

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

Bhara Yudha Pamungkas / 2311102186

IF-11-06

Dosen Pengampu :

Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Struktur kontrol merupakan komponen penting dalam bahasa pemrograman karena berfungsi untuk mengatur alur eksekusi program. Dalam bahasa Go (Golang), struktur kontrol yang tersedia mencakup pengulangan (loops), seleksi (conditional statements), dan berbagai bentuk pengendalian alur lainnya, seperti goroutine untuk konkuren.

2.1 Pengulangan (Loops)

Pengulangan dalam Golang adalah mekanisme untuk menjalankan satu blok kode secara berulang hingga suatu kondisi terpenuhi. Golang terutama menggunakan for loop yang fleksibel. Struktur dasarnya terdiri dari inisialisasi variabel, kondisi pengulangan, dan pernyataan yang dieksekusi setelah setiap iterasi. Selain for loop sederhana, Golang juga mendukung for range untuk mengiterasi elemen dalam array, slice, map, atau string. Kata kunci break digunakan untuk keluar dari loop, sedangkan continue untuk melewati iterasi saat ini. Dengan memahami konsep pengulangan, Anda dapat membuat program Golang yang lebih dinamis dan efisien untuk menyelesaikan berbagai tugas.

2.2 Seleksi (Conditional Statements)

Seleksi dalam Golang adalah mekanisme untuk membuat keputusan dalam program berdasarkan kondisi tertentu. Konstruksi utama yang digunakan untuk seleksi adalah if dan else. Dengan if, kita dapat mengevaluasi sebuah ekspresi boolean. Jika ekspresi bernilai benar (true), maka blok kode di dalam if akan dijalankan. Jika tidak, kita dapat menggunakan else untuk menjalankan blok kode alternatif. Selain itu, Golang juga mendukung switch case untuk mengevaluasi nilai dari suatu ekspresi dan menjalankan blok kode yang sesuai. Seleksi ini sangat penting untuk membuat program yang lebih dinamis dan dapat beradaptasi dengan berbagai kondisi.

2.3 Goroutine dan Pengendalian Konkuren

Goroutine dan pengendalian konkurensi dalam Golang adalah fitur yang sangat kuat untuk membuat program berjalan secara bersamaan. **Goroutine** adalah unit eksekusi ringan yang memungkinkan kita menjalankan fungsi secara konkuren. **Channels** digunakan sebagai saluran komunikasi antar goroutine untuk bertukar data. Dengan kombinasi goroutine dan channels, kita dapat membangun aplikasi yang sangat responsif dan efisien, seperti server web yang dapat menangani banyak permintaan secara simultan. Golang menyediakan alat yang mudah digunakan untuk mengelola konkurensi, sehingga kita dapat membangun aplikasi yang skalabel dan dapat memanfaatkan sepenuhnya sumber daya multi-core.

II. GUIDED

1. Soal Studi Case

Buat program Go yang menerima tiga input string berbeda dari pengguna dan kemudian mengeluarkan string dalam urutan aslinya, diikuti oleh nilainya

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 D:\Kampus\Alpro 2\Modul 3\tugas3> go run .
Masukan input string : 1
Masukan input string : 2
Masukan input string : 3
Output awal = 1 2 3
Output akhir = 2 3 1
```

Deskripsi Program

Program ini adalah sebuah program sederhana yang menerima tiga input string dari pengguna, menampilkan urutan awal input tersebut, kemudian menukar nilai dari ketiga variabel string tersebut. Program menampilkan hasil akhir setelah pertukaran nilai dilakukan.

2. Soal Studi Case

Kode yang diberikan adalah program Go sederhana yang menentukan apakah suatu tahun merupakan tahun kabisat. Program tersebut meminta pengguna untuk memasukkan tahun, memproses masukan tersebut untuk memeriksa apakah tahun tersebut memenuhi kriteria sebagai tahun kabisat, lalu mengeluarkan hasilnya. Penentuan tahun kabisat ditangani oleh fungsi `isLeapYear` yang menerapkan aturan untuk tahun kabisat: suatu tahun merupakan tahun kabisat jika habis dibagi 400 atau jika habis dibagi 4 tetapi tidak habis dibagi 100.

Source code :

