

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 2
REVIEW STRUKTUR KONTROL**



Disusun Oleh :

PRIESTY AMEILIANA MAULIDAH / 2311102175

IF-11-05

Dosen Pengampu :

ARIF AMRULLOH, S.KOM.,M.KOM

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

1.) DASAR TEORI

Struktur program go

Program utama selalu mempunyai dua komponen berikut :

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

1.)Koding,kompilasi,dan eksekusi go

Koding

- Tidak berbeda dengan penulisan program sumber dalam bahasa lain,program go harus dibuat menggunakan penyunting teks dan disimpan dalam format teks, bukan dalam format dokumen(doc,docx,atau lainnya).
- Setiap program go disimpan dalam file teks dengan ekstensi*.go, dengan nama bebas,sebagainya nama file adalah nama untuk program tersebut.
- Setiap satu program lengkap go disimpan dalam satu folder tersendiri.

Kompilasi

Beberapa Bahasa pemrogram dirancang untuk diimplemenrasikan sebagai interpreter dan lainnya sebagai kompilator.

Tipe data dan instruksi dasar

1.) Data dan variabel

Variabel adalah nama dari suatu lokasi dimemori,yang data dengan tipe tertentu dapat disimpan.

- Nama variabel dimulai dengan huruf dan dapat diikuti dengan sejumlah huruf,angka,atau garis bawah.
- Tipe data yang umum tersedia adalah integer,real,Boolean,karakter,dan string.
- Nilai data yang disimpan dalam variabel dapat diperoleh dengan menyebutkan langsung nama variabel.
- Informasi alamat atau lokasi dari variabel dapat diperoleh dengan menambahkan prefiks& didepan nama variabel

- Jika variabel berisi alamat memori ,prefiks*pada variabel tersebut akan memberikan nilai yang tersimpan dalam memori yang lokasinya disimpan dalam variabel tersebut.
- Bahasa go menganut kesesuaian tipe data yang ketat,tipe data yang berbeda tidak boleh dicampur dalam satu ekspresi,bahkan tipe data masih yang sejenis,misalnya masih sama-sama sama interger(int dan int32).
- Variabel harus dideklarasikan dulu sebelum digunakan,variabel juga harus dianalisis dulu(diisi data) agar nilai yang tersimpan diketahui dengan jelas eksekusi algoritma menjadi terprediksi.

2.) Instruksi dasar

3.) Konstanta simbolik

Konstanta dapat diberi nama untuk memudahkan mengingat maksud dan manfaat dari nilai yang diberi nama tersebut.

2.) GUIDED

Soal Studi Case

1. 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.

Sourcecode

```
modul 2 > -o helago > ...
1  package main
2
3  import (
4      "bufio"
5      "fmt"
6      "os"
7      "strings"
8  )
9
10 func main() {
11     // Urutan warna yang benar
12     correctOrder := []string{"merah", "kuning", "hijau", "ungu"}
13
14     // Membaca input untuk 5 percobaan
15     reader := bufio.NewReader(os.Stdin)
16     success := true
17
18     for i := 1; i <= 5; i++ {
19         fmt.Printf("Percobaan %d: ", i)
20
21         // Membaca input dari pengguna
22         input, _ := reader.ReadString('\n')
23         input = strings.TrimSpace(input)
24
25         // Memisahkan input berdasarkan spasi
26         colors := strings.Split(input, " ")
27
28         // Mengecek apakah urutan warna sesuai
29         for j := 0; j < 4; j++ {
30             if colors[j] != correctOrder[j] {
31                 success = false
32                 break
33             }
34         }
35
36         // Jika ada percobaan yang tidak sesuai, keluar dari loop
37         if !success {
38             break
39         }
40     }
41
42     // Menampilkan hasil
43     if success {
44         fmt.Println("BERHASIL: true")
45     } else {
46         fmt.Println("BERHASIL: false")
47     }
48 }
```

