

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2
RIVIEW STRUKTUR KONTROL**



Disusun Oleh :

HANIF REYHAN ZHAFRAN ADRYTONA / 2311102266

11 IF 06

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Struktur kontrol dalam Go mencakup aliran sekuensial, percabangan, perulangan, dan transfer. Program Go biasanya menjalankan instruksi secara berurutan, tetapi percabangan seperti if, if-else, dan switch memungkinkan pemilihan jalur eksekusi berdasarkan kondisi tertentu, dengan switch yang secara default tidak membutuhkan break untuk menghindari fall-through. Perulangan dilakukan menggunakan for, yang berfungsi baik sebagai loop umum, while loop, atau untuk iterasi pada koleksi data. Kontrol aliran juga didukung oleh break, continue, dan return, yang memungkinkan pengaturan aliran program dengan lebih fleksibel. Menguasai struktur kontrol ini penting untuk menulis kode Go yang efektif dan mudah dipahami.

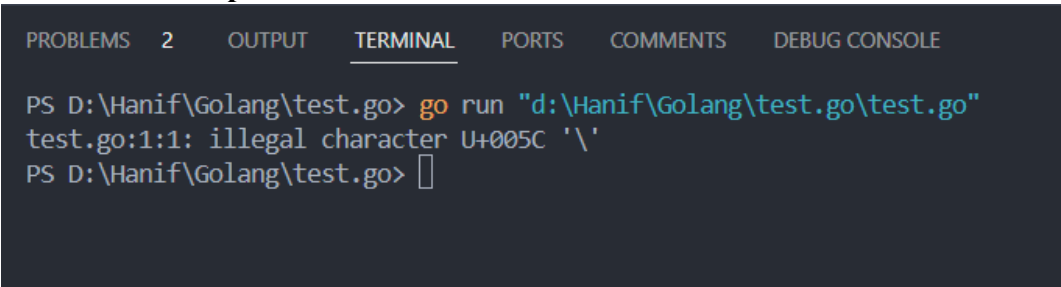
II. GUIDED

Guided 1.0

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

Screenshoot Output



```
PROBLEMS 2 OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE

PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
test.go:1:1: illegal character U+005C '\'
PS D:\Hanif\Golang\test.go> 
```

Deskripsi Program

Program Go ini meminta tiga string dari pengguna, menyimpannya dalam variabel satu, dua, dan tiga, lalu menampilkan urutan awalnya. Setelah itu, program merotasi nilai-nilai string: nilai di satu dipindah ke dua, dua ke tiga, dan tiga ke satu, lalu menampilkan urutan yang baru. Dalam studi kasus lainnya,

untuk menentukan apakah sebuah tahun adalah tahun kabisat, program akan menerima input berupa bilangan bulat dan memeriksa apakah bilangan tersebut memenuhi syarat: tahun kabisat adalah yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100, dan menampilkan hasilnya dalam bentuk true atau false.

Guided 1.2

Sourcecode

```
package main

import "fmt"

func main() {
    var tahun int
    var kabisat bool
    fmt.Print("Tahun: ")
    fmt.Scanln(&tahun)
    if tahun%400 == 0 {
        kabisat = true
    } else if tahun%100 == 0 {
        kabisat = false
    } else if tahun%4 == 0 {
        kabisat = true
    } else {
        kabisat = false
    }
    fmt.Println("Kabisat:", kabisat)
}
```

Screenshoot Output

```
PROBLEMS  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE

PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
Tahun: 2024
Kabisat: true
PS D:\Hanif\Golang\test.go> 
```

Deskripsi Program

Program Go ini menghitung apakah suatu tahun adalah tahun kabisat dengan menggunakan operasi modulus 4. Pertama, pengguna diminta untuk memasukkan tahun sebagai input. Program kemudian memeriksa sisa pembagian tahun tersebut dengan 4. Jika hasilnya menunjukkan sisa bagi (yaitu, tahun dibagi 4 tidak menghasilkan sisa 0), maka program akan mengeluarkan output false, yang berarti tahun tersebut bukan tahun kabisat. Sebaliknya, jika sisa bagi dari operasi modulus 4 adalah 0, program akan mengeluarkan output *true*, menandakan bahwa tahun tersebut memenuhi salah satu syarat untuk menjadi tahun kabisat. Dengan cara ini, program secara efektif mengidentifikasi tahun kabisat berdasarkan aturan yang berlaku.