```
package main

import (
    "fmt"
)

func main() {
    var tahun int

    fmt.Print("Tahun: ")
    fmt.Scanln(&tahun)
    kabisat := isLeapYear(tahun)
    fmt.Printf("Kabisat: %t\n", kabisat)
}

func isLeapYear(year int) bool {
    if year%400 == 0 {
        return true
    }
    if year%100 == 0 {
        return false
    }
    if year%4 == 0 {
        return true
    }
}
```

```
    return false  
}
```

Screenshoot Output :

```
PS D:\go> go run .\Alpro 2\Modul 3\tugas3>
Tahun: 2024
Kabisat: true
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .
Tahun: 2004
Kabisat: true
PS D:\Kampus\Alpro 2\Modul 3\tugas3> |
```

Deskripsi Program

Program ini meminta input tahun dari pengguna dan menentukan apakah tahun tersebut merupakan tahun kabisat. Menggunakan aturan pembagian: tahun yang habis dibagi 400 adalah kabisat, yang habis dibagi 100 tetapi tidak habis dibagi 400 bukan kabisat, dan yang habis dibagi 4 tetapi tidak habis dibagi 100 adalah kabisat. Program kemudian menampilkan hasilnya (true/false).

3. Soal Studi Case

Kode Go yang diberikan mendefinisikan program yang menghitung volume dan luas permukaan bola berdasarkan radius yang diberikan pengguna. Program ini terdiri dari fungsi `CalculateSphereProperties` yang melakukan perhitungan dan `main` fungsi yang berfungsi sebagai titik masuk program. Program meminta pengguna untuk memasukkan radius, memvalidasinya, menghitung properti yang diperlukan, lalu menampilkan hasilnya. Perhitungan didasarkan pada rumus matematika standar untuk volume dan luas permukaan bola.

Source code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var jariJari float64

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

    // Menghitung volume bola
    volumeBola := (4.0 / 3.0) * math.Pi * math.Pow(jariJari, 3)

    // Menghitung luas permukaan bola
    luasKulit := 4 * math.Pi * math.Pow(jariJari, 2)

    // Menampilkan hasil
    fmt.Printf("Bola dengan jari-jari %.2f memiliki volume %.4f dan luas kulit %.4f\n", jariJari, volumeBola, luasKulit)
}
```

Screenshoot Output :

```
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .
Masukkan jari-jari bola: 10
Bola dengan jari-jari 10.00 memiliki volume 4188.7902 dan luas kulit 1256.6371
PS D:\Kampus\Alpro 2\Modul 3\tugas3> |
```

Deskripsi Program

Program ini meminta input jari-jari sebuah bola dari pengguna, lalu menghitung dan menampilkan volume serta luas permukaan bola. Volume bola dihitung dengan rumus $\frac{4}{3} \times \pi \times \text{jariJari}^3$, sedangkan luas permukaan bola dihitung dengan rumus $4 \times \pi \times \text{jariJari}^2$, menggunakan fungsi `math.Pi` dan `math.Pow` dari paket `math`. Program kemudian menampilkan hasil perhitungan volume dan luas permukaan dengan format desimal dua tempat untuk jari-jari dan empat tempat untuk volume serta luas kulit bola.

III. UNGUIDED

1. Soal Studi Case

Kode yang diberikan adalah program Go (Golang) yang mendefinisikan fungsi untuk mengonversi suhu dari Celsius ke tiga skala suhu lainnya: Fahrenheit, Reamur, dan Kelvin. ConvertTemperatureFungsi tersebut mengambil suhu dalam Celsius sebagai input dan mengembalikan suhu yang sesuai dalam tiga skala lainnya. Fungsi tersebut mainberfungsi sebagai titik masuk program, yang meminta pengguna untuk memasukkan suhu dalam Celsius, melakukan konversi, lalu menampilkan hasilnya.

