.

### A. IF ( Percabangan )

Dalam Go if yang sederhana itu bentuknya seperti ini:

```
package main

import "fmt"

func main() {
    temperature := 75

    if temperature > 80 {
        fmt.Println("It's hot! Go swimming.")
    } else if temperature > 60 {
        fmt.Println("It's pleasant. Go for a hike.")
    } else {
        fmt.Println("It's cold. Stay inside.")
    }
}
```

Mari kita uraikan kodenya:

- Kita mendefinisikan sebuah variabel suhu dan memberikannya nilai 75.
- Kita menggunakan pernyataan “jika” untuk memeriksa apakah suhu lebih besar dari 80.
- Jika kondisinya benar, program akan mencetak “Panas! Pergilah berenang.”
- Jika kondisinya salah, kita beralih ke pernyataan “else if”. Kita memeriksa apakah suhu lebih besar dari 60.
- Jika kondisi “else if” benar, program akan mencetak “It's pleasant. Pergilah mendaki.”
- Jika kedua kondisi tersebut salah, blok “else” akan dieksekusi, mencetak “Ini dingin. Tetap di dalam.”

Pernyataan “if”, “else if”, dan “else” membantu bot cuaca kita membuat keputusan berdasarkan suhu!

## B. For ( Perulangan )

Pengulangan for pada Go mirip tapi tidak sama dengan C. Ia menggabungkan *for* dan *while* dan tidak ada *for-while*.

```
package main

import "fmt"

func main() {
    for day := 1; day <= 5; day++ {
        fmt.Printf("Day %d: Weather forecast goes here!\n", day)
    }
}
```

Output kode ini akan mencetak output berikut:

```
Day 1: Weather forecast goes here!
Day 2: Weather forecast goes here!
Day 3: Weather forecast goes here!
Day 4: Weather forecast goes here!
Day 5: Weather forecast goes here!
```

Mari kita uraikan perulangan “for”:

- Kita menginisialisasi variabel hari dengan nilai 1.
- Perulangan terus berlanjut selama hari kurang dari atau sama dengan 5.
- Setelah setiap iterasi, kita menambah hari dengan 1 menggunakan pernyataan `day++`.
- Di dalam perulangan, kita mencetak pesan dengan hari, nomor cuaca saat ini.

## C. Switch ( Menu )

Berikut ini contoh penggunaan pernyataan “switch”:

```
package main

import "fmt"

func main() {
    weatherCode := 3
    switch weatherCode {
    case 1:
```

```
        fmt.Println("Sunny")
    case 2:
        fmt.Println("Partly cloudy")
    case 3:
        fmt.Println("Cloudy")
    case 4:
        fmt.Println("Rainy")
    case 5:
        fmt.Println("Snowy")
    default:
        fmt.Println("Unknown weather code")
    }
}
```

Pada output kode ini, kita memiliki variabel `weatherCode` dengan nilai 3. Pernyataan “switch” akan memeriksa nilai `weatherCode` dan mengeksekusi kasus yang cocok. Jika tidak ada yang cocok, maka kasus “default” akan dieksekusi.

Untuk contoh ini, outputnya adalah:

```
Cloudy
```

Pernyataan “switch” memudahkan untuk menangani berbagai kondisi dengan cara yang bersih dan mudah dibaca.

## II. GUIDED

### 1. Soal Studi Case

#### Sourcecode

```
package main

import
    "fmt"

func main() {

    nama := "Aji Noto Sutrisno"
    fmt.Print(nama)
}
```

#### Output

```
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 4\main.go"
Aji Noto Sutrisno
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4>
```

#### Deskripsi

Program ini dibuat untuk menampilkan sebuah nama mahasiswa, Program ini memiliki algoritma :

- Inisialisasi nama = aji noto sutrisno
- Lalu tampilkan nama

### 2. 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):

```
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
```

### Sourcecode

```
package main

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

func main() {
    correctOrder := []string{"merah", "kuning", "hijau",
    "ungu"}

    reader := bufio.NewReader(os.Stdin)
    success := true

    for i := 1; i <= 5; i++ {
        fmt.Print("Percobaan ", i, " : ")
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)

        colors := strings.Split(input, " ")

        for j := 0; j < 4; j++ {
            if colors[j] != correctOrder[j] {
                success = false
                break
            }
        }
        if !success {
            break
        }
    }

    if success {
```

```

        fmt.Println("Berhasil : true")
    }else {
        fmt.Println("Berhasil : false")
    }
}

