

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



Disusun Oleh:

NAMA : Titanio Francy Naddiansa

NIM: 2311102289

KELAS: IF-11-07

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO**

2024

I. DASAR TEORI

Golang, atau Go, adalah bahasa pemrograman open source yang dirancang untuk memudahkan pengembangan perangkat lunak yang sederhana, andal, dan efisien. Bahasa ini dikembangkan pada September 2007 oleh Robert Griesemer, Rob Pike, dan Ken Thompson di Google, dan secara resmi diumumkan pada November 2009. Golang dibuat dengan tujuan untuk mempercepat proses kompilasi dan eksekusi, sekaligus mempermudah penulisan kode yang efektif. Bahasa ini memiliki kesamaan dengan C dalam hal sintaksis dan pengkodean. Sebagai proyek open source, kode sumber untuk kompiler, pustaka, dan alat-alatnya tersedia secara bebas untuk umum. Golang dapat dijalankan di berbagai sistem operasi, termasuk Unix-like, Linux, FreeBSD, OpenBSD, macOS, Plan 9, dan Windows. Popularitas Go saat ini meningkat karena banyak programmer menyukai kesederhanaannya dan kecepatannya yang lebih unggul dibandingkan beberapa bahasa pemrograman lain.

Karakteristik Go Language

Go memiliki sejumlah fitur utama yang menjadikannya unik dan ideal untuk berbagai kasus pemrograman:

- **Tipe Statis dan Dikompilasi:** Go adalah bahasa pemrograman yang menggunakan tipe statis, di mana tipe data variabel ditentukan saat kompilasi. Hal ini membantu mendeteksi kesalahan lebih awal dan meningkatkan kinerja karena kode dioptimalkan pada saat kompilasi.
- **Bahasa Pemrograman Sistem:** Go dirancang sebagai bahasa pemrograman sistem, serupa dengan C dan C++ dalam hal pengendalian memori dan kemampuan menghasilkan program yang berkinerja tinggi.
- **Model Concurrency (Goroutines):** Salah satu keunggulan utama Go adalah dukungannya terhadap model concurrency melalui Goroutines. Goroutines adalah thread ringan yang memungkinkan program dijalankan secara bersamaan dengan lebih efisien, sangat bermanfaat pada sistem dengan banyak inti CPU.
- **Garbage Collection:** Go memiliki sistem garbage collection otomatis yang menangani pengelolaan memori, sehingga pengembang tidak perlu khawatir mengalokasikan dan melepaskan memori secara manual.
- **Cross-Platform:** Program yang ditulis dalam Go dapat dikompilasi dan dijalankan di berbagai platform seperti Windows, macOS, dan Linux tanpa memerlukan perubahan pada kode sumber.

II. GUIDED

1. Source Code

```
package main

import "fmt"

func main() {

    var nama string = "menggokil"

    var umur int = 20

    var tinggi float64 = 175.5

    var isSunny bool = false

    var inisial rune = 'A'

    fmt.Println("Nama:", nama)

    fmt.Println("Umur:", umur)

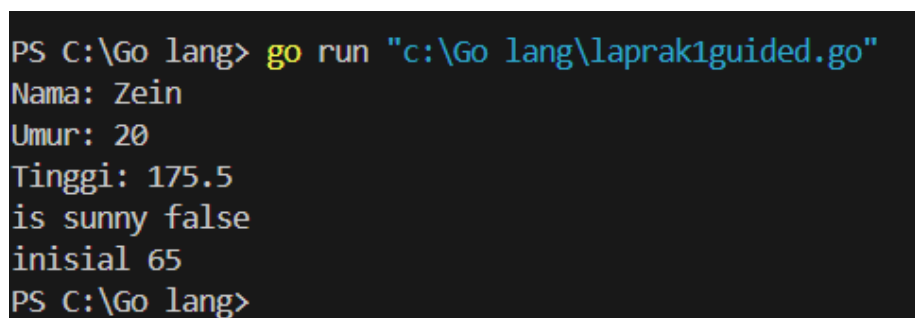
    fmt.Println("Tinggi:", tinggi)

    fmt.Println("is sunny", isSunny)

    fmt.Println("inisial", inisial)

}
```

