

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
RIVIEW dan STUKTUR KONTROL**



**Disusun Oleh :
PETRA PRIADI S.P GINTING (2311102273)
IF-06**

**Dosen Pengampu :
ABEDNEGO DWI SEPTIADI**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024**

I. DASAR TEORI

Struktur kontrol dalam golang

Struktur kontrol dalam bahasa pemrograman Go (Golang) mencakup beberapa elemen penting seperti if statement, for loop, dan switch statement. If statement digunakan untuk mengevaluasi kondisi dan menjalankan blok kode jika kondisinya benar. For loop adalah satu-satunya jenis loop di Go, yang digunakan untuk mengulangi blok kode selama kondisi yang ditentukan benar, serta dapat digunakan untuk iterasi melalui elemen-elemen dalam slice atau map. Switch statement digunakan untuk memeriksa beberapa kondisi dan menjalankan blok kode yang sesuai dengan kondisi yang terpenuhi.

RIVIEW adalah pola yang sering digunakan dalam pengembangan perangkat lunak untuk memastikan bahwa data diproses dengan benar dan efisien. RIVIEW adalah singkatan dari Read, Initialize, Validate, Execute, Write. Pola ini melibatkan membaca data dari sumber, menginisialisasi variabel dan struktur data yang diperlukan, memvalidasi data, menjalankan logika bisnis utama, dan menulis hasil kembali ke sumber atau output yang diinginkan. Dengan memahami dan menerapkan struktur kontrol serta pola RIVIEW dalam Golang, Anda dapat menulis kode yang lebih bersih, mudah dipahami, dan efisien.

II. GUIDED

1. Soal Studi Case

Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silahkan masukkan data yang sesuai sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut ?

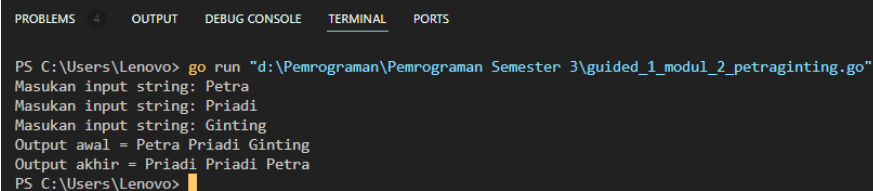
Sourcecode

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

Screenshoot Output



```
PROBLEMS 4 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\guided_1_modul_2_petrasingting.go"
Masukan input string: Petra
Masukan input string: Priadi
Masukan input string: Ginting
Output awal = Petra Priadi Ginting
Output akhir = Priadi Priadi Petra
PS C:\Users\Lenovo>
```

Deskripsi Program

Program di atas merupakan program yang menyuruh pengguna memasukkan inputan berupa string sebanyak tiga kali, kemudian disusun panjang menjadi sebuah kalimat sesuai urutan. Kemudian program akan mengubah nilai variabel yaitu nilai variabel satu disimpan sementara ke temp, kemudian nilai variabel dua dipindahkan ke satu dan nilai awal dari satu yang disimpan di temp dipindahkan ke tiga dan disusun menjadi satu kalimat yaitu Priadi Priadi Petra.

2. Soal Studi Case

Tahun kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut merupakan tahun kabisat (true) atau bukan (false)?

Sourcecode

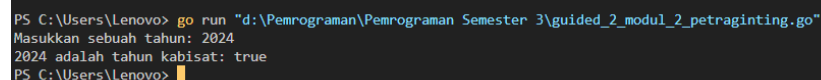
```
package main

import "fmt"

func main() {
    var tahun int
    fmt.Print("Masukkan sebuah tahun: ")
    fmt.Scanln(&tahun)

    if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {
        fmt.Println(tahun, "adalah tahun kabisat: true")
    } else {
        fmt.Println(tahun, "bukan tahun kabisat: false")
    }
}
```