Source Code :

```
package main
import "fmt"
func main() {
var celsius float64

fmt.Print("Masukkan suhu dalam derajat Celsius: ")
fmt.Scanln(&celsius)

// Menghitung konversi suhu
fahrenheit := (celsius * 9 / 5) + 32
reamur := celsius * 4 / 5
kelvin := celsius + 273.15

// Menampilkan hasil
fmt.Printf("Suhu dalam Fahrenheit: %.2f\n", fahrenheit)
fmt.Printf("Suhu dalam Reamur: %.2f\n", reamur)
fmt.Printf("Suhu dalam Kelvin: %.2f\n", kelvin)
}
```

Screenshoot Output :

```
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .
Masukkan suhu dalam derajat Celsius: 32
Suhu dalam Fahrenheit: 89.60
Suhu dalam Reamur: 25.60
Suhu dalam Kelvin: 305.15
```

Deskripsi Program

Program ini mengonversi suhu dari Celsius ke tiga skala suhu lainnya: Fahrenheit, Reamur, dan Kelvin. Pengguna diminta memasukkan suhu dalam derajat Celsius, kemudian program menghitung konversi menggunakan rumus: $Fahrenheit = (Celsius \times 9/5) + 32$, $Reamur = Celsius \times 4/5$, dan $Kelvin = Celsius + 273.15$. Setelah menghitung, program menampilkan hasil konversi suhu tersebut dalam tiga skala dengan dua angka desimal untuk setiap hasil.

2. Soal Studi Case

Membuat program dalam bahasa Go 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).

Source code

```
package main

import (
    "fmt"
)

func main() {
    var data [5]int
    var karakter [3]rune

    fmt.Println("===== Program ASCII =====")

    // Membaca 5 angka dari input
```

```

    fmt.Println("Masukkan 5 angka (dari 32 hingga 127),
dipisahkan dengan spasi:")
    for i := 0; i < 5;
        i++ {
        fmt.Scan(&data
[i])
if data[i] < 32 || data[i] > 127 {
fmt.Println("Angka harus antara 32 dan 127.") return
}
}

    fmt.Println("\n===== Keluaran =====")

// Mencetak karakter untuk 5 angka yang dimasukkan
    fmt.Print("Karakter yang sesuai: ")
    for i := 0; i < 5; i++ {
    fmt.Print(string(rune(data[i])))
    }

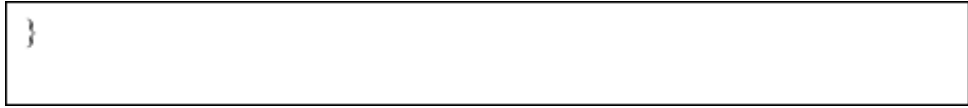
    fmt.Println()

// Membaca 3 karakter dari input
    fmt.Println("Masukkan 3 karakter yang berdampingan
(tanpa spasi):")
    var input string
    fmt.Scan(&inp
ut)

    if len(input) != 3 {
    fmt.Println("Harap masukkan tepat 3 karakter.")
    return
    }

// Mengonversi karakter ke rune
    for i, char := range input {
    karakter[i] = char
    }

```



Screenshoot Output

```
PS C:\Golang> go run "c:\Golang\tempCodeRunnerFile.go"
===== Program ASCII =====
Masukkan 5 angka (dari 32 hingga 127), dipisahkan dengan spasi:
66 97 103 117 115

===== Keluaran =====
Karakter yang sesuai: Bagus
Masukkan 3 karakter yang berdampingan (tanpa spasi):
SNO

===== Keluaran =====
Karakter setelah yang dimasukkan: TOP
PS C:\Golang>
```

Deskripsi Program

Kode tersebut merupakan program Go yang meminta pengguna memasukkan 5 angka (32-127) dan 3 karakter berdampingan, lalu menampilkan karakter ASCII yang sesuai dengan angka dan 3 karakter berikutnya.

Bagian 1:

- Meminta pengguna memasukkan 5 angka (32-127) dan memvalidasi inputnya.
- Mencetak karakter ASCII yang sesuai dengan angka-angka yang dimasukkan.

Bagian 2:

- Meminta pengguna memasukkan 3 karakter berdampingan.
- Mengonversi karakter ke tipe data rune.
- Mencetak 3 karakter berikutnya dari karakter yang dimasukkan.