Screenshoot Output



```
PS C:\Go lang> go run "c:\Go lang\laprak1guided.go"
Nama: Zein
Umur: 20
Tinggi: 175.5
is sunny false
inisial 65
PS C:\Go lang>
```

Deskripsi Program

Program di atas mendeklarasikan lima variabel dengan tipe data berbeda: string, int, float64, bool, dan rune. Variabel-variabel tersebut disimpan dengan nilai seperti nama ("Zein"), umur (20), tinggi (175.5), status cuaca (false), dan inisial

('A'). Kemudian, program mencetak nilai-nilai ini ke layar menggunakan fungsi `fmt.Println()`

2. Source Code

```
package main
```

```
import (  
    "fmt"  
)
```

```
func cekTahunKabisat(tahun int) bool {  
    if tahun%400 == 0 {  
        return true  
    } else if tahun%100 == 0 {  
        return false  
    } else if tahun%4 == 0 {  
        return true  
    } else {  
        return false  
    }  
}
```

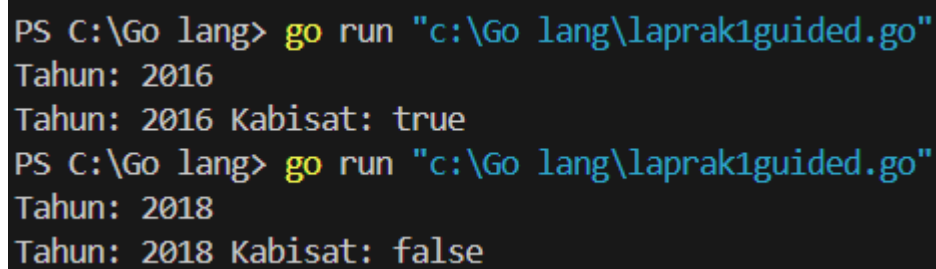
```
func main() {  
    var tahun int  
    fmt.Print("Tahun: ")  
    fmt.Scan(&tahun)
```

```
kabisat := cekTahunKabisat(tahun)

fmt.Printf("Tahun: %d Kabisat: %t\n", tahun, kabisat)

}
```

Screenshoot Output



```
PS C:\Go lang> go run "c:\Go lang\laprak1guided.go"
Tahun: 2016
Tahun: 2016 Kabisat: true
PS C:\Go lang> go run "c:\Go lang\laprak1guided.go"
Tahun: 2018
Tahun: 2018 Kabisat: false
```

Deksripsi Program

Program di atas memeriksa apakah suatu tahun merupakan tahun kabisat. user memasukkan tahun, lalu program memeriksa tahun tersebut menggunakan fungsi cekTahunKabisat(). Tahun kabisat adalah tahun yang habis dibagi 400, atau habis dibagi 4 tapi tidak habis dibagi 100. Hasilnya dicetak sebagai true (kabisat) atau false (bukan kabisat).

3. Source Code

```
package main

import "fmt"

func main() {

    var temperaturCelsius float64

    fmt.Print("Temperatur Celsius: ")

    fmt.Scanln(&temperaturCelsius)
```

```

var temperaturFahrenheit float64 = (temperaturCelsius * 9 / 5) + 32

var temperaturReamur float64 = temperaturCelsius * 4 / 5

var temperaturKelvin float64 = temperaturCelsius + 273.15

fmt.Printf("Derajat Fahrenheit: %.0f\n", temperaturFahrenheit)

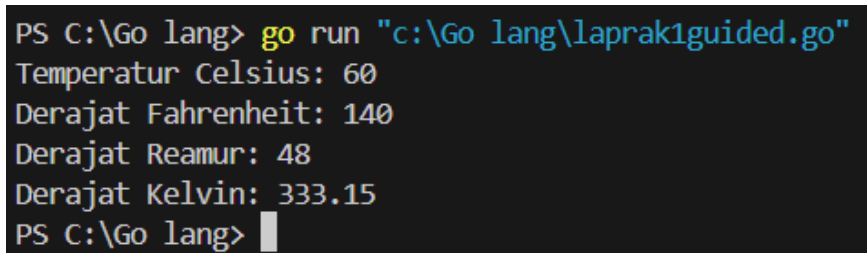
fmt.Printf("Derajat Reamur: %.0f\n", temperaturReamur)

fmt.Printf("Derajat Kelvin: %.2f\n", temperaturKelvin)

}