Screenshoot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\guided_2_modul_2_petragingting.go"
Masukkan sebuah tahun: 2024
2024 adalah tahun kabisat: true
PS C:\Users\Lenovo>
```

Deskripsi Program

Program meminta pengguna untuk memasukkan sebuah tahun, kemudian memeriksa apakah tahun tersebut memenuhi salah satu dari dua kondisi: apakah tahun habis dibagi 400, atau tahun habis dibagi 4 tetapi tidak habis dibagi 100. Jika salah satu kondisi terpenuhi, program mencetak bahwa tahun tersebut adalah tahun kabisat; jika tidak, program mencetak bahwa tahun tersebut bukan tahun kabisat.

3. Soal Studi Case

Buat program bola yang menerima input jari-jari suatu bola (bilangan bulat). Tampilkan volume dan luas kulit bola. $\text{volumebola} = \frac{4}{3}\pi r^3$ dan $\text{luasbola} = 4\pi r^2$ ($\pi = 3.1415926535$).

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var Jarijari float64

    fmt.Print("Masukkan jari-jari bola: ")
    fmt.Scanln(&Jarijari)

    volume := (4.0 / 3.0) * math.Pi * math.Pow(Jarijari,
3)

    Luaspermukaan := 4 * math.Pi * math.Pow(Jarijari, 2)

    fmt.Printf("Volume bola dengan jari-jari %.2f adalah
%.2f\n", Jarijari, volume)

    fmt.Printf("Luas permukaan bola dengan jari-jari
%.2f adalah %.2f\n", Jarijari, Luaspermukaan)
}
```

Screenshoot Output

```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\guided_3_modul_2_petrating.go"
Masukkan jari-jari bola: 100
Volume bola dengan jari-jari 100.00 adalah 4188790.20
Luas permukaan bola dengan jari-jari 100.00 adalah 125663.71
PS C:\Users\Lenovo>
```

Deskripsi Program

Program di atas merupakan program perhitungan volume dan luas permukaan sebuah bola. Program tersebut meminta pengguna untuk menginputkan nilai jari-jari yang kemudian akan mengeksekusi inputan dengan rumus yang dibuat dan menghasilkan nilai yang sesuai dengan hasil pengoprasian pada rumus.

III. UNGUIDED

1. Soal Studi Case

Dibaca nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam Fahrenheit

$$Celsius = (Fahrenheit - 32) \times \frac{5}{9} \quad Reamur = Celsius \times \frac{4}{5} \quad Kelvin = (Fahrenheit + 459.67) \times \frac{5}{9}$$

(Contoh input/output, **Teks bergaris bawah** adalah input dari user):

Temperatur Celsius: 50
Derajat Fahrenheit: 122

Lanjutkan program di atas, sehingga temperatur dinyatakan juga dalam derajat Reamur dan Kelvin.

(Contoh input/output, **Teks bergaris bawah** adalah input dari user):

Temperatur Celsius: 50
Derajat Reamur: 40
Derajat Fahrenheit: 122
Derajat Kelvin: 323

Sourcecode

```
package main

import "fmt"

func celsiusKeFahrenheit(celsius float64) float64 {
    return (celsius * 9 / 5) + 32
}

func celsiusKeReamur(celsius float64) float64 {
    return celsius * 4 / 5
}

func celsiusKeKelvin(celsius float64) float64 {
    return celsius + 273.15
}

func main() {
    var celsius float64
    fmt.Print("Masukkan suhu dalam Celsius: ")
    fmt.Scanln(&celsius)

    fahrenheit := celsiusKeFahrenheit(celsius)
```

```

    reamur := celsiusKeReamur(celsius)
    kelvin := celsiusKeKelvin(celsius)

    fmt.Printf("Temperatur Celsius: %.2f\n", celsius)
    fmt.Printf("Derajat Reamur: %.2f\n", reamur)
    fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)
    fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)
}

```

Screenshoot Output

```

PS D:\Pemrograman\Pemrograman Semester 3> go run "d:\Pemrograman\Pemrograman Semester 3\guided_2_modul_2_petraging.go"
Masukkan suhu dalam Celsius: 50
Temperatur Celsius: 50.00
Derajat Reamur: 40.00
Derajat Fahrenheit: 122.00
Derajat Kelvin: 323.15
PS D:\Pemrograman\Pemrograman Semester 3>

