

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

Rendi Widya Anggita/2311102278

S1IF-11-06

Dosen Pengampu :

ABEDNEGO DWI SEPTIADI

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Struktur kontrol merupakan elemen penting dalam pemrograman yang digunakan untuk mengatur alur program berdasarkan kondisi tertentu atau untuk mengulangi beberapa pernyataan. Pada bahasa pemrograman Go (Golang), struktur kontrol berfungsi agar program dapat mengambil keputusan atau mengontrol bagaimana dan kapan bagian tertentu dari kode dijalankan.

Dalam bahasa go, program utama selalu mempunyai dua komponen, yaitu *package main* dan *func main()*.

- Package main merupakan tanda bahwa file tersebut merupakan program utama
- Func main() berisi kode utama dari sebuah program go

Jenis-Jenis Struktur Kontrol di Golang

1. Percabangan

Dalam Golang, percabangan dapat dilakukan menggunakan pernyataan *if*, *else if*, dan *else*, serta menggunakan *switch*. Berikut adalah penjelasan masing-masing:

If-Else : Pernyataan ini digunakan untuk mengeksekusi program ketika kondisi tertentu terpenuhi. Jika kondisi tidak terpenuhi, maka program di dalam else akan dieksekusi.

Contoh :

```
if x > 0 {  
    fmt.Println("x adalah bilangan positif")  
} else {  
    fmt.Println("x adalah bilangan negatif")  
}
```

Switch : Pernyataan ini digunakan untuk memilih salah satu dari beberapa program untuk dieksekusi. Golang memiliki keunikan pada switch, yaitu penggunaan keyword fallthrough, yang memungkinkan eksekusi berlanjut ke case berikutnya meskipun kondisinya tidak terpenuhi.

Contoh :

```
switch {  
  case x < 0:  
    fmt.Println("x negatif")  
  case x == 0:  
    fmt.Println("x nol")  
  case x > 0:  
    fmt.Println("x positif")  
    fallthrough  
  default:  
    fmt.Println("Ini adalah default case")  
}
```

2. Perulangan

Perulangan di Golang biasanya dilakukan dengan menggunakan pernyataan for.

Contoh :

```
for i := 0; i < 10; i++ {  
  fmt.Println(i)  
}
```

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 yang sebenarnya yang dilakukan program tersebut.

```
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)
}
```

Sourcecode

```
package main

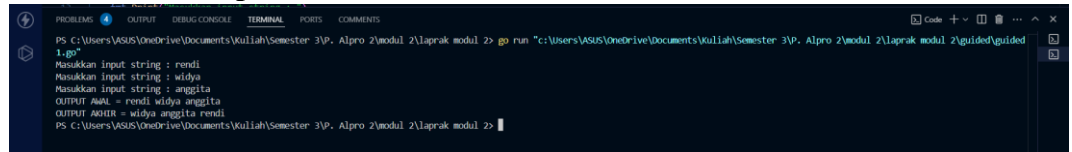
import "fmt"

func main() {

    var satu, dua, tiga, temp string

    fmt.Print("Masukkan input string : ")
    fmt.Scanln(&satu)
    fmt.Print("Masukkan input string : ")
    fmt.Scanln(&dua)
    fmt.Print("Masukkan 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)
}
```

Screenshot Output

A screenshot of a Go IDE terminal window. The terminal shows the execution of a Go program. The user enters three input strings: 'rendi', 'widya', and 'anggita'. The program outputs 'OUTPUT AWAL = rendi widya anggita' and 'OUTPUT AKHIR = widya anggita rendi', demonstrating a string swap operation. The terminal path is 'PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2>'.

```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\guided\guided1.go"
Masukkan input string : rendi
Masukkan input string : widya
Masukkan input string : anggita
OUTPUT AWAL = rendi widya anggita
OUTPUT AKHIR = widya anggita rendi
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2>
```

Deskripsi Program

Program di atas bertujuan untuk membaca tiga input string dari pengguna, menampilkannya dalam urutan awal, lalu menukar posisi string tersebut dan menampilkannya kembali dalam urutan yang telah diubah.