```

Screenshoot Output



```

PS C:\Go lang> go run "c:\Go lang\laprak1guided.go"
Temperatur Celsius: 60
Derajat Fahrenheit: 140
Derajat Reamur: 48
Derajat Kelvin: 333.15
PS C:\Go lang>

```

Deskripsi Program

Program di atas mengonversi suhu dari Celsius ke Fahrenheit, Reamur, dan Kelvin. user diminta memasukkan suhu dalam Celsius menggunakan `fmt.Scanln()`, kemudian program menghitung konversinya dengan rumus: untuk Fahrenheit $(\text{Celsius} * 9 / 5) + 32$, untuk Reamur $\text{Celsius} * 4 / 5$, dan untuk Kelvin $\text{Celsius} + 273.15$. Hasil konversi ditampilkan dengan format desimal yang sesuai menggunakan `fmt.Printf()`.

4. Source Code

```

package main

import "fmt"

func main() {

    var (

        satu, dua, tiga string

        temp string

```

)

```
fmt.Print("Masukan input string: ")
```

```
fmt.Scanln(&satu)
```

```
fmt.Print("Masukan input string: ")
```

```
fmt.Scanln(&dua)
```

```
fmt.Print("Masukan input string: ")
```

```
fmt.Scanln(&tiga)
```

```
fmt.Println("Output awal = " + satu + " " + dua + " " + tiga)
```

```
temp = satu
```

```
satu = dua
```

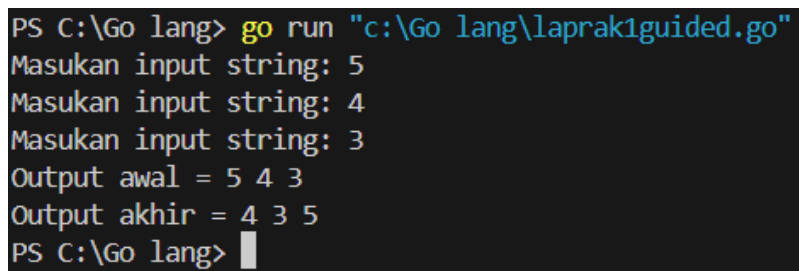
```
dua = tiga
```

```
tiga = temp
```

```
fmt.Println("Output akhir = " + satu + " " + dua + " " + tiga)
```

```
}
```

Screenshoot output



```
PS C:\Go lang> go run "c:\Go lang\laprak1guided.go"
Masukan input string: 5
Masukan input string: 4
Masukan input string: 3
Output awal = 5 4 3
Output akhir = 4 3 5
PS C:\Go lang>
```

Deskripsi Program

Program di atas meminta tiga input string dari user menggunakan `fmt.Scanln()` dan menampilkannya dalam urutan awal. Setelah itu, program menukar posisi string dengan menggunakan variabel sementara `temp`. Nilai satu dipindahkan ke `temp`, dua dipindahkan ke satu, tiga dipindahkan ke dua, dan akhirnya nilai di `temp` dipindahkan ke tiga. Setelah penukaran, program menampilkan kembali string dengan urutan yang sudah diubah.