```

Deskripsi Program

Kode ini merubah suhu dari Celsius ke Fahrenheit, Reamur, dan Kelvin. Pertama, kode mendefinisikan tiga fungsi untuk masing-masing konversi: `celsiusKeFahrenheit`, `celsiusKeReamur`, dan `celsiusKeKelvin`. Dalam fungsi main, pengguna diminta memasukkan suhu dalam Celsius. Setelah itu, suhu yang dimasukkan dikonversi menggunakan rumus yang sudah dibuat dan hasilnya ditampilkan.

2. Studi Case

Tipe karakter sebenarnya hanya apa yang tampak dalam tampilan. Di dalamnya tersimpan dalam bentuk biner 8 bit (byte) atau 32 bit (rune) saja.

Buat program ASCII yang akan membaca 5 buah data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII)

Masukan terdiri dari dua baris. Baris pertama berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

Keluaran juga terdiri dari dua baris. Baris pertama berisi 5 buah representasi karakter dari data yang diberikan, yang berdampingan satu dengan lain, tanpa dipisahkan spasi. Baris kedua berisi 3 buah karakter (juga tidak dipisahkan oleh spasi).

No.	Masukan	Keluaran
1	66 97 103 117 115 SNO	Bagus TOP

Catatan: Gunakan `fmt.Scanf("%c", &var)` untuk pembacaan satu karakter dan `fmt.Printf("%c", var)` untuk penulisan satu karakter.

Sourcecode

```
package main

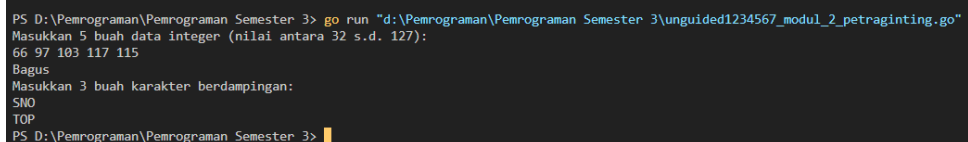
import (
    "fmt"
)

func main() {
    var intData [5]int
    fmt.Println("Masukkan 5 buah data integer (nilai
antara 32 s.d. 127):")
    for i := 0; i < 5; i++ {
        fmt.Scan(&intData[i])
    }
    for _, val := range intData {
        fmt.Printf("%c", val)
    }
    fmt.Println()

    fmt.Println("Masukkan 3 buah karakter
berdampingan:")
    fmt.Scan(&charData)

    if len(charData) != 3 {
        fmt.Println("Masukkan harus terdiri dari 3
karakter.")
        return
    }
    for _, char := range charData {
        fmt.Printf("%c", char+1)
    }
    fmt.Println()
}
```

Screenshoot Output



```
PS D:\Pemrograman\Pemrograman Semester 3> go run "d:\Pemrograman\Pemrograman Semester 3\unguided1234567_modul_2_petraginging.go"
Masukkan 5 buah data integer (nilai antara 32 s.d. 127):
66 97 103 117 115
Bagus
Masukkan 3 buah karakter berdampingan:
SNO
TOP
PS D:\Pemrograman\Pemrograman Semester 3> █
```

Deskripsi Program

Kode ini meminta pengguna untuk memasukkan lima bilangan bulat dalam rentang 32 hingga 127, yang kemudian disimpan dalam array `intData`. Setelah itu, setiap nilai dalam array dikonversi menjadi karakter ASCII dan ditampilkan. Selanjutnya, program meminta pengguna untuk memasukkan tiga karakter secara berdampingan. Jika input tidak terdiri dari tiga karakter, program akan menampilkan pesan kesalahan dan berhenti. Jika input valid, setiap karakter dalam input akan ditingkatkan satu nilai ASCII dan hasilnya ditampilkan.

3. 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 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 berurutan 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.