Pertama program mendeklarasikan 4 variabel yaitu satu,dua,tiga, dan temp yang bertipe data string. Variabel ini akan menyimpan sementara inputan dari user. Program kemudian meminta pengguna untuk memasukkan tiga input string satu per satu.

Pertama, pengguna diminta memasukkan nilai string pertama dan disimpan dalam variabel satu. Lalu, pengguna diminta memasukkan string kedua yang disimpan di variabel dua. Terakhir, pengguna memasukkan string ketiga yang disimpan di variabel tiga.

Setelah pengguna memasukkan ketiga input, program menampilkan hasilnya pada OUTPUT AWAL, di mana ketiga string yang diinput ditampilkan berurutan sesuai dengan urutan input pengguna.

Program kemudian melakukan penukaran posisi string. Nilai yang ada di satu disimpan sementara ke dalam variabel temp. Nilai dari dua dipindahkan ke satu. Nilai dari tiga dipindahkan ke dua. Nilai yang ada di temp (yang awalnya nilai satu) dipindahkan ke tiga. Setelah itu program akan menampilkan kembali inputan tersebut dengan urutan yang sudah diubah pada OUTPUT AKHIR

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).

1	Tahun: <u>2016</u> Kabisat: true
2	Tahun: <u>2000</u> Kabisat: true
3	Tahun: <u>2018</u> Kabisat: false

Sourcecode

```
package main

import "fmt"

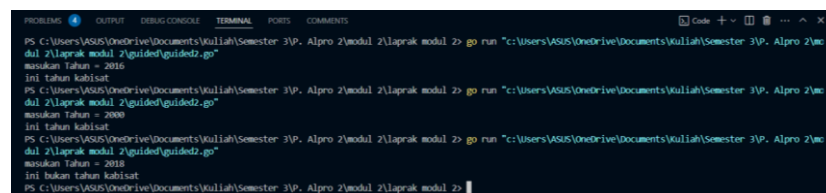
func main() {

    var tahun int

    fmt.Print("masukan Tahun = ")
    fmt.Scanln(&tahun)

    if tahun%400 == 0 {
        fmt.Println("ini tahun kabisat")
    } else if tahun%100 == 0 {
        fmt.Println("ini bukan tahun kabisat")
    } else if tahun%4 == 0 {
        fmt.Println("ini tahun kabisat")
    } else {
        fmt.Print("ini bukan tahun kabisat")
    }
}
```

Screenshoot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\guided2.go"
masukan Tahun = 2016
ini tahun kabisat
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\guided2.go"
masukan Tahun = 2000
ini tahun kabisat
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\guided2.go"
masukan Tahun = 2018
ini bukan tahun kabisat
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2>
```

Deskripsi Program

Program di atas berfungsi untuk memeriksa apakah sebuah tahun yang dimasukkan oleh pengguna adalah tahun kabisat atau bukan.

Pertama program akan meminta pengguna untuk memasukkan sebuah angka, yaitu tahun yang ingin diperiksa. Tahun tersebut disimpan dalam variabel bernama tahun, yang bertipe int (bilangan bulat). Kemudian program akan mengecek inputan dari pengguna dengan beberapa kondisi.

Kondisi pertama: Jika tahun tersebut bisa dibagi habis dengan 400, maka program akan mencetak "ini tahun kabisat".

Kondisi kedua: Apakah tahun tersebut bisa dibagi habis dengan 100. Jika iya, program akan mencetak "ini bukan tahun kabisat".

Kondisi ketiga: Jika kedua kondisi di atas tidak terpenuhi, program akan memeriksa apakah tahun tersebut bisa dibagi habis dengan 4. Jika iya, maka program akan mencetak "ini 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 $\text{luasbola} = 4\pi r^2$ ($\pi = 3.1415926535$).



Jejari = 5
Bola dengan jejari 5 memiliki volume 523.5988 dan luas kulit 314.1593

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("Bola dengan jejaringnya %.0f memiliki volume %.4f dan luas kulit %.4f\n", jarijari, volume, luasPermukaan)
}
```