Screenshoot Output

```
PS D:> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\hello.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\hello.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
BERHASIL: false
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

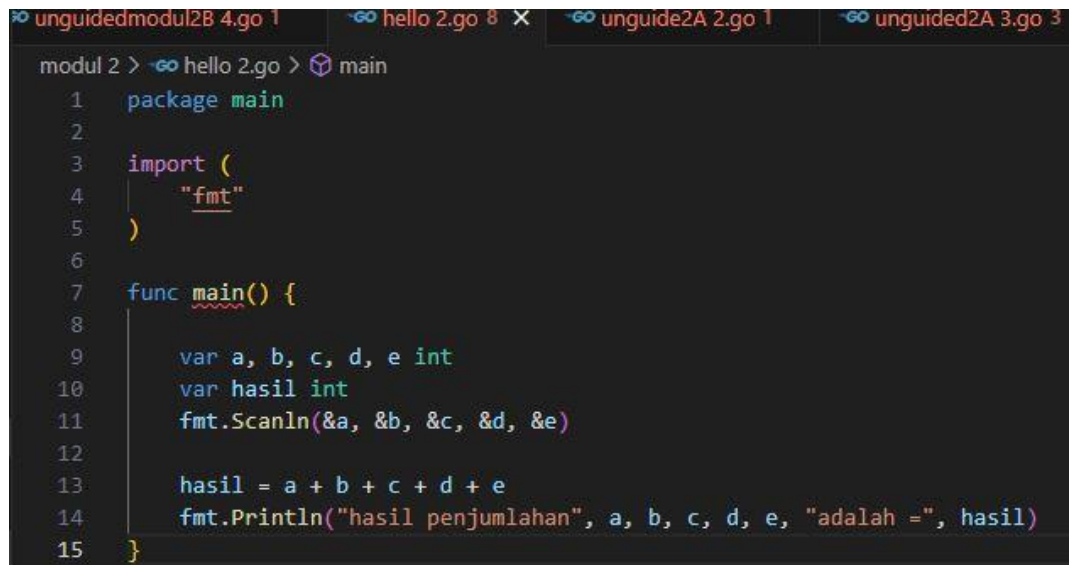
Inisialisasi urutan warna: Program menyimpan urutan warna dalam array `correctOrder`: "merah", "kuning", "hijau", "ungu". Pengguna diminta memasukkan urutan warna lima kali. Setiap input diambil dengan `bufio.NewReader` dan disimpan dalam variabel input. Memproses input: Input pengguna akan dihilangkan karakter newline dan dipisah menjadi array string berdasarkan spasi. Split. Pengecekan urutan warna: Program memeriksa apakah urutan warna yang dimasukkan sesuai dengan yang benar setelah pemecahan input. Proses ini membandingkan elemen array `colors` dengan `correctOrder`. Jika ada warna yang tidak sesuai, variabel `success` diatur ke `false` dan pengecekan dihentikan. Keluar lebih awal jika gagal: Program akan berhenti sebelum lima percobaan jika ada kesalahan. Jika semua percobaan benar, variabel `success` tetap `true`. Output akhir program: Jika semua percobaan sesuai urutan benar, akan dicetak "BERHASIL: true". Jika tidak, "BERHASIL: false". Gagal.

GUIDED

Soal Studi Case

2.

Sourcecode



```
modul 2 > go hello 2.go > main
1 package main
2
3 import (
4     "fmt"
5 )
6
7 func main() {
8
9     var a, b, c, d, e int
10    var hasil int
11    fmt.Scanln(&a, &b, &c, &d, &e)
12
13    hasil = a + b + c + d + e
14    fmt.Println("hasil penjumlahan", a, b, c, d, e, "adalah =", hasil)
15 }
```

Screenshoot Output



```
PROBLEMS 22 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\hello 2.go"
2 3 4 5 6
hasil penjumlahan 2 3 4 5 6 adalah = 20
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

Deklarasi variabel: Program mendeklarasikan lima variabel bilangan bulat (int) yaitu a, b, c, d, dan e, serta variabel hasil untuk menyimpan penjumlahan kelimanya. Program meminta pengguna memasukkan lima bilangan bulat menggunakan fungsi `fmt.Scanln` membaca lima input sekaligus dan menyimpannya di variabel a, b, c, d, dan e. Program kemudian menjumlahkan nilai-nilai tersebut dan menyimpan hasilnya dalam variabel hasil. Output program menampilkan hasil penjumlahan beserta angka-angka yang dijumlahkan.