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var warna1, warna2, warna3, warna4 string
    berhasil := true

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        fmt.Scan(&warna1, &warna2, &warna3, &warna4)

        if warna1 != "merah" || warna2 != "kuning" ||
        warna3 != "hijau" || warna4 != "ungu" {
            berhasil = false
        }
    }
}
```

```

    }

    if berhasil {
        fmt.Println("Hasil : SUKSES")
    } else {
        fmt.Println("Hasil : GAGAL")
    }
}

```

Screenshoot Output

```

PS D:\Pemrograman\Pemrograman Semester 3> go run "d:\Pemrograman\Pemrograman Semester 3\unguided1234567_modul_2_petragingting.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
Hasil : SUKSES
PS D:\Pemrograman\Pemrograman Semester 3>

```

```

PS D:\Pemrograman\Pemrograman Semester 3> go run "d:\Pemrograman\Pemrograman Semester 3\unguided1234567_modul_2_petragingting.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: hijau kuning merah ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
Hasil : GAGAL
PS D:\Pemrograman\Pemrograman Semester 3>

```

Deskripsi Program

Empat variabel string (warna1, warna2, warna3, warna4) dideklarasikan untuk menyimpan input pengguna. Variabel boolean berhasil diinisialisasi dengan nilai true. Ini akan digunakan untuk melacak apakah semua percobaan berhasil. Loop for berjalan 5 kali, meminta pengguna untuk memasukkan empat warna setiap kali. Fungsi fmt.Scan membaca nilai input ke dalam warna1, warna2, warna3, dan warna4. Pernyataan if memeriksa apakah nilai input sesuai dengan warna yang diharapkan: “merah”, “kuning”, “hijau”, dan “ungu”. Jika ada input yang tidak sesuai, berhasil diatur ke false. Setelah loop, program memeriksa nilai berhasil. Jika berhasil masih true, program mencetak “Hasil : Sukses”.

Jika berhasil false, program mencetak “Hasil: Gagal”.

4. 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 – teratal – kamboja – anggrek

Modul Praktikum Algoritma dan Pemrograman 2

Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita.

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

Tampilkan isi pita setelah proses input selesai.

Sourcecode

```
package main

import "fmt"

func main() {
    var a, pita string
    i := 0
    for a != "SELESAI" {
        fmt.Printf("\nBunga %v : ", i+1)
        fmt.Scanln(&a)

        if a != "SELESAI" {
            a += " - "
            pita += a
            i++
        }
    }
}
```

```

    }
    fmt.Printf("\nPita   : %v\n", pita)
    fmt.Printf("Bunga  : %v\n", i)
}

```

Screenshot Output

```

PS D:\Pemrograman\Pemrograman Semester 3> go run "d:\Pemrograman\Pemrograman Semester 3\unguided1234567_modul_2_petraging.go"
Bunga 1 : Melati
Bunga 2 : Mawar
Bunga 3 : Anggrek
Bunga 4 : Matahari
Bunga 5 : Tulip
Bunga 6 : SELESAI
Pita   : Melati - Mawar - Anggrek - Matahari - Tulip -
Bunga : 5
PS D:\Pemrograman\Pemrograman Semester 3>

```

Deskripsi Program

Kode di atas adalah program yang meminta pengguna untuk memasukkan nama bunga secara berulang hingga pengguna memasukkan kata "SELESAI". Setiap kali pengguna memasukkan nama bunga, program menambahkan nama tersebut ke dalam string `pita` dengan tambahan " - " di akhir, dan menghitung jumlah bunga yang dimasukkan dengan variabel `i`. Setelah pengguna memasukkan "SELESAI", program berhenti dan mencetak string `pita` yang berisi semua nama bunga yang dimasukkan, serta jumlah total bunga yang dimasukkan. Program ini menggunakan loop `for` yang terus berjalan hingga kondisi `a != "SELESAI"` terpenuhi.

5. Studi Case

Setiap hari Pak Andi membawa banyak barang belanjaan dari pasar dengan mengendarai sepeda motor. Barang belanjaan tersebut dibawa dalam kantong terpal di kiri-kanan motor. Sepeda motor tidak akan oleng jika selisih berat barang di kedua kantong sisi tidak lebih dari 9 kg.

Buatlah program dengan bahasa golang untuk 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. Pada modifikasi program tersebut, program akan menampilkan true jika selisih kedua isi kantong lebih dari atau sama dengan 9 kg. Program berhenti memproses apabila total berat isi kedua kantong melebihi 150 kg atau salah satu kantong beratnya negatif

Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var berat1, berat2, totalBerat float64

    for {
        fmt.Print("Masukkan berat belanjaan di kedua kantong: ")
        fmt.Scan(&berat1, &berat2)

        if berat1 < 0 || berat2 < 0 {
            fmt.Println("Proses selesai.")
            break
        }

        totalBerat = berat1 + berat2
        if totalBerat > 150 {
            fmt.Println("Proses selesai.")
            break
        }

        selisih := berat1 - berat2
        if selisih < 0 {
            selisih = -selisih
        }

        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 D:\Pemrograman\Pemrograman Semester 3> go run "d:\Pemrograman\Pemrograman
Semester 3\unguided1234567_modul_2_petragingting.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:\Pemrograman\Pemrograman Semester 3> █
```