Guided 1.3

Sourcecode

```
package main
import "fmt"
func main() {
    maxF := 100
    f0 := 0
    f1 := 1
    f2 := 1
    fmt.Println("Bilangan pertama:", f1)
    for f2 <= maxF {
        f0 = f1
        f1 = f2
        f2 = f1 + f0
        fmt.Println("Bilangan berikutnya:", f1)
    }
}
```

Screenshoot Output

```
PROBLEMS  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE

● PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
  Bilangan pertama: 1
  Bilangan berikutnya: 1
  Bilangan berikutnya: 2
  Bilangan berikutnya: 3
  Bilangan berikutnya: 5
  Bilangan berikutnya: 8
  Bilangan berikutnya: 13
  Bilangan berikutnya: 21
  Bilangan berikutnya: 34
  Bilangan berikutnya: 55
  Bilangan berikutnya: 89
○ PS D:\Hanif\Golang\test.go> 
```

Deskripsi Program

Program di atas menghasilkan dan mencetak bilangan Fibonacci hingga nilai maksimum yang ditentukan, yaitu 100. Bilangan Fibonacci dimulai dengan nilai 0 dan 1, dan setiap bilangan berikutnya merupakan penjumlahan dari dua bilangan sebelumnya. Program ini memulai dengan f0 sebagai 0, f1 sebagai 1, dan f2 sebagai 1. Pertama, program mencetak nilai pertama dari deret, yaitu 1. Selanjutnya, di dalam loop for, nilai f0, f1, dan f2 terus diperbarui untuk menghasilkan bilangan Fibonacci berikutnya, dan loop akan terus berjalan selama nilai f2 tidak melebihi 100. Setiap bilangan yang dihasilkan dicetak hingga mencapai batas tersebut.

III. UNGUIDED

Soal Studi Case

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

Sourcecode

```
package main
import (
    "fmt"
)
func main() {
    var celsius float64
    fmt.Print("Masukkan suhu dalam derajat Celsius: ")
    fmt.Scanf("%f", &celsius)
    reamur := (4.0 / 5.0) * celsius
    fahrenheit := (celsius * 9.0 / 5.0) + 32.0
    kelvin := celsius + 273.15
    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



```
PROBLEMS  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE

PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
Masukkan suhu dalam derajat Celsius: 45
Temperatur Celsius: 45.00
Derajat Reamur: 36.00
Derajat Fahrenheit: 113.00
Derajat Kelvin: 318.15
PS D:\Hanif\Golang\test.go> █
```

Deskripsi Program

Program Go ini dirancang untuk mengonversi suhu yang dimasukkan oleh pengguna dalam derajat Celsius ke tiga skala suhu lainnya: Reamur, Fahrenheit, dan Kelvin. Setelah mengimpor paket fmt, program meminta pengguna untuk memasukkan suhu dalam derajat Celsius dan

menyimpannya dalam variabel bertipe float64. Selanjutnya, program menghitung suhu dalam derajat Reaumur dengan rumus $((\frac{4}{5}) \times C)$, derajat Fahrenheit dengan rumus $((C \times \frac{9}{5}) + 32)$, dan Kelvin dengan menambahkan 273.15 ke suhu Celsius. Akhirnya, program menampilkan hasil konversi suhu dalam format dua desimal untuk setiap skala, sehingga pengguna dapat dengan mudah melihat hasilnya. Program ini berguna untuk aplikasi yang memerlukan konversi antara berbagai satuan suhu.

Unguided 2

SOAL :

Berikut adalah salinan teks dari gambar: 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).

Sourcecode

```
package main

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

func main() {
    var a, b, c, d, e int
    reader := bufio.NewReader(os.Stdin)

    fmt.Println("Masukkan 5 buah angka (nilai antara 32 hingga 127):")
```



```

    fmt.Scanf("%d %d %d %d %d", &a, &b, &c, &d, &e)

    // Mencetak hasil karakter dari angka yang diinput
    fmt.Printf("Hasil karakter dari angka yang diinput:
%c%c%c%c%c\n", a, b, c, d, e)

    fmt.Println("Masukkan karakter (contoh input: 123):")
    inputStr, _ := reader.ReadString('\n')

    shiftedStr := ""
    for _, char := range inputStr {
        // Mengabaikan karakter newline dan carriage return
        if char != '\n' && char != '\r' {
            shiftedStr += string(char + 1) // Menggeser karakter
            satu posisi
        }
    }

    fmt.Printf("Hasil karakter dari inputan yang digeser: %s\n",
shiftedStr)
}

```

Screenshoot Output