GUIDED

Soal Studi Case

3. Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

NAM	NMK
NAM > 80	A
72.5 < NAM <= 80	AB
65 < NAM <= 72.5	B

$57.5 < \text{NAM} \leq 65$	BC
$50 < \text{NAM} \leq 57.5$	C
$40 < \text{NAM} \leq 50$	D
$\text{NAM} \leq 40$	E

Sourcecode

```

-Go hello 4.go > ...
1  package main
2
3  import "fmt"
4
5  func main () {
6      var nam float32
7      var nmk string
8
9      // meminta input nilai
10     fmt.Print("masukkan niali: ")
11     fmt.Scan(&nam)
12
13     // logika penentuan nilai huruf berdasarkan nilai numerik
14     if nam > 80 {
15         nmk = "A"
16     } else if nam > 75.5{
17         nmk = "B"
18     } else if nam > 65 {
19         nmk = "C"
20     } else if nam > 50 {
21         nmk = "D"
22     } else if nam > 40 {
23         nmk = "E"
24     } else{
25         nmk = "F"
26     }
27
28     // menampilkan hasil
29     fmt.Printf("nilai indeks untuk nilai %.2f adalah %s\n", nam, nmk)
30 }

```

Screenshoot Output

```

PROBLEMS 22 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\hello 4.go"
masukkan niali: 80.1
nilai indeks untuk nilai 80.10 adalah A
PS D:\1-Priesty AM\alpro 3 semester 3>

```

Deskripsi Program

Program meminta pengguna memasukkan nilai numerik (float32) dan menyimpannya dalam variabel 'nam'. Menentukan nilai huruf: Program menggunakan pernyataan if-else if untuk menentukan nilai huruf. Jika nilai > 80, maka nilai hurufnya adalah "A". Jika nilai > 75.5, maka hurufnya "B". Jika nilai > 65, huruf "C"; jika nilai > 50, huruf "D". Nilai huruf: "E" untuk nilai > 40, "F" untuk nilai ≤ 40. Menampilkan hasil: Setelah nilai huruf (nmk) ditentukan, program menampilkan hasilnya dengan fmt.Printf, mencetak input dan nilai huruf dalam format dua desimal. Output:

- Nilai 80 = A

- Nilai 73 = C

3.) **UNGUIDED**

Modul 2B

Soal Studi Case

1. 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. E 14
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 warna sesuai dengan informasi yang diberikan pada paragraf sebelumnya, dan false untuk urutan warna lainnya

Sourcecode


```

modul2 > -o hello.go > ...
1 package main
2
3 import (
4     "bufio"
5     "fmt"
6     "os"
7     "strings"
8 )
9
10 func main() {
11     // Urutan warna yang benar
12     correctOrder := []string{"merah", "kuning", "hijau", "ungu"}
13
14     // Membaca input untuk 5 percobaan
15     reader := bufio.NewReader(os.Stdin)
16     success := true
17
18     for i := 1; i <= 5; i++ {
19         fmt.Printf("Percobaan %d: ", i)
20
21         // Membaca input dari pengguna
22         input, _ := reader.ReadString('\n')
23         input = strings.TrimSpace(input)
24
25         // Memisahkan input berdasarkan spasi
26         colors := strings.Split(input, " ")
27
28         // Mengecek apakah urutan warna sesuai
29         for j := 0; j < 4; j++ {
30             if colors[j] != correctOrder[j] {
31                 success = false
32                 break
33             }
34         }
35
36         // Jika ada percobaan yang tidak sesuai, keluar dari loop
37         if !success {
38             break
39         }
40     }
41
42     // Menampilkan hasil
43     if success {
44         fmt.Println("BERHASIL: true")
45     } else {
46         fmt.Println("BERHASIL: false")
47     }
48 }