Screenshoot Output

A screenshot of a Go IDE's terminal window. The terminal shows the execution of a Go program. The prompt is 'PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\p. Alpro 2\modul 2\laprak modul 2>'. The user enters 'go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\p. Alpro 2\modul 2\laprak modul 2\guided\guided3.go"'. The output is: 'Masukkan jari-jari bola : 5', 'bola dengan jari-jari 5 memiliki volume 523.5988 dan luas kulit 314.1593'. The prompt returns to 'PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\p. Alpro 2\modul 2\laprak modul 2>'.

```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\p. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\p. Alpro 2\modul 2\laprak modul 2\guided\guided3.go"
Masukkan jari-jari bola : 5
bola dengan jari-jari 5 memiliki volume 523.5988 dan luas kulit 314.1593
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\p. Alpro 2\modul 2\laprak modul 2>
```

Deskripsi Program

Program di atas berfungsi untuk menghitung volume dan luas permukaan bola berdasarkan nilai jari-jari yang dimasukkan oleh pengguna.

Pertama program meminta pengguna untuk memasukkan nilai jari-jari bola. Nilai tersebut disimpan dalam variabel jarijari yang bertipe float64. Kemudian program menghitung volume bola menggunakan rumus volume bola di mana r adalah jari-jari bola. Kemudian program menghitung luas permukaan bola.

Lalu program menampilkan hasil volume dan luas permukaan dalam format yang telah ditentukan.

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

    fahrenheit := (celsius * 9 / 5) + 32

    reamur := celsius * 4 / 5

    kelvin := celsius + 273.15

    fmt.Printf("Temperatur Celsius: %.0f\n", celsius)
    fmt.Printf("Derajat Reamur: %.0f\n", reamur)
    fmt.Printf("Derajat Fahrenheit: %.0f\n", fahrenheit)
    fmt.Printf("Derajat Kelvin: %.0f\n", kelvin)
}
```

Screenshoot Output

```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2>
Vuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2\unguided\unguided.go"
Masukkan suhu dalam derajat Celsius: 50
Temperatur Celsius: 50
Derajat Reamur: 40
Derajat Fahrenheit: 122
Derajat Kelvin: 323
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2> |
```

Deskripsi Program

Program di atas adalah program golang untuk mengkonversi suhu celcius ke fahrenheit, reamur, dan kelvin dengan inputan pengguna dalam format celcius.

Pertama program meminta pengguna untuk memasukkan nilai suhu dalam derajat Celsius. Input ini disimpan dalam variabel bertipe float64 bernama celsius. Kemudian program melakukan perhitungan konversi ke tiga suhu lainnya sesuai dengan rumus yang ada. Setelah semua konversi selesai, program akan menampilkan output dalam suhu Celsius, Fahrenheit, Reamur, dan Kelvin.

2. Soal 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 int1, int2, int3, int4, int5 int
    var var1, var2, var3 byte

    // Input 5 angka integer
    fmt.Print("Masukkan 5 buah data integer: ")
    fmt.Scanf("%d %d %d %d %d", &int1, &int2, &int3, &int4,
    &int5)

    // Membersihkan newline yang tersisa dari input
    sebelumnya
    fmt.Scanf("%c", new(byte))

    // Input 3 karakter tanpa spasi
```

```

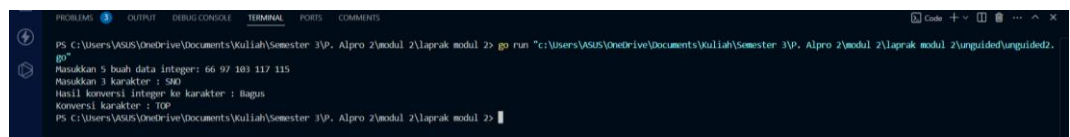
fmt.Print("Masukkan 3 karakter : ")
fmt.Scanf("%c", &var1) // Input karakter pertama
fmt.Scanf("%c", &var2) // Input karakter kedua
fmt.Scanf("%c", &var3) // Input karakter ketiga

// Mencetak hasil konversi integer ke karakter (ASCII)
fmt.Printf("Hasil konversi integer ke karakter :
%c%c%c%c%c\n", int1, int2, int3, int4, int5)

// Mencetak karakter setelah karakter yang diinput
fmt.Printf("Konversi karakter : %c%c%c\n", var1+1,
var2+1, var3+1)
}