```

PROBLEMS  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE

.\test.go:11:5: syntax error: unexpected : at end of statement
PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
Masukkan 5 buah angka (nilai antara 32 hingga 127):
49
Hasil karakter dari angka yang diinput: 1
Masukkan karakter (contoh input: 123):
932
Hasil karakter dari inputan yang digeser: :43
PS D:\Hanif\Golang\test.go>

```

Deskripsi Program

Program di atas ditulis dalam bahasa Go dan bertujuan untuk menerima input dari pengguna berupa lima angka bulat, yang nilainya berada dalam rentang 32 hingga 127, serta satu string karakter. Setelah meminta pengguna untuk memasukkan angka-angka tersebut, program menggunakan `fmt.Scanf` untuk membaca dan menyimpan nilai-nilai tersebut ke dalam variabel a, b, c, d, dan e, yang kemudian dikonversi menjadi karakter

berdasarkan tabel ASCII dan dicetak sebagai output. Selanjutnya, program meminta pengguna untuk memasukkan karakter dalam bentuk string, yang dibaca menggunakan `*bufio.Reader*` untuk menangani input dengan benar. Dalam proses ini, program menggeser setiap karakter dalam input string dengan menambahkannya 1 (misalnya, karakter 'a' menjadi 'b'), sambil mengabaikan karakter newline dan carriage return. Hasil dari karakter yang diinputkan dan karakter yang telah digeser ditampilkan kepada pengguna, memberikan contoh penggunaan konversi dan manipulasi string dalam bahasa pemrograman Go.

UNGUIDED 3

Siswa kelas IPA disalah 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.

Sourcecode

```
package main

import "fmt"

func main() {
    const percobaan = 5
    expectedColors := [4]string{"merah", "kuning", "hijau", "ungu"}
    var hasilPercobaan bool = true

    for i := 0; i < percobaan; i++ {
        var warna [4]string
        fmt.Printf("Masukkan warna untuk percobaan ke-%d (pisahkan dengan spasi):\n", i+1)
        for j := 0; j < 4; j++ {
            fmt.Scan(&warna[j])
        }

        for j := 0; j < 4; j++ {
            if warna[j] != expectedColors[j] {
                hasilPercobaan = false
            }
        }
    }
}
```