```

Screenshoot Output

```

PS D:> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\hello.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: merah kuning hijau ungu
Percobaan 5: merah kuning hijau ungu
BERHASIL: true
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\hello.go"
Percobaan 1: merah kuning hijau ungu
Percobaan 2: merah kuning hijau ungu
Percobaan 3: merah kuning hijau ungu
Percobaan 4: ungu kuning hijau merah
BERHASIL: false
PS D:\1-Priesty AM\alpro 3 semester 3>

```

Deskripsi Program

Inisialisasi urutan warna: Program menyimpan urutan warna dalam array `correctOrder`: "merah", "kuning", "hijau", "ungu". Pengguna diminta

memasukkan urutan warna lima kali. Setiap input diambil dengan `bufio.NewReader` dan disimpan dalam variabel `input`. Memproses input: Input pengguna akan dihilangkan karakter newline dan dipisah menjadi array string berdasarkan spasi. `Split`. Pengecekan urutan warna: Program memeriksa apakah urutan warna yang dimasukkan sesuai dengan yang benar setelah pemecahan input. Proses ini membandingkan elemen array `colors` dengan `correctOrder`. Jika ada warna yang tidak sesuai, variabel `success` diatur ke `false` dan pengecekan dihentikan. Keluar lebih awal jika gagal: Program akan berhenti sebelum lima percobaan jika ada kesalahan. Jika semua percobaan benar, variabel `success` tetap `true`. Output akhir program: Jika semua percobaan sesuai urutan benar, akan dicetak "BERHASIL: true". Jika tidak, "BERHASIL: false". Gagal.

UNGUIDED

Soal Studi Case

2. 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

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

```
modul2B 1.go 1 | unguidedmodul2B 2.go 1 X | unguidedmodul2B 3.go 1
modul 2 > unguidedmodul2B 2.go > ...
1 package main
2
3 import (
4     "fmt"
5     "strings"
6 )
7
8 func main() {
9     var bunga string
10    var pita string
11    var count int
12
13    for {
14        fmt.Print("Bunga: ")
15        fmt.Scanln(&bunga)
16
17        if bunga == "SELESAI" {
18            break
19        }
20
21        if count > 0 {
22            pita += " - "
23        }
24        pita += bunga
25        count++
26    }
27
28    fmt.Printf("Bunga: %d\n", count)
29    fmt.Printf("Pita: %s\n", strings.TrimSpace(pita))
30 }
31
```

Screenshoot Output

```

modul2 > unguidedmodul2B 2.go > main
4  "fmt"
5  "strings"
6  )
7
8  func main() {
9      var N int
10     fmt.Print("Masukkan jumlah bunga: ")
11     fmt.Scan(&N)
12
13     var pita string
14     for i := 1; i <= N; i++ {
15         var bunga string
16         fmt.Printf("Bunga %d: ", i)
17         fmt.Scan(&bunga)
18
19         if bunga == "SELESAI" {
20             break
21         }
22
23         if pita == "" {
24             pita = bunga
25         } else {
26             pita += " " + bunga
27         }
28     }
29
30     bungalist := strings.Split(pita, " ")
31     fmt.Printf("Pita: %s\n", pita)
32     fmt.Printf("Jumlah bunga: %d\n", len(bungalist))
33 }

```

Deskripsi Program

Inisialisasi bunga: Program meminta pengguna memasukkan jumlah bunga yang ingin ditambahkan, disimpan dalam variabel `N`. **Input nama bunga:** Program meminta pengguna memasukkan nama bunga satu per satu hingga mencapai jumlah yang diminta (`N`) atau hingga pengguna mengetik "SELESAI" untuk menghentikan program. Jika `pita` kosong, nama bunga pertama langsung diisi ke dalam `pita`. Program akan menambahkan nama bunga ke variabel `pita` dengan spasi sebagai pemisah untuk membentuk string yang berisi seluruh nama bunga. Setelah semua bunga dimasukkan atau pengguna menghentikan dengan "SELESAI," string `pita` akan dipecah menjadi array menggunakan `strings`.