```

Screenshoot Output



```

PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2\angguides\angguides2.
go
Masukkan 5 buah data integer: 66 97 103 117 115
Masukkan 3 karakter : SBO
Hasil konversi integer ke karakter : Bagus
Konversi karakter : TOP
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2>

```

Deskripsi Program

Program ini mengimplementasikan operasi dasar untuk membaca input integer dan karakter, serta mengonversi integer ke karakter menggunakan representasi tabel ASCII yang menghasilkan dua baris output: satu yang berisi karakter hasil konversi dari integer dan satu lagi yang berisi karakter yang diinput.

Pertama program meminta pengguna untuk memasukkan 5 buah data integer yang nilainya berada dalam rentang antara 32 hingga 127 (sesuai dengan tabel ASCII). Data ini disimpan dalam variabel bertipe int(int1, int2, dst). Setelah itu, program meminta pengguna untuk memasukkan 3 buah karakter tanpa spasi. Karakter ini disimpan dalam variabel bertipe rune(char1, char2, dan char3).

Program kemudian mengonversi data integer yang dimasukkan ke dalam karakter ASCII dengan menggunakan format %c di fungsi Printf. Hasilnya dicetak dalam satu baris tanpa spasi. Kemudian, karakter yang diinput oleh pengguna dicetak dalam satu baris tanpa spasi.

3. 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 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.

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

Percobaan 1:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 2:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 3:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 4:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 5:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
BERHASIL:	true			
Percobaan 1:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 2:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 3:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
Percobaan 4:	<u>ungu</u>	<u>kuning</u>	<u>hijau</u>	<u>merah</u>
Percobaan 5:	<u>merah</u>	<u>kuning</u>	<u>hijau</u>	<u>ungu</u>
BERHASIL:	false			

Sourcecode

```
package main

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

func main() {
    // Urutan warna yang benar
    correctOrder := []string{"merah", "kuning", "hijau",
"ungu"}

    // Membaca input dari pengguna sebanyak 5 kali percobaan
    scanner := bufio.NewScanner(os.Stdin)
    var allCorrect bool = true

    for i := 1; i <= 5; i++ {
        fmt.Printf("Percobaan %d: ", i)
        scanner.Scan()
        input := scanner.Text()

        // Memecah input menjadi array warna
        colors := strings.Split(input, " ")

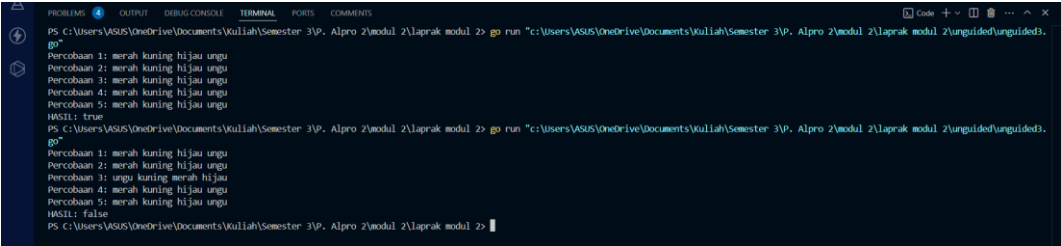
        // Cek apakah input sesuai dengan urutan yang benar
        if len(colors) != 4 {
            fmt.Println("Input tidak valid, harap masukkan 4
warna.")
            allCorrect = false
            continue
        }

        for j, color := range colors {
            if color != correctOrder[j] {
                allCorrect = false
                break
            }
        }
    }
}
```

```
    }
  }
}

// Output hasil setelah 5 percobaan
if allCorrect {
    fmt.Println("HASIL: true")
} else {
    fmt.Println("HASIL: false")
}
}
```