```

## Screenshoot Output

```

PS D:\Semester 3\P Alpro\Laparak 1\Unguided 2B.1> go run "d:\Semester 3\P Alpro\Laparak 1\Unguided 2B.1\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:\Semester 3\P Alpro\Laparak 1\Unguided 2B.1> go run "d:\Semester 3\P Alpro\Laparak 1\Unguided 2B.1\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:\Semester 3\P Alpro\Laparak 1\Unguided 2B.1>

```

## Deskripsi Program

Program ini dibuat untuk mencocokkan inputan *user* dengan array yang sudah ada,

- Inisialisasi slice/array *CorrectOrder*, *reader*, *succes*, *colors*
- *Bufio* digunakan untuk memperpanjang input
- *Strings.TrimSpace* digunakan untuk menginputkan string dengan mengabaikan spasi
- Program melakukan perulangan untuk mencocokkan inputan user (*colors[]*) dengan (*CorrectOrder[]*)
- Jika tidak cocok program akan selesai

Cara kerja dari program ini adalah *user* menginputkan 4 warna, kemudia program akan mencocokkan inputan *user* dengan data yang sudah ada, apabila cocok akan keluar *output* "Berhasil : true" begitu juga sebaliknya

## 3. Soal Studi Case

### Sourcecode

```

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)
}

```

## Screenshoot Output

```
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 4\main.go"
1 2 3 4 5
Hasil Penjumlahan 1 2 3 4 5 adalah 15
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> █
```

## Deskripsi Program

Program diatas dibuat untuk perhitungan penjumlahan dengan lima inputan dari *user*. Algoritma dari program diatas :

- Inisialisasi variable a, b, c, d, e, hasil dengan tipe data *integer*
- *User* menginputkan 5 buah Bilangan bulat real
- Hasil = a + b + c + d + e
- Kemudian Print "Hasil"

Cara kerja Program diatas adalah *user* dapat menginputkan lima buah inputan numerik, kemudian program akan mengeksekusi dengan sebuah fungsi penjumlahan sederhana, kemudian akan menampilkan output "Hasil Penjumlahan 1 2 3 4 5 adalah 15"

## 4. 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 < 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.

## Sourcecode

```
package main

import "fmt"

func main() {
    var nam float32
    var nmk string
    fmt.Print("Masukkan Nilai : ")
    fmt.Scan(&nam)
```



```

    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"
    }

    fmt.Printf("Nilai Indeks untuk nilai %.2f adalah\n", nam, nmk)
    //%.2f merupakan batasan dibelakang koma
}

```

### Screenshoot Output

```

PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 4\main.go"
Masukkan Nilai : 92
Nilai Indeks untuk nilai 92.00 adalah A
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4>

```

### Deskripsi Program

Program ini dibuat untuk membantu memilah sebuah nilai akhir dengan cara mengategorikan setiap nilai, Algoritmanya :

- Inisialisasi variable nam float32, nmk string
- User input
- Pemilahan inputan user dengan berbagai kondisi
- Output nilai akhir beserta gradenya

Cara kerja program ini adalah *user* menginputkan sebuah nilai random kemudian program akan menentukan inputan tersebut memiliki grade A-E, lalu program akan menampilkan output ”

### III. UNGUIDED

#### 1. Soal Studi Case

Buatlah sebuah Program yang menerima input bilangan bulat positif ( $\neq 0$ )

- N merupakan sebuah perulangan sebanyak N kali dan disimpan ke dalam Pita
- Jika user mengetikkan SELESAI. Kemudian tampilkan isi Pita beserta banyaknya bunga

#### Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var (
        N      = 10
        bunga string
        Pita   []string
    )
    //fmt.Print("N : ") //input awal yang tidak terpakai
    //fmt.Scanln(&N)     //input awal yang tidak terpakai
    for i := 1; i <= N; i++ {
        fmt.Printf("Bunga %d: ", i)
        fmt.Scanln(&bunga)
        if strings.EqualFold(bunga, "Selesai") {
            break // Equafold digunakan untuk
                mencocokkan antara inputan user dengan selesai tanpa
                memperhatikan huruf kapital atau tidak
        }
        Pita = append(Pita, bunga) //append berfungsi
        untuk menginput suatu nilai kedalam array
    }
    temp := strings.Join(Pita, " - ") //digunakan untuk
    menggabungkan nilai array/slice kemudian diberikan
    tambahab "-"
    fmt.Println("Pita : " + temp + " - ")
    fmt.Print("Bunga : ", len(Pita))
}
```

## Screenshoot Output

### a. Program a

```
PS D:\Semester 3\P Alpro\Laprak 1\Unguided 2B.2> go run "d:\Semester 3\P Alpro\Laprak 1\Unguided 2B.2\main.go"
N : 3
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Pita : Kertas - Mawar - Tulip -
PS D:\Semester 3\P Alpro\Laprak 1\Unguided 2B.2> go run "d:\Semester 3\P Alpro\Laprak 1\Unguided 2B.2\main.go"
N : 0
Pita : -
PS D:\Semester 3\P Alpro\Laprak 1\Unguided 2B.2> █
```

### b. Program b

```
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 1> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 1\main.go"
Bunga 1: Kertas
Bunga 2: Mawar
Bunga 3: Tulip
Bunga 4: Selesai
Pita : Kertas - Mawar - Tulip -
Bunga : 3
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 1> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 1\main.go"
Bunga 1: selesai
Pita : -
Bunga : 0
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 1> █
```

## Deskripsi Program

Program ini dibuat untuk digunakan untuk menambahkan data bunga kedalam *array* **Pita**, kemudian program akan menghitung isi dari *array* tersebut.

Algoritmanya :

- Inisialisasi variable **N** = 10/ **N** inputan *user*, **bunga** *string*, *array/slice* **Pita**
- `for i := 1; i <= N; i++`, digunakan untuk melakukan perulangan sebanyak **N**
- Kemudian *user* menginputkan nama-nama **bunga** yang di inginkan
- Program akan meminta terus menerus inputan sebanyak **N**, tetapi `if strings.EqualFold(bunga, "Selesai")` Program akan berhenti
- `EqualFold` adalah sebuah inputan *user* tanpa memperdulikan besar kecilnya huruf
- `Pita = append(Pita, bunga)`, `Append` digunakan untuk menginput data kedalam *array*
- `temp := strings.Join(Pita, " - ")`, `join` digunakan untuk menampilkan semua nilai pada array dan diberi “-“
- Kemudian menampilkan data *array* dengan panjang *array* `fmt.Print("Bunga : ", len(Pita))`

Cara kerja program adalah user menginputkan nama bunga lalu program akan memindah data bunga ke array **Pita** dan dilakukan terus menerus sejumlah panjang array, akan tetapi jika user menginputkan “SELESAI” program akan berhenti berjalan.

## 2. Soal Studi Case

Buatlah sebuah Program yang menerima dua input bilangan real positif untuk menentukan berat suatu benda

- a. Melakukan looping terus menerus, program akan berhenti dengan sebuah kondisi yaitu, jika selisih kedua inputan tidak lebih dari 9kg, dan jika salah satu dari kedua inputan ada yang berisi lebih dari 9kg maka program akan berhenti.
- b. Modifikasi program a, program akan menampilkan nilai 'true' jika selisih kedua isi lebih dari atau sama dengan 9kg. Program akan berhenti apabila total berat kedua inputan melebihi 150kg, atau salah satu kantong berisi negatif

### Sourcecode

```
package main

import "fmt"

func main() {
    var kiri, kanan float32

    for {
        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
        fmt.Scanln(&kiri, &kanan)
        /* if a == 9 || b == 9 {
            break
        } else if a-b >= 9 {
            break
        }*/ // modifikasi dari A

        if kiri < kanan {
            kiri, kanan = kanan, kiri
            //kondisi menukar nilai inputan agar tidak negatif
        }

        if kiri+kanan >= 150 || kiri < 0 || kanan < 0
        {
            break
        }
        oleng := kiri-kanan > 9
        fmt.Printf("Sepeda motor pak Andi akan oleng : %v\n", oleng)
    }
    fmt.Println("Proses Selesai.")
}
```

## Screenshoot Output

### a. Program a