III. UNGUIDED

2b

1. Source Code

```
package main
```

```
import (  
    "fmt"  
)
```

```
type DataEksperimen [5][4]string
```

```
func InputDataEksperimen(data *DataEksperimen) {  
    for i := 0; i < 5; i++ {  
        fmt.Printf("Percobaan %d:\n", i+1)  
        for j := 0; j < 4; j++ {  
            fmt.Scan(&data[i][j])  
        }  
    }  
}
```

```
func IsValid(data DataEksperimen, warnaYangDiHarapkan [4]string) bool {  
    for i := 0; i < 5; i++ {  
        for j := 0; j < 4; j++ {  
            if data[i][j] != warnaYangDiHarapkan[j] {  
                return false  
            }  
        }  
    }  
}
```



```

    return true
}

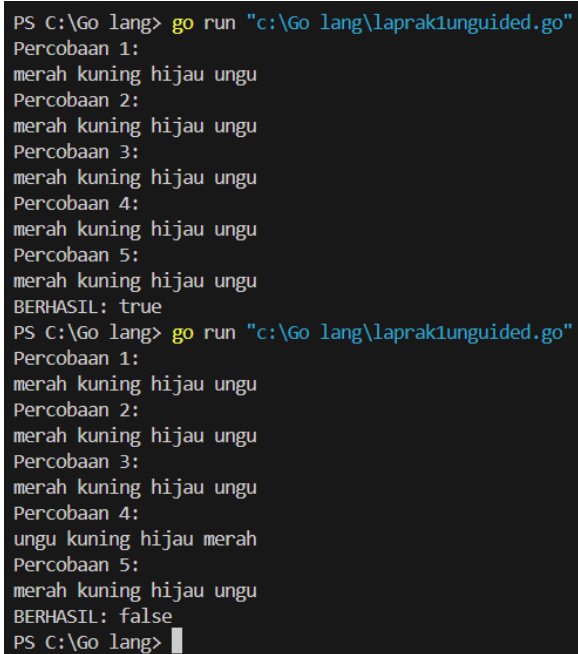
func PrintHasil(isValid bool) {
    if isValid {
        fmt.Println("BERHASIL: true")
    } else {
        fmt.Println("BERHASIL: false")
    }
}

func main() {
    var dataEksperimen DataEksperimen
    warnaYangDiHarapkan := [4]string{"merah", "kuning", "hijau", "ungu"}

    InputDataEksperimen(&dataEksperimen)
    isValid := IsValid(dataEksperimen, warnaYangDiHarapkan)
    PrintHasil(isValid)
}

```

Screenshot Output



```

PS C:\Go lang> go run "c:\Go lang\laprak1unguided.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 C:\Go lang> go run "c:\Go lang\laprak1unguided.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
Percobaan 5:
merah kuning hijau ungu
BERHASIL: false
PS C:\Go lang>

```

Deskripsi Program

Program ini menerima input 5 percobaan berupa 4 warna, kemudian memvalidasi apakah urutan warna pada setiap percobaan sesuai dengan urutan yang diharapkan (merah, kuning, hijau, ungu) dan menampilkan hasil "BERHASIL: true" jika semua percobaan valid, atau "BERHASIL: false" jika ada yang tidak sesuai.

2. Source Code

```
package main
import (
    "fmt"
    "strings"
    "os"
)

func main() {
    var n int
    var bunga, pita string
    var count int

    fmt.Fprint(os.Stdout, "Masukkan jumlah bunga (N): ")
    fmt.Fscan(os.Stdin, &n)

    if n == 0 {
        fmt.Fprintln(os.Stdout, "Pita: ")
        fmt.Fprintln(os.Stdout, "Bunga: 0")
        return
    }

    for count < n {
        count++
        fmt.Fprintf(os.Stdout, "Bunga %d: ", count)
        fmt.Fscan(os.Stdin, &bunga)

        if strings.EqualFold(bunga, "SELESAI") {
            count--
            break
        }

        if len(pita) > 0 {
            pita += " - "
```