Screenshoot Output



Deskripsi Program

Program ini berfungsi untuk mengecek input warna dari 4 gelas reaksi sebanyak 5 kali percobaan. Setiap inputan warna akan dibandingkan dengan urutan warna yang benar (merah, kuning, hijau, ungu). Jika semua percobaan memiliki urutan warna yang benar, maka akan mencetak true. Jika ada percobaan yang tidak sesuai, maka akan mencetak false.

4. Soal 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 – teratai – kamboja – anggrek

23 | 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.

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

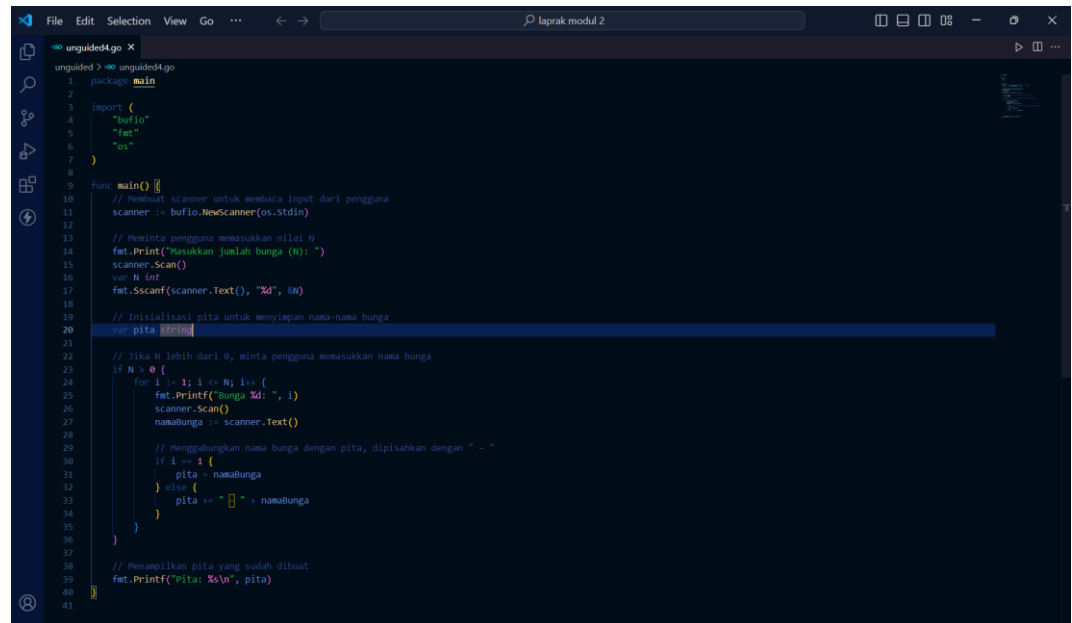
N: 3 Bunga 1: Kertas Bunga 2: Mawar Bunga 3: Tulip Pita: Kertas – Mawar – Tulip –	N : 0 Pita :
---	-----------------

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 di bawah ini (teks bergaris bawah adalah input/read):

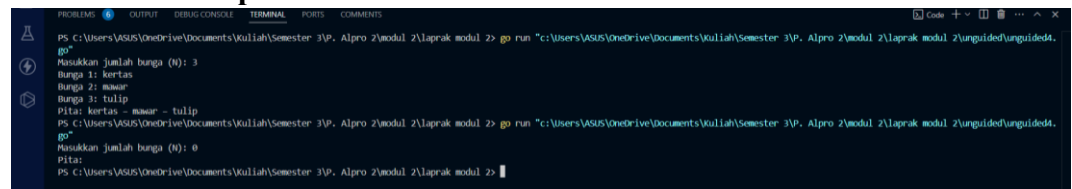
Bunga 1: Kertas Bunga 2: Mawar Bunga 3: Tulip Bunga 4: SELESAI Pita: Kertas – Mawar – Tulip – Bunga: 3	Bunga 1: SELESAI Pita : Bunga: 0
---	--