```
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 2> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 2\main.go"
Masukkan berat belanjaan di kedua kantong: 5.5 1.0
Masukkan berat belanjaan di kedua kantong: 7.1 8.5
Masukkan berat belanjaan di kedua kantong: 2 6
Masukkan berat belanjaan di kedua kantong: 9 5.8
Proses Selesai.
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 2> █
```

### b. Program b

```
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 2> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 2\main.go"
Masukkan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng : false
Masukkan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng : true
Masukkan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng : false
Masukkan berat belanjaan di kedua kantong: 59.5 98.7
Proses Selesai.
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 2> █
```

## Deskripsi Program

Program ini dibuat untuk menampilkan apakah berat disetiap kantong itu tidak memiliki selisih yang jauh, karena jika selisihnya jauh maka akan oleng, Algoritmanya :

- Insialisasi Variable **kiri**, **kanan** float
- Program akan melakukan perulangan secara terus menerus apabila tidak memenuhi kondisi `if kiri+kanan >= 150 || kiri < 0 || kanan < 0`

Cara kerjanya program akan meminta dua inputan untuk kantong kiri dan kanan, kemudian program akan menentukan apakah memenuhi kondisi untuk berhenti, apabila memenuhi kondisinya, program akan berhenti dan jika tidak memenuhi program akan terus menerus meminta inputan dari *user*

## 3. Soal Studi Case

Buatlah sebuah Program dengan inputan sebuah bilangan K, kemudian menghitung nilai  $f(K)$  sesuai dengan persamaan fungsi berikut

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

## Sourcecode

```
package main

import "fmt"

func main() {
```

```

var K float64

fmt.Print("Nilai : ")
fmt.Scanln(&K)

atas := ((4*K+2)*(4*K+2))
bawah := ((4*K+1)*(4*K+3))

f := atas / bawah

fmt.Printf("Nilai f(k) = %.10f", f)
//%.10f digunakan untuk membatasi panjang output hasil
}

```

### Screenshoot Output

```

PS D:\Pratikum Alpro 2\Laprak 2\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 3\main.go"
Nilai : 100
Nilai f(k) = 1.0000061880
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 3> █

```

### Deskripsi Program

Program ini dibuat untuk menghitung sebuah fungsi dengan inputan dari *user*, Algoritmanya :

- Inisialisasi variable K dan `atas := ((4*K+2)*(4*K+2))`; `bawah := ((4*K+1)*(4*K+3))`; `f := atas / bawah`
- `fmt.Printf("Nilai f(k) = %.10f", f)` digunakan untuk output yang dibatasi sepanjang 10

Cara kerjanya adalah *user* menginputkan sebuah nilai, kemudian program mengeksekusi dengan fungsi yang ada, lalu mengembalikan outputnya “Nilai f(k) = 1.0000061880”

### 4. Soal Studi Case

Buatlah sebuah Program yang dimana program digunakan untuk menghitung berat (dalam gram), dan total berat dalam kg dan sisanya dalam gram. Biaya pengiriman memiliki beberapa kondisi,

- Rp. 10.000,- per kg.
- Jika > 500 gram, maka biaya kirim Rp. 5,- per gram
- Jika < 500 gram, maka biaya kirim Rp. 15,- per gram
- Sisa berat < 1000 gram akan digratiskan, apabila total berat > 10kg

### Sourcecode

```
package main
```

```

import (
    "fmt"
)

func main() {
    var (
        totalBerat int
        sisaHarga   int
        hargaPerKg  = 10000
        totalBiaya  int
    )

    fmt.Print("Masukkan berat persel (gram): ")
    fmt.Scanln(&totalBerat)

    kg := totalBerat / 1000
    gram := totalBerat % 1000
    fmt.Printf("Detail berat: %d kg + %d gram\n", kg,
gram)

    biayaKg := kg * hargaPerKg

    if gram >= 500 {
        sisaHarga = gram * 5
    } else {
        sisaHarga = gram * 15
    }

    if totalBerat > 10000 && gram <= 1000 {
        sisaHarga = 0
    }

    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n",
biayaKg, sisaHarga)
    totalBiaya = biayaKg + sisaHarga
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}

```

## Screenshoot Output