```

        break
    }
}

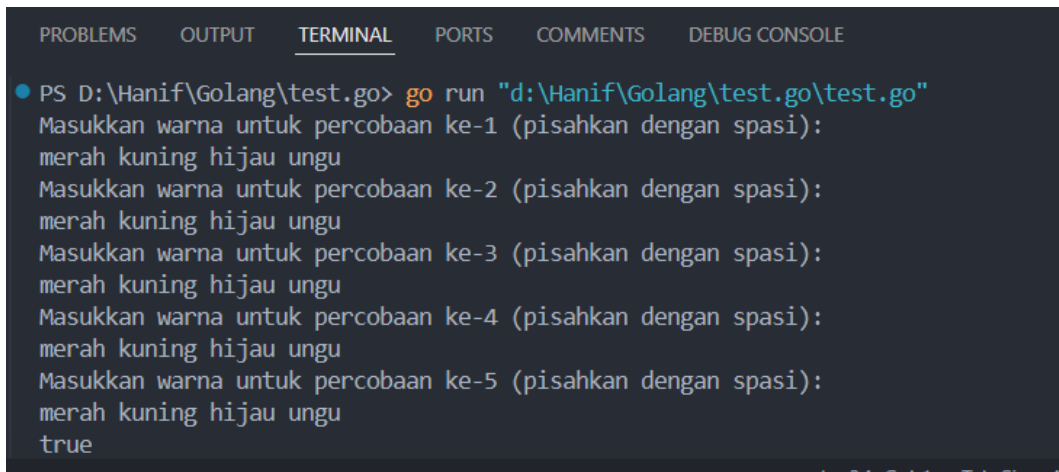
if !hasilPercobaan {
    break
}

}

fmt.Println(hasilPercobaan)
}

```

Screenshoot Output



```

PROBLEMS  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE
● PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
Masukkan warna untuk percobaan ke-1 (pisahkan dengan spasi):
merah kuning hijau ungu
Masukkan warna untuk percobaan ke-2 (pisahkan dengan spasi):
merah kuning hijau ungu
Masukkan warna untuk percobaan ke-3 (pisahkan dengan spasi):
merah kuning hijau ungu
Masukkan warna untuk percobaan ke-4 (pisahkan dengan spasi):
merah kuning hijau ungu
Masukkan warna untuk percobaan ke-5 (pisahkan dengan spasi):
merah kuning hijau ungu
true

```

Deskripsi Program

Program di atas adalah program go yang memeriksa apakah urutan warna yang dimasukkan oleh pengguna sebanyak 5 kali percobaan sesuai dengan urutan warna yang diharapkan, yaitu "merah", "kuning", "hijau", dan "ungu". Pada setiap percobaan, pengguna diminta untuk memasukkan empat warna, dan program akan membandingkan setiap warna yang dimasukkan dengan urutan yang diharapkan. Jika ada satu saja percobaan yang tidak sesuai, variabel `hasilPercobaan` akan diset menjadi `false` dan program segera berhenti memeriksa percobaan lainnya. Di akhir, program mencetak `true` jika semua percobaan berhasil, dan `false` jika ada satu atau lebih percobaan yang tidak sesuai.

UNGUIDED 4

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. Tampilkan isi pita setelah proses input selesai. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read) 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 dibawah ini (teks bergaris bawah adalah input/read):

Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

func main() {
    var pita string
    var count int
    var limit int

    fmt.Print("Masukkan jumlah bunga yang ingin ditambahkan: ")
    fmt.Scan(&limit)

    for count < limit {
        var bunga string
        fmt.Printf("Bunga %d: ", count+1)
        fmt.Scan(&bunga)

        if strings.ToUpper(bunga) == "SELESAI" {
            break
        }

        if count == 0 {
            pita += bunga
        } else {
            pita += " - " + bunga
        }
        count++
    }

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

Screenshoot Output

```
PROBLEMS  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE

PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
Masukkan jumlah bunga yang ingin ditambahkan: 5
Bunga 1: mawar
Bunga 2: melati
Bunga 3: anggrek
Bunga 4: tulip
Bunga 5: kertas
Pita: mawar - melati - anggrek - tulip - kertas
Bunga 3: anggrek
Bunga 4: tulip
Bunga 5: kertas
Pita: mawar - melati - anggrek - tulip - kertas
Jumlah Bunga: 5
PS D:\Hanif\Golang\test.go>
```

Deskripsi Program

Program di atas adalah program dari go memungkinkan pengguna untuk memasukkan daftar nama bunga hingga mencapai jumlah yang diinginkan atau hingga pengguna mengetikkan "SELESAI" untuk menghentikan proses input lebih awal. Setelah pengguna memasukkan jumlah batas bunga yang diinginkan, program akan memulai pengambilan input secara iteratif, menyimpan nama setiap bunga yang dimasukkan dalam variabel `pita` dengan tanda pemisah " - ". Jika pengguna mengetik "SELESAI", program akan menghentikan pengambilan input dan menampilkan string `pita` yang berisi daftar bunga beserta jumlah total bunga yang dimasukkan. Program ini menggunakan penggabungan string untuk membentuk daftar bunga dan case-insensitive check untuk perintah "SELESAI".

UNGUIDED 5

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. Tampilkan isi pita setelah proses input selesai. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read) 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 dibawah ini (teks bergaris bawah adalah input/read):

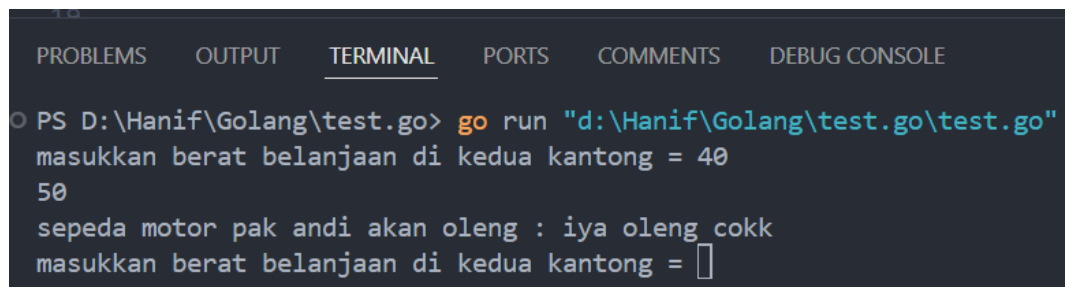
Sourcecode