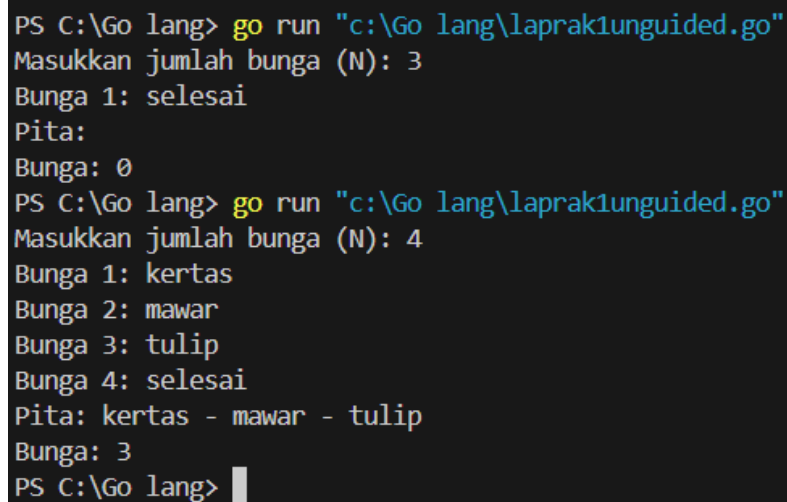
```

    }
    pita += bunga
}

fmt.Fprintf(os.Stdout, "Pita: %s\n", pita)
fmt.Fprintf(os.Stdout, "Bunga: %d\n", count)
}

```

Screenshot Output



```

PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Masukkan jumlah bunga (N): 3
Bunga 1: selesai
Pita:
Bunga: 0
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Masukkan jumlah bunga (N): 4
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Bunga 4: selesai
Pita: kertas - mawar - tulip
Bunga: 3
PS C:\Go lang>

```

Deskripsi Program

Program ini menerima input jumlah bunga N, lalu meminta user memasukkan nama bunga hingga sebanyak N atau sampai pengguna mengetikkan "SELESAI". Nama bunga disusun dalam pita yang dipisahkan oleh tanda " - ". Setelah selesai, program menampilkan pita beserta jumlah bunga yang dimasukkan. Jika N adalah 0, program langsung menampilkan pita kosong.

3. Source Code

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var beratKantong1, beratKantong2, totalBerat float64

    for {
        fmt.Println("Masukan berat belanjaan di kedua kantong: ")
        _, err := fmt.Scanf("%f %f", &beratKantong1, &beratKantong2)
        if err != nil {
            fmt.Println("Input tidak valid. Silakan coba lagi.")
            continue
        }
        if beratKantong1 < 0 || beratKantong2 < 0 {
            fmt.Println("Proses selesai.")
            break
        }
        totalBerat = beratKantong1 + beratKantong2

        if totalBerat > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := math.Abs(beratKantong1 - beratKantong2)

        if selisih >= 9 {
            fmt.Println("Sepeda motor pak Andi akan oleng: true")
        } else {
            fmt.Println("Sepeda motor pak Andi akan oleng: false")
        }
    }
}
```

Screenshoot Output

```
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: Input tidak valid. Silakan coba lagi.
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: Input tidak valid. Silakan coba lagi.
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: Input tidak valid. Silakan coba lagi.
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
PS C:\Go lang> █
```

Deskripsi Program

Program ini meminta input berat dari dua kantong belanjaan Pak Andi. Program akan terus meminta input hingga pengguna memasukkan berat negatif atau total berat kedua kantong melebihi 150 kg. Program kemudian menghitung selisih berat antara kedua kantong dan menentukan apakah sepeda motor Pak Andi akan oleng. Jika selisih beratnya 9 kg atau lebih, outputnya adalah true; jika tidak, outputnya adalah false.

4. Source Code

package main

```
import (
    "fmt"
    "math"
)
```

```
func f(k int) float64 {
    numerator := math.Pow(float64(4*k+2), 2)
    denominator := float64((4*k + 1) * (4*k + 3))
    return numerator / denominator
}
```

```
func sqrt2Approximation(K int) float64 {
    product := 1.0
    for k := 0; k <= K; k++ {
        product *= f(k)
    }
    return product
}
```