Sourcecode



```
1 package main
2
3 import (
4     "bufio"
5     "fmt"
6     "os"
7 )
8
9 func main() {
10     // Membuat scanner untuk membaca input dari pengguna
11     scanner := bufio.NewScanner(os.Stdin)
12
13     // Meminta pengguna memasukkan nilai N
14     fmt.Print("Masukkan jumlah bunga (N): ")
15     scanner.Scan()
16     var N int
17     fmt.Sscanf(scanner.Text(), "%d", &N)
18
19     // Inisialisasi pita untuk menyimpan nama-nama bunga
20     var pita string
21
22     // Jika N lebih dari 0, minta pengguna memasukkan nama bunga
23     if N > 0 {
24         for i := 1; i <= N; i++ {
25             fmt.Printf("Bunga %d: ", i)
26             scanner.Scan()
27             namaBunga := scanner.Text()
28
29             // Menggabungkan nama bunga dengan pita, dipisahkan dengan " - "
30             if i == 1 {
31                 pita = namaBunga
32             } else {
33                 pita += " - " + namaBunga
34             }
35         }
36     }
37
38     // Menampilkan pita yang sudah dibuat
39     fmt.Printf("Pita: %s\n", pita)
40 }
41
```

Screenshot Output



```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\unguided\unguided4.go"
Masukkan jumlah bunga (N): 3
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Pita: kertas - mawar - tulip
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\unguided\unguided4.go"
Masukkan jumlah bunga (N): 0
Pita:
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2>
```

Deskripsi Program

Program ini membaca jumlah bunga N yang diinput oleh pengguna. Kemudian, program meminta pengguna memasukkan nama bunga sebanyak N kali. Semua nama bunga tersebut akan digabungkan menjadi satu string (pita) dengan tanda " - " sebagai pemisah antar bunga. Jika N sama dengan 0, program akan mencetak pita kosong.

Note : setelah modifikasi

Sourcecode

```
package main

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

func main() {
    // Membuat scanner untuk membaca input dari pengguna
    scanner := bufio.NewScanner(os.Stdin)

    // Inisialisasi pita untuk menyimpan nama-nama bunga
    var pita string
    jumlahBunga := 0

    // Meminta input bunga secara berulang hingga pengguna
    // mengetik "SELESAI"
    for {
        fmt.Printf("Bunga %d: ", jumlahBunga+1)
        scanner.Scan()
        namaBunga := scanner.Text()

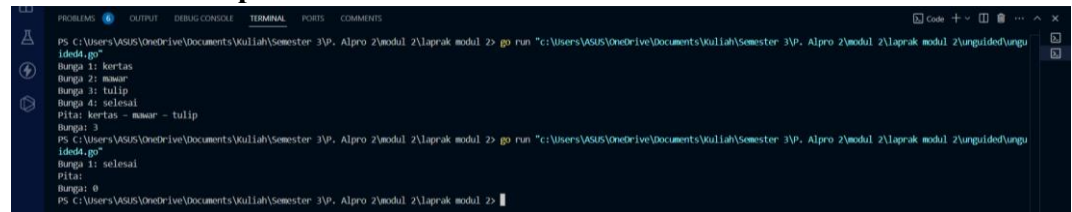
        // Jika pengguna mengetik "SELESAI", hentikan input
        if namaBunga == "SELESAI" || namaBunga == "selesai">{
            break
        }

        // Menggabungkan nama bunga dengan pita, dipisahkan
        // dengan " - "
        if jumlahBunga == 0 {
            pita = namaBunga
        } else {
            pita += " - " + namaBunga
        }

        // Menambah jumlah bunga
        jumlahBunga++
    }

    // Menampilkan pita yang sudah dibuat dan jumlah bunga
    // yang dimasukkan
    fmt.Printf("Pita: %s\n", pita)
    fmt.Printf("Bunga: %d\n", jumlahBunga)
}
```