```
package main

import (
    "fmt"
)

func main() {
    var berat1, berat2 float64
    for {
        fmt.Print("masukkan berat belanjaan di kedua kantong = ")
        fmt.Scan(&berat1, &berat2)
        if berat1 >= 9 || berat2 >= 9 {
            fmt.Println("sepeda motor pak andi akan oleng : iya oleng cekk")
        } else {
            fmt.Println("sepeda motor pak andi akan oleng : aman dan tidak oleng")
        }
    }
}
```

Screenshoot Output



```
PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
masukkan berat belanjaan di kedua kantong = 40
50
sepeda motor pak andi akan oleng : iya oleng cekk
masukkan berat belanjaan di kedua kantong = 
```

Deskripsi Program

Program ini memeriksa kestabilan sepeda motor Pak Andi berdasarkan berat belanjaan yang dimasukkan di dua kantong. Program secara terus-menerus meminta pengguna untuk memasukkan berat belanjaan pada kedua kantong. Jika salah satu kantong memiliki berat 9 kg atau lebih, program akan mencetak pesan bahwa sepeda motor Pak Andi akan oleng. Jika berat kedua kantong di bawah 9 kg, program mencetak pesan bahwa sepeda motor aman dan tidak oleng. Program berjalan dalam loop tak terbatas, sehingga pengguna dapat terus memasukkan berat baru dan melihat hasilnya sampai program dihentikan secara manual.

UNGUIDED 6

buatlah 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 dibawah ini (teks bergaris bawah adalah input/read):

Sourcecode

```
package main

import (
    "fmt"
    "math"
)

func hitungF(k int) float64 {
    pembilang := math.Pow(float64(4*k+2), 2)
    penyebut := float64((4*k + 1) * (4*k + 3))
    return pembilang / penyebut
}

func hitungAkarDua(k int) float64 {
    hasil := 1.0
    for i := 0; i <= k; i++ {
        hasil *= hitungF(i)
    }
    return hasil
}

func main() {
    var k int
    fmt.Print("Masukkan nilai K: ")
    fmt.Scan(&k)

    nilaiF := hitungF(k)
    fmt.Printf("Nilai f(%d) = %.10f\n", k, nilaiF)

    akarDua := hitungAkarDua(k)
    fmt.Printf("Nilai mendekati akar 2 untuk K = %d: %.10f\n", k,
    akarDua)
}
```

Screenshoot Output

```
PROBLEMS  OUTPUT  TERMINAL  PORTS  COMMENTS  DEBUG CONSOLE

● PS D:\Hanif\Golang\test.go> go run "d:\Hanif\Golang\test.go\test.go"
Masukkan nilai K: 10
Nilai f(10) = 1.0005672150
Nilai mendekati akar 2 untuk K = 10: 1.4062058441
○ PS D:\Hanif\Golang\test.go> 
```

Deskripsi Program

Program ini bertujuan untuk menghitung nilai suatu fungsi $f(k)$ dan mendekati nilai akar dari 2 menggunakan deret produk dari fungsi tersebut. Fungsi `hitungF(k)` menerima nilai integer k sebagai parameter dan mengembalikan hasil perhitungan $f(k) = (4k+1)(4k+3)(4k+2)^2$, yang merupakan bagian dari deret produk. Fungsi `hitungAkarDua(k)` menggunakan perhitungan $f(k)$ untuk mengalikan hasil dari setiap $f(i)$, di mana i adalah nilai yang bervariasi dari 0 hingga k , menghasilkan nilai yang mendekati akar 2. Dalam fungsi `main()`, pengguna diminta untuk memasukkan nilai k , kemudian program akan menghitung dan mencetak nilai $f(k)$ serta hasil perkalian dari deret tersebut untuk mendekati akar 2. Hasilnya ditampilkan dengan format desimal hingga sepuluh tempat, yang memberikan presisi yang cukup baik dalam pendekatan akar 2 berdasarkan nilai k yang dimasukkan.