```

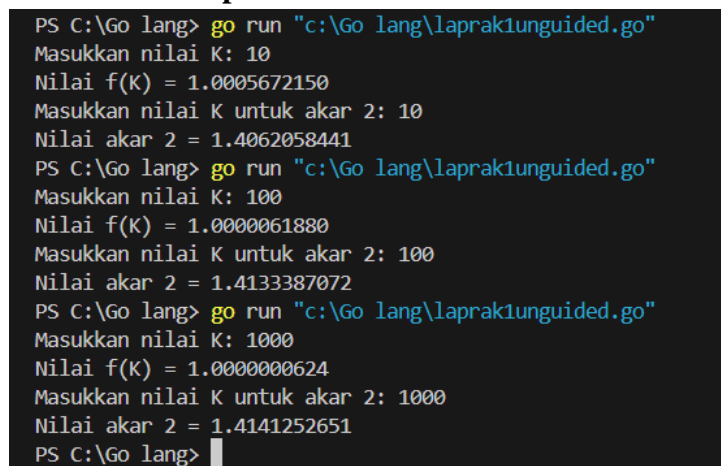
func main() {
    var K int

    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&K)
    fmt.Printf("Nilai f(K) = %.10f\n", f(K))

    fmt.Print("Masukkan nilai K untuk akar 2: ")
    fmt.Scan(&K)
    fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2Approximation(K))
}

```

Screenshoot Output



```

PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Masukkan nilai K: 10
Nilai f(K) = 1.0005672150
Masukkan nilai K untuk akar 2: 10
Nilai akar 2 = 1.4062058441
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Masukkan nilai K: 100
Nilai f(K) = 1.0000061880
Masukkan nilai K untuk akar 2: 100
Nilai akar 2 = 1.4133387072
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Masukkan nilai K: 1000
Nilai f(K) = 1.0000000624
Masukkan nilai K untuk akar 2: 1000
Nilai akar 2 = 1.4141252651
PS C:\Go lang>

```

Deskripsi Program

Program ini menghitung nilai dari fungsi matematis tertentu dan memberikan perkiraan akar dua. Fungsi $f(k)$ menghitung rasio antara kuadrat dari $4*k + 2$ dan produk dari $(4*k + 1)$ dan $(4*k + 3)$. Dalam fungsi $\text{sqrt2Approximation}(K)$, program mengalikan hasil dari fungsi $f(k)$ dari $k = 0$ hingga $k = K$ untuk mendapatkan perkiraan nilai akar dua. user diminta untuk memasukkan nilai K untuk kedua perhitungan, dan hasilnya ditampilkan dengan presisi 10 desimal.

2c

1. Source Code

```
package main

import "fmt"

func main() {
    var berat, kg, gr, biayaKirim, tambahanBiaya, totalBiaya int

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

    kg = berat / 1000
    gr = berat % 1000

    if kg > 10 {
        tambahanBiaya = 0
    } else if gr >= 500 {
        tambahanBiaya = gr * 5
    } else {
        tambahanBiaya = gr * 15
    }

    biayaKirim = kg * 10000
    totalBiaya = biayaKirim + tambahanBiaya

    fmt.Printf("Detail berat: %d kg + %d gr\n", kg, gr)
    fmt.Printf("Detail biaya: Rp. %d + Rp. %d\n", biayaKirim, tambahanBiaya)
    fmt.Printf("Total biaya: Rp. %d\n", totalBiaya)
}
```

Screenshoot Output

```
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Berat parcel (gram): 8500
Detail berat: 8 kg + 500 gr
Detail biaya: Rp. 80000 + Rp. 2500
Total biaya: Rp. 82500
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Berat parcel (gram): 9250
Detail berat: 9 kg + 250 gr
Detail biaya: Rp. 90000 + Rp. 3750
Total biaya: Rp. 93750
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Berat parcel (gram): 11750
Detail berat: 11 kg + 750 gr
Detail biaya: Rp. 110000 + Rp. 0
Total biaya: Rp. 110000
PS C:\Go lang> █
```

Deskripsi Program

Program ini menghitung biaya pengiriman parcel berdasarkan berat yang dimasukkan dalam gram. Pertama, program mengonversi berat dari gram ke kilogram dan sisa gram. Jika berat lebih dari 10 kg, tidak ada tambahan biaya. Jika sisa gram lebih dari atau sama dengan 500, biaya tambahan dihitung dengan tarif Rp. 5 per gram, sedangkan jika kurang dari 500 gram, tarifnya adalah Rp. 15 per gram. Biaya pengiriman dasar dihitung sebagai Rp. 10.000 per kilogram. Program kemudian menampilkan detail berat, rincian biaya, dan total biaya pengiriman.

2. Source Code