```

PS D:\Pratikum Alpro 2\Laprak 2\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 3\main.go"
Masukkan berat persel (gram): 8250
Detail berat: 8 kg + 250 gram
Detail biaya: Rp. 80000 + Rp. 3750
Total biaya: Rp. 83750
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 3\main.go"
Masukkan berat persel (gram): 9250
Detail berat: 9 kg + 250 gram
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 3> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 3\main.go"
Masukkan berat persel (gram): 11750
Detail berat: 11 kg + 750 gram
Detail biaya: Rp. 110000 + Rp. 3750
Total biaya: Rp. 110000
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 3> █

```

### Deskripsi Program

Program ini dibuat untuk menentukan konversi gram ke kg dan sisa gram, kemudian program akan memberikan sebuah harga disetiap kg, gram. Algoritmanya :

- Inisialisasi variable `totalBerat int; sisaHarga int ; hargaPerKg = 10000 ; totalBiaya int`
- `kg := totalBerat / 1000` mengambil kg
- `gram := totalBerat % 1000` Modulus, mengambil nilai sisa
- `biayaKg := kg * hargaPerKg` Menghitung total biaya perkg
- `if totalBerat > 10000 && gram <= 1000` Menentukan diskon

Cara kerjanya, user menginputkan total berat yang dimiliki, kemudian program akan membaginya menjadi kg dan gram, kemudian akan memiliki harga yang sesuai dengan beratnya, dan apabila memenuhi kondisi diskon akan diberikan diskon sisa yaitu kurang dari 1000 gram

### 5. Soal Studi Case

Buatlah sebuah Program yang menginputkan bilangan bulat  $b$  memiliki faktor bilangan  $f > 0$

- a. Inputan bilangan bulat  $b$  dan  $b > 1$ , Program akan menampilkan faktor dari  $b$
- b. Lanjutkan program a setelah menerima masukkan bilangan bulat  $b > 0$ . program akan menampilkan faktor dari  $b$ , dan kemudian program akan menentukan apakah  $b$  bilangan prima.

### Sourcecode

```
package main

import "fmt"

func main() {

    var nilai int
    var faktor []int

    fmt.Print("Bilangan: ")
    fmt.Scanln(&nilai)
    if nilai > 0 {
        for i := 1; i <= nilai; i++ {
            if nilai%i == 0 {
                faktor = append(faktor, i)
                //append input nilai array
            }
        }
    }
}
```



```

        fmt.Print("Faktor: ")
        for j := 0; j < len(faktor); j++ {

            fmt.Print(faktor[j], " ")

        }
        fmt.Print("\n")
        if len(faktor) == 2 {
            //len adalah panjang array

            fmt.Print("Prima: true")
        } else {
            fmt.Print("Prima: false")
        }
    }
}

```

## Screenshoot Output

### a. Program a

```

PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 4\main.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 4\main.go"
Bilangan: 7
Faktor: 1 7
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> █

```

### b. Program b

```

PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 4\main.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> go run "d:\Pratikum Alpro 2\Laprak 2\Unguided 4\main.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\Pratikum Alpro 2\Laprak 2\Unguided 4> █

```

## Deskripsi Program

Program ini dibuat untuk menentukan faktor dari bilangan dan bilangan prima.

Algoritmanya :

- Inisialisasi sebuah variable `var nilai int; var faktor []int`
- Melakukan Perulangan `for i := 1; i <= nilai; i++` panjangnya sesuai nilai inputan
- Kemudian melakukan perulangan untuk menampilkan nilai array `for j := 0; j < len(faktor); j++`
- `if len(faktor) == 2` Digunakan untuk menentukan kondisi apakah nilai tersebut merupakan bilangan prima

Cara kerjanya user memasukan inputan bilangan kemudian program akan menjalankan fungsi, fungsi yang ada seperti perulangan untuk membagi, lalu percabangan yang digunakan untuk menentukan apakah bilangan prima

#### **IV. DAFTAR PUSTAKA**

T. TowardsDev, "Golang 101: Control Structures, Conditionals, Loops, and Branching," *Towards Dev*, Sep. 25, 2023.  
<https://towardsdev.com/golang-101-control-structures-conditionals-loops-and-branching-d127f760c2f3>.