Mempersingkat teks: Pisahkan bunga berdasarkan spasi menjadi array `bungalist`. **Output:** Program menampilkan dua hal: `Pita: Seluruh bunga disusun dalam satu string, dipisahkan oleh spasi.` `Jumlah bunga:` Menampilkan total bunga yang telah dimasukkan dengan menghitung panjang array `bungalist`. Tentu! Silakan berikan teks yang ingin Anda ringkas, dan saya akan membantu mempersingkatnya.

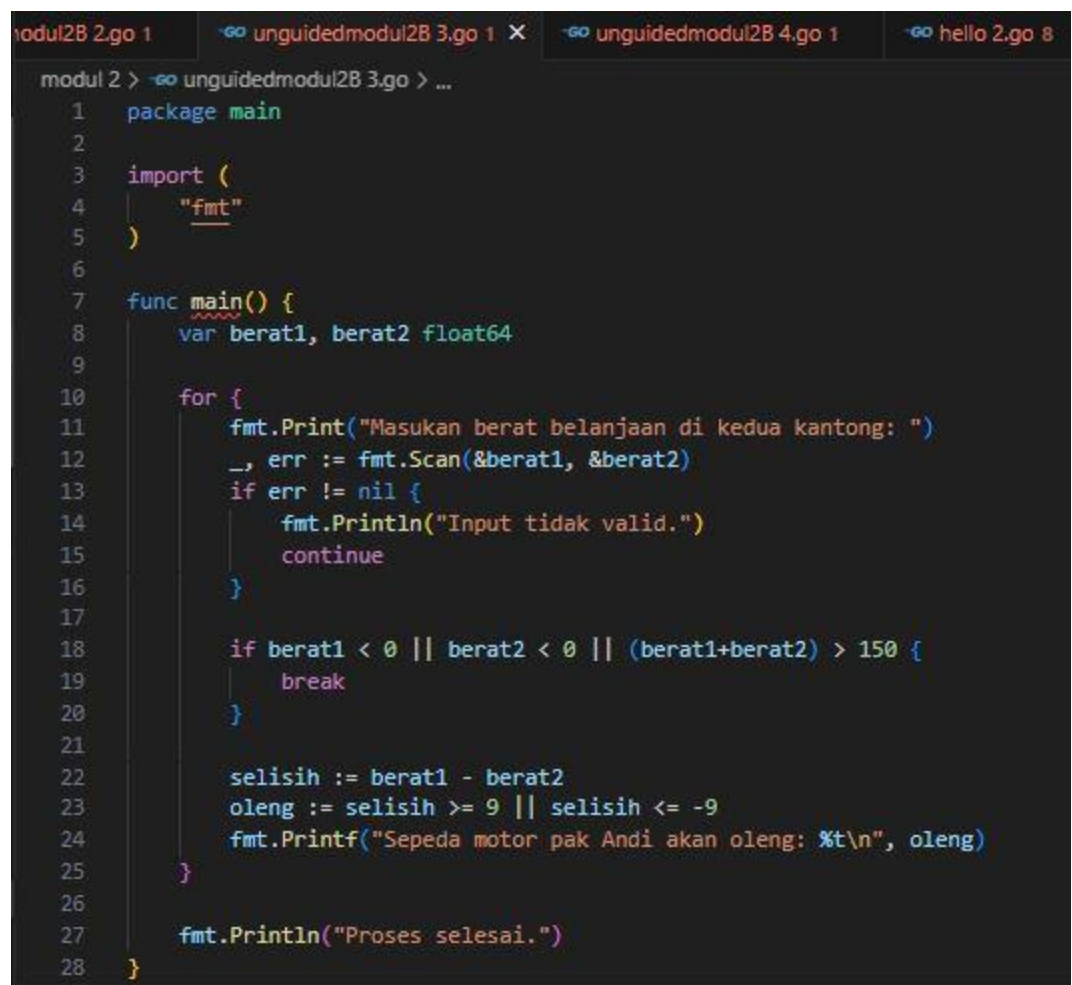
UNGUIDED

Soal Studi Case

3. 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.