```
package main
```

```
import "fmt"
```

```
func main() {
    var nam float64
    var nmk string
    fmt.Print("Nilai akhir mata kuliah: ")
    fmt.Scanln(&nam)
    if nam > 80 {
        nmk = "A"
    } else if nam > 72.5 {
        nmk = "AB"
    } else if nam > 65 {
        nmk = "B"
    } else if nam > 57.5 {
```

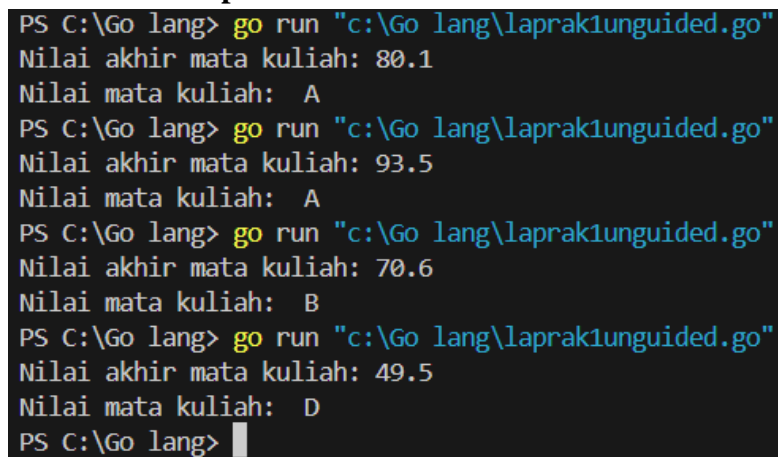


```

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

```

Screenshot Output



```

PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Nilai akhir mata kuliah: 80.1
Nilai mata kuliah: A
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Nilai akhir mata kuliah: 93.5
Nilai mata kuliah: A
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Nilai akhir mata kuliah: 70.6
Nilai mata kuliah: B
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Nilai akhir mata kuliah: 49.5
Nilai mata kuliah: D
PS C:\Go lang>

```

Deskripsi Program

Program ini menerima input nilai akhir mata kuliah dalam bentuk angka desimal. Berdasarkan nilai tersebut, program akan menentukan dan menampilkan grade (nilai huruf) yang sesuai menggunakan rentang nilai yang telah ditentukan: A, AB, B, BC, C, D, atau E. Setiap rentang nilai ditetapkan untuk memberikan grade yang sesuai dengan kriteria yang telah ditentukan, di mana nilai tertinggi (A) diberikan untuk nilai di atas 80, dan nilai terendah (E) untuk nilai di bawah atau sama dengan 40.

3. Source Code

```
package main

import "fmt"

func findFactors(b int) []int {
    var factors []int
    for i := 1; i <= b; i++ {
        if b%i == 0 {
            factors = append(factors, i)
        }
    }
    return factors
}

func isPrime(b int) bool {
    factors := findFactors(b)
    return len(factors) == 2
}

func main() {
    var b int
    fmt.Print("Bilangan: ")
    fmt.Scan(&b)

    factors := findFactors(b)
    fmt.Print("Faktor: ")
    for _, factor := range factors {
        fmt.Printf("%d ", factor)
    }
    fmt.Println()

    primeStatus := isPrime(b)
    fmt.Printf("Prima: %t\n", primeStatus)
}
```

Screenshoot Output

```
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS C:\Go lang> go run "c:\Go lang\laprak1unguided.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS C:\Go lang> 
```

Deskripsi Program

Program ini menerima input sebuah bilangan bulat dan kemudian menghitung faktor-faktornya. Fungsi findFactors digunakan untuk menemukan semua angka yang dapat membagi bilangan tersebut tanpa sisa. Selanjutnya, program juga memeriksa apakah bilangan tersebut merupakan bilangan prima dengan menggunakan fungsi isPrime, yang menentukan bahwa bilangan adalah prima jika hanya memiliki dua faktor (1 dan bilangan itu sendiri). Hasilnya, program menampilkan semua faktor yang ditemukan serta status apakah bilangan tersebut prima atau tidak.