Deskripsi Program

Kode di atas adalah program yang meminta pengguna untuk memasukkan berat belanjaan di dua kantong terpal secara berulang. Program akan terus meminta input hingga salah satu dari dua kondisi terpenuhi: salah satu kantong memiliki berat negatif atau total berat kedua kantong melebihi 150 kg, yang akan menghentikan program dengan mencetak "Proses selesai." Selain itu, program menghitung selisih berat antara kedua kantong dan mencetak "Sepeda motor Pak Andi akan oleng: true" jika selisihnya 9 kg atau lebih, atau "Sepeda motor Pak Andi akan oleng: false" jika selisihnya kurang dari 9 kg. Program ini memastikan bahwa beban pada sepeda motor tetap seimbang dan tidak melebihi batas yang aman.

6. Studi Case

Diberikan sebuah persamaan sebagai berikut ini.

informatics lab

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

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

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

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

Modifikasi program sebelumnya yang menerima input integer K dan menghitung $\sqrt{2}$ untuk K tersebut. Hampiran $\sqrt{2}$ dituliskan dalam ketelitian 10 angka di belakang koma.

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

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

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var k int

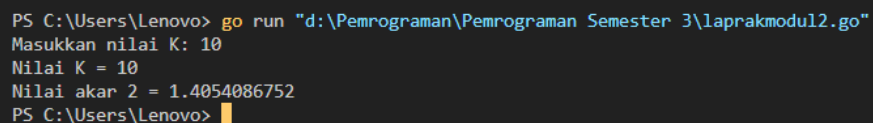
    fmt.Print("Masukkan nilai K: ")
    _, err := fmt.Scanf("%d", &k)
    if err != nil {
        fmt.Println("Error saat membaca input:", err)
        return
    }

    result := calculateSqrt2(k)
    fmt.Printf("Nilai K = %d\n", k)
    fmt.Printf("Nilai akar 2 = %.10f\n", result)
}

func calculateSqrt2(k int) float64 {
    result := 1.0
    for i := 0; i < k; i++ {
        result *= calculateF(float64(i))
    }
    return result
}

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

Screenshoot Output



```
PS C:\Users\Lenovo> go run "d:\Pemrograman\Pemrograman Semester 3\laprakmodul2.go"
Masukkan nilai K: 10
Nilai K = 10
Nilai akar 2 = 1.4054086752
PS C:\Users\Lenovo>
```

Deskripsi Program

Program meminta pengguna untuk memasukkan nilai integer (k), yang menentukan jumlah iterasi untuk perhitungan. Fungsi calculateSqrt2 mengalikan hasil dari fungsi calculateF sebanyak (k) kali untuk mendekati nilai akar kuadrat dari 2. Fungsi calculateF menghitung nilai berdasarkan rumus $\frac{(4k+2)^2}{(4k+1)(4k+3)}$. Hasil akhir dari perhitungan ini ditampilkan dengan presisi hingga 10 desimal.