Screenshoot Output



```
PS C:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\unguided\fungsi\idada.go"
Bunga 1: kertas
Bunga 2: mawar
Bunga 3: tulip
Bunga 4: selesai
Pita: kertas - mawar - tulip
Bunga: 3
PS C:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2\unguided\fungsi\idada.go"
Bunga 1: selesai
Pita:
Bunga: 0
PS C:\Users\VASUS\OneDrive\Documents\Kuliah\Semester 3\P. Alpro 2\modul 2\laprak modul 2>
```

Deskripsi Program

Program diatas adalah modifikasi dari program sebelumnya, dimana program akan berhenti ketika pengguna menginputkan kata “selesai”, tanpa perlu menentukan bunga sebelumnya, lalu setiap bunga yang diinputkan akan digabung menjadi sebuah pita. Kemudian program akan menampilkan nama bunga dalam pita, dan juga jumlah total bunga yang sudah dimasukkan.

5. Soal 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 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.

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

```
Masukan berat belanjaan di kedua kantong: 5.5 1.0
Masukan berat belanjaan di kedua kantong: 7.1 8.5
Masukan berat belanjaan di kedua kantong: 2 6
Masukan berat belanjaan di kedua kantong: 9 5.8
Proses selesai.
```

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.

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

```
Masukan berat belanjaan di kedua kantong: 5 10
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 55.6 70.2
Sepeda motor pak Andi akan oleng: true
Masukan berat belanjaan di kedua kantong: 72.3 66.9
Sepeda motor pak Andi akan oleng: false
Masukan berat belanjaan di kedua kantong: 59.5 98.7
Proses selesai.
```



```
package main

import (
    "fmt"
    "math"
)

func main() {
    var kantong1, kantong2 float64
    for {
        fmt.Print("Masukan berat belanjaan di kedua kantong: ")
        fmt.Scanln(&kantong1, &kantong2)

        total := kantong1 + kantong2

        if total > 150 {
            fmt.Println("Total berat tidak boleh melebihi 150 kg.")
            fmt.Println("Program Selesai")
            break // Keluar dari loop dan program selesai
        } else if kantong1 < 0 || kantong2 < 0 {
            fmt.Println("Berat tidak boleh negatif.")
        } else if math.Abs(kantong1-kantong2) < 9 {
            fmt.Println("Sepeda motor akan oleng: false")
        } else {
            fmt.Println("Sepeda motor akan oleng: true")
        }
    }
}
```

```
PS C:\Users\ASUS\OneDrive\Documents\Ukiah\Semester 3\IP Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Ukiah\Semester 3\IP Alpro 2\modul 2\laprak modul 2\UnggulidungSungdes.go"
Masukan berat belanjaan di kedaa Kantong: 5 10
Sepeda motor akan oleng: false
Masukan berat belanjaan di kedaa Kantong: 55.6 70.2
Sepeda motor akan oleng: true
Masukan berat belanjaan di kedaa Kantong: 72.3 66.9
Sepeda motor akan oleng: false
Masukan berat belanjaan di kedaa Kantong: 59.5 98.7
Total berat tidak boleh melebihi 150 kg.
Program Selesai.
```

Program di atas adalah program untuk mengecek apakah sepeda motor akan oleng berdasarkan selisih berat antara dua kantong belanjaan. Pengguna

diminta memasukkan berat untuk kantong 1 dan kantong 2, kemudian program menghitung total beratnya. Jika total berat melebihi 150 kg, program akan berhenti dan memberi peringatan. Jika salah satu berat negatif, program juga akan memberi peringatan namun tetap melanjutkan.

Program menggunakan fungsi `math.Abs` untuk menghitung selisih berat kedua kantong. Jika selisihnya kurang dari 9 kg, motor tidak akan oleng (`false`), dan jika lebih dari atau sama dengan 9 kg, motor akan oleng (`true`). Proses ini berulang hingga total berat melebihi batas 150 kg.