Kode tersebut mendemonstrasikan penggunaan array, konversi tipe data rune dan string, serta operasi aritmatika pada karakter.

3. Soal Studi Case

Kode Go yang diberikan mendefinisikan aplikasi konsol sederhana yang meminta pengguna untuk memasukkan urutan empat warna. Program tersebut memeriksa apakah warna yang dimasukkan cocok dengan urutan yang telah ditetapkan: "merah", "kuning", "hijau", dan "ungu". Pengguna memiliki lima kali kesempatan untuk memasukkan urutan yang benar. Jika pengguna berhasil mencocokkan urutan tersebut, pesan berhasil akan dicetak; jika tidak, pesan gagal akan ditampilkan.

Source Code :

```
package main

import (
    "fmt"
)

func main() {
    var data [5]int
    var karakter [3]rune

    fmt.Println("==== Program ASCII ====")

    // Membaca 5 angka dari input
    fmt.Println("Masukkan 5 angka (dari 32 hingga 127), dipisahkan dengan spasi:")
    for i := 0; i < 5; i++ {
        fmt.Scan(&data[i])
        if data[i] < 32 || data[i] > 127 {
            fmt.Println("Angka harus antara 32 dan 127.")
            return
        }
    }

    fmt.Println("\n==== Keluaran ====")
    // Mencetak karakter untuk 5 angka yang dimasukkan
    fmt.Print("Karakter yang sesuai: ")
    for i := 0; i < 5; i++ {
        fmt.Print(string(rune(data[i])))
    }
    fmt.Println()

    // Membaca 3 karakter dari input
    fmt.Println("Masukkan 3 karakter yang berdampingan (tanpa spasi):")
    var input string
    fmt.Scanln(&input)

    if len(input) != 3 {
        fmt.Println("Harap masukkan tepat 3 karakter.")
        return
    }

    // Mengonversi karakter ke rune
```

```

for i := 0; i < 3; i++ {
    karakter[i] = rune(input[i])
}

fmt.Println("\n===== Keluaran =====")
// Mencetak 3 karakter setelah karakter yang dimasukkan
fmt.Print("Karakter setelah yang dimasukkan: ")
for _, char := range karakter {
    fmt.Print(string(char + 1)) // Menampilkan karakter berikutnya
}
fmt.Println()
}

```

Screenshot Output :

```

PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .
===== Program ASCII =====
Masukkan 5 angka (dari 32 hingga 127), dipisahkan dengan spasi:
39
75
81
92
126

===== Keluaran =====
Karakter yang sesuai: 'KQ\~
Masukkan 3 karakter yang berdampingan (tanpa spasi):
Harap masukkan tepat 3 karakter.
PS D:\Kampus\Alpro 2\Modul 3\tugas3> 

```

Deskripsi Program

Program ini adalah program untuk mengonversi angka ke karakter ASCII dan mengubah karakter yang dimasukkan oleh pengguna. Pertama, program meminta pengguna untuk memasukkan 5 angka antara 32 hingga 127, yang kemudian dikonversi menjadi karakter ASCII dan ditampilkan. Setelah itu, program meminta pengguna untuk memasukkan 3 karakter yang berdampingan tanpa spasi, lalu menampilkan karakter berikutnya dari masing-masing karakter yang dimasukkan. Program ini menggunakan array untuk menyimpan data angka dan karakter, serta memanfaatkan fungsi `rune` untuk menangani konversi karakter ASCII.

4. Soal Studi Case

Kode Go yang disediakan adalah aplikasi konsol sederhana yang memungkinkan pengguna memasukkan nama-nama bunga. Program tersebut meminta pengguna untuk memasukkan jumlah bunga yang ingin mereka masukkan, mengumpulkan nama-nama bunga, dan memformatnya menjadi satu string yang dipisahkan oleh " - ". Program tersebut juga menghitung jumlah bunga yang dimasukkan dan menampilkan jumlah ini. Program tersebut menghentikan proses input jika pengguna mengetik "SELESAI".

Source Code :

```
package main

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

func main() {
    var N int
    var bunga string
    var pita string
    var jumlahBunga int

    fmt.Print("Masukkan banyak bunga (N): ")
    fmt.Scanln(&N)

    if N <= 0 {
        fmt.Println("Input N harus bilangan bulat positif.")
        return
    }
}
```

```
scanner := bufio.NewScanner(os.Stdin)
for i := 1; i <= N; i++ {
    fmt.Printf("Bunga %d: ", i)
    scanner.Scan()
    bunga = scanner.Text()

    if bunga == "SELESAI" {
        break
    }

    if i > 1 {
        pita += " - "
    }
    pita += bunga
    jumlahBunga++
}

fmt.Println("Pita:", pita)
fmt.Println("Jumlah Bunga:", jumlahBunga)
}
```


Screenshot Output :

```
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .  
Masukkan banyak bunga (N): 8  
Bunga 1: 2  
Bunga 2: 5  
Bunga 3: 1  
Bunga 4: 56  
Bunga 5: 1  
Bunga 6: 6  
Bunga 7: 3  
Bunga 8: 4  
Pita: 2 - 5 - 1 - 56 - 1 - 6 - 3 - 4  
Jumlah Bunga: 8  
PS D:\Kampus\Alpro 2\Modul 3\tugas3> 
```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan jumlah bunga (N) yang ingin dicatat, dan kemudian mengizinkan pengguna untuk memasukkan nama bunga satu per satu. Jika pengguna mengetik "SELESAI", program akan menghentikan permintaan input. Selama proses ini, nama bunga yang dimasukkan akan disimpan dalam format pita yang dipisahkan oleh " - ", dan program juga menghitung jumlah total bunga yang dicatat. Setelah selesai, program menampilkan daftar nama bunga dan jumlah total bunga yang telah dimasukkan oleh pengguna

5. Soal Studi Case Sebelum modifikasi

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 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.

Source Code :

```
package main

import ( "fmt"
)

func main() {

var kantong1, kantong2 float64
for {

    // Input the weight of the two bags

    fmt.Print("Masukan berat belanjaan di kedua kantong: ")
    fmt.Scan(&kantong1, &kantong2)

    // Check if either of the bags contains 9 kg or more
    if kantong1 >= 9 || kantong2 >= 9 {
        fmt.Println("Proses selesai.")
        break
    }

}

}
```

Screenshot Program :

```
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .
Masukan berat belanjaan di kedua kantong: 2 14
Proses selesai.
```

Deskripsi program

Program ini meminta pengguna untuk memasukkan berat belanjaan di dua kantong secara berulang hingga salah satu kantong mencapai berat 9 kg atau lebih. Setelah pengguna memasukkan berat kedua kantong, program akan memeriksa apakah berat salah satu kantong memenuhi syarat tersebut. Jika ya, program akan menampilkan pesan "Proses selesai." dan berhenti. Jika tidak, program akan terus meminta input hingga kondisi tersebut terpenuhi, sehingga menciptakan loop yang berlanjut sampai salah satu kantong melebihi batas berat yang ditentukan.

Sesudah modifikasi

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

Source Code :

```
package main

import (
    "fmt"
    "math"
)

func main() {
    var kantong1, kantong2 float64

    for {
        // Meminta input dari pengguna
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scan(&kantong1, &kantong2)

        // Memeriksa apakah salah satu kantong beratnya negatif atau total
        // berat melebihi 150 kg
        if kantong1 < 0 || kantong2 < 0 || kantong1+kantong2 > 150 {
            fmt.Println("Proses selesai.")
            break
        }
    }
}
```

```
// Menampilkan true jika selisih berat kedua kantong >= 9 kg
if math.Abs(kantong1-kantong2) >= 9 {
    fmt.Println("Sepeda motor pak Andi akan oleng: true")
} else {
    fmt.Println("Sepeda motor pak Andi akan oleng: false")
}
}
```

Screenshot Output :

```
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .  
Masukan berat belanjaan di kedua kantong: 3 19  
Sepeda motor pak Andi akan oleng: true  
Masukan berat belanjaan di kedua kantong: |
```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan berat belanjaan di dua kantong secara berulang. Program memeriksa apakah salah satu kantong memiliki berat negatif atau jika total berat kedua kantong melebihi 150 kg; jika kondisi tersebut terpenuhi, program akan menghentikan proses dengan menampilkan pesan "Proses selesai." Jika berat kedua kantong valid, program akan mengecek apakah selisih berat antara kedua kantong adalah 9 kg atau lebih, dan menampilkan pesan yang sesuai untuk menentukan apakah sepeda motor Pak Andi akan "oleng" atau tidak.