6. Soal Studi Case

Diberikan sebuah persamaan sebagai berikut ini.

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

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 fK(K float64) float64 {
    numerator := (4*K + 2) * (4*K + 2)
    denominator := (4*K + 1) * (4*K + 3)
    return numerator / denominator
}

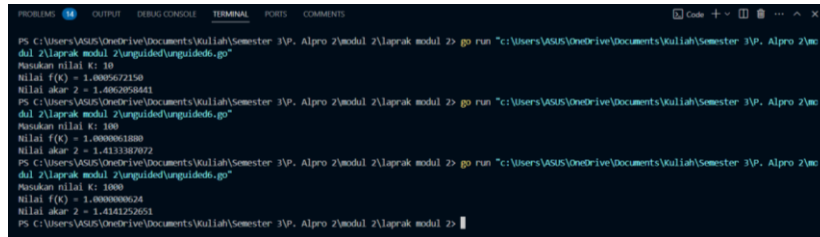
func sqrt2_approx(K int) float64 {
    result := 1.0
    for i := 0; i <= K; i++ {
        numerator := (4*float64(i) + 2) * (4*float64(i) + 2)
        denominator := (4*float64(i) + 1) * (4*float64(i) +
3)
        result *= numerator / denominator
    }
    return result
}

func main() {
    var K float64
    fmt.Print("Masukan nilai K: ")
    fmt.Scan(&K)

    resultF := fK(K)
    fmt.Printf("Nilai f(K) = %.10f\n", resultF)

    Kint := int(math.Round(K))
    resultSqrt2 := sqrt2_approx(Kint)
    fmt.Printf("Nilai akar 2 = %.10f\n", resultSqrt2)
}
```

Screenshoot Output

A screenshot of a terminal window showing the execution of a Go program. The terminal has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', 'PORTS', and 'COMMENTS'. The 'TERMINAL' tab is active. The command prompt shows the user is in the directory 'C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2'. The program is run with the command 'go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2\unguided\unguidede.go"'. The output shows three iterations of a loop where the user inputs a value 'K' and the program calculates 'f(K)' and 'akar 2'. The results are displayed with 10 decimal places.

```
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2\unguided\unguidede.go"
Masukan nilai K: 10
nilai f(K) = 1.0005672150
nilai akar 2 = 1.4062598441
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2\unguided\unguidede.go"
Masukan nilai K: 100
nilai f(K) = 1.0000061880
nilai akar 2 = 1.4142135624
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2> go run "c:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2\unguided\unguidede.go"
Masukan nilai K: 1000
nilai f(K) = 1.0000000024
nilai akar 2 = 1.4142135624
PS C:\Users\ASUS\OneDrive\Documents\Kuliah\Semester 3\Pro. Alpro 2\modul 2\laprak modul 2>
```

Deskripsi Program

Program ini dibuat dalam bahasa Go untuk menghitung dua hal: nilai fungsi matematika berdasarkan input pengguna dan perkiraan nilai akar dua.

Pertama, program meminta pengguna memasukkan nilai bilangan bulat K. Nilai ini kemudian digunakan untuk menghitung fungsi matematika tertentu. Setelah melakukan perhitungan, hasilnya ditampilkan dalam bentuk angka desimal dengan presisi tinggi.

Selain itu, program juga melakukan perhitungan untuk mendekati nilai akar dua. Caranya adalah dengan menjalankan sebuah perulangan (loop) dari 0 hingga nilai K, di mana setiap langkah perhitungan ditambahkan ke hasil akhir yang disebut hitungfk. Nilai ini mewakili perkiraan akar dua.

Di akhir program, dua hasil ditampilkan: nilai fungsi matematika dan nilai perkiraan akar dua, keduanya dengan 10 angka di belakang koma. Program ini sederhana karena hanya membutuhkan input, menjalankan beberapa perhitungan, dan menampilkan hasilnya.