6. Soal Studi Case

Sebelum modifikasi

Membuat program yang menerima input sebuah bilangan sebagai K, kemudian menghitung dan menampilkan nilai f(K) sesuai dengan rumus berikut!

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

Sebelum Modifikasi

```
package main

import (
    "fmt"
)

func main() {
    // Declare the variable to hold the input
    var K int

    // Ask for user input
    fmt.Print("Nilai K = ")
    fmt.Scanf("%d", &K)

    // Initialize the value of v2 (approximation of sqrt(2))
    v2 := 1.0

    // Loop through the range from 0 to K to compute the
    product
    for k := 0; k <= K; k++ {
        numerator := (4*float64(k) + 2) * (4*float64(k) + 2)
        denominator := (4*float64(k) + 1) * (4*float64(k) + 3)
        v2 *= numerator / denominator
    }

    // Display the result as the approximation of sqrt(2)
    fmt.Printf("Nilai akar 2 = %.10f\n", v2)
}
```

Screenshot program

```
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .  
Nilai K = 85  
Nilai akar 2 = 1.4131861744  
PS D:\Kampus\Alpro 2\Modul 3\tugas3>
```

Deskripsi program

Program ini meminta pengguna untuk memasukkan nilai K dan kemudian menghitung pendekatan nilai $\sqrt{2}$ menggunakan metode produk. Program menggunakan rumus yang melibatkan pembilang dan penyebut berdasarkan indeks k dalam rentang 0 hingga K . Hasil perhitungan akhirnya ditampilkan dengan presisi sepuluh angka desimal.

Sesudah modifikasi

Memodifikasi program yang menerima input sebuah bilangan sebagai K , kemudian menghitung $\sqrt{2}$ untuk k tersebut. Hampiran $\sqrt{2}$ dituliskan dalam ketelitian 10 angka dibelakang koma dengan menggunakan rumus berikut ini!

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

Source Code

```
package main  
  
import (  
    "fmt"  
)  
  
func main() {  
    // Declare the variable to hold the input  
    var K int  
    // Ask for user input  
    fmt.Print("Nilai K = ")  
    fmt.Scanf("%d", &K)  
  
    // Initialize the value of v2 (approximation of sqrt(2))  
    v2 := 1.0  
  
    // Loop through the range from 0 to K to compute the product
```

```
for k := 0; k <= K; k++ {  
    numerator := (4*float64(k) + 2) * (4*float64(k) + 2)  
    denominator := (4*float64(k) + 1) * (4*float64(k) + 3)  
    v2 *= numerator / denominator  
}  
  
// Display the result as the approximation of sqrt(2) fmt.Printf("Nilai  
akar 2 = %.10f\n", v2)  
}
```


Screenshot Output

```
PS D:\Kampus\Alpro 2\Modul 3\tugas3> go run .  
Nilai K = 91
```

Deskripsi Program

Program ini menghitung pendekatan nilai $2\sqrt{2}$ menggunakan metode produk berdasarkan input pengguna. Setelah meminta pengguna untuk memasukkan nilai KKK, program menginisialisasi variabel `v2` untuk menyimpan hasil perhitungan, dimulai dari 1.0. Program kemudian menggunakan loop untuk menghitung produk dari pembilang dan penyebut yang didefinisikan dalam rumus, dengan pembilang dihitung sebagai $(4k+2)^2(4k+2)^2$ dan penyebut sebagai $(4k+1)(4k+3)(4k+1)(4k+3)$ untuk setiap kkk dari 0 hingga KKK. Hasil akhirnya adalah nilai pendekatan $2\sqrt{2}$, yang ditampilkan dengan format sepuluh angka desimal. Program ini memberikan cara untuk mendekati nilai akar kuadrat dua dengan menggunakan metode numerik