Sourcecode



```
modul2B 2.go 1  -go unguidedmodul2B 3.go 1 X  -go unguidedmodul2B 4.go 1  -go hello 2.go 8
modul 2 > -go unguidedmodul2B 3.go > ...
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      var berat1, berat2 float64
9
10     for {
11         fmt.Print("Masukan berat belanjaan di kedua kantong: ")
12         _, err := fmt.Scan(&berat1, &berat2)
13         if err != nil {
14             fmt.Println("Input tidak valid.")
15             continue
16         }
17
18         if berat1 < 0 || berat2 < 0 || (berat1+berat2) > 150 {
19             break
20         }
21
22         selisih := berat1 - berat2
23         oleng := selisih >= 9 || selisih <= -9
24         fmt.Printf("Sepeda motor pak Andi akan oleng: %t\n", oleng)
25     }
26
27     fmt.Println("Proses selesai.")
28 }
```

Screenshoot Output

```
PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguidedmodul2B 3.go"
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.
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

Inisialisasi variabel berat: Program mendeklarasikan dua variabel float64, berat1 dan berat2, untuk menyimpan berat belanjaan dari dua kantong yang diinput pengguna. Program menggunakan perulangan for tanpa kondisi untuk terus meminta input berat belanjaan dari pengguna hingga kondisi tertentu terpenuhi. Input dibaca dengan fungsi fmt.Scan. Jika pengguna memasukkan input tidak valid, program akan menampilkan pesan "Input tidak valid" dan meminta input ulang. Pengecekan berat: Program akan memeriksa apakah berat1 atau berat2 negatif, atau jika total berat melebihi 150 kg. Jika salah satu kondisi terpenuhi, program akan berhenti. Hanya berat positif dan total maksimum 150 kg yang diterima. Menghitung selisih berat: Program menghitung selisih berat kedua kantong (selisih := berat1 - berat2). Jika selisih absolut >= 9...

UNGUIDED

Soal Studi Case

4. 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.

Sourcecode

```
modul 2 > -go unguidedmodul2B 4.go > sqrt2Approximation
1 package main
2
3 import (
4     "fmt"
5     "math"
6 )
7
8 func f(k int) float64 {
9     return math.Pow(float64(4*k+2), 2) / float64((4*k+1)*(4*k+3))
10 }
11
12 func sqrt2Approximation(k int) float64 {
13     approx := 0.0
14     for i := 0; i <= k; i++ {
15         approx += f(i)
16     }
17     return approx
18 }
19
20 func main() {
21     var k int
22     for {
23         fmt.Print("Nilai K = ")
24         _, err := fmt.Scan(&k)
25         if err != nil {
26             break
27         }
28         approx := sqrt2Approximation(k)
29         fmt.Printf("Nilai akar 2 %.10f\n", approx)
30     }
31 }
```

Screenshoot Output

```
PROBLEMS 24 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguidedmodul2B 4.go"
Priesty AM\alpro 3 semester 3\modul 2\unguidedmodul2B 4.go
Nilai K: 10
Nilai akar: 3.9503470889
Priesty AM\alpro 3 semester 3\modul 2\unguidedmodul2B 4.go
Nilai K: 10
Priesty AM\alpro 3 semester 3\modul 2\unguidedmodul2B 4.go
Nilai K: 10
Nilai akar: 3.9503470889
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguidedmodul2B 4.go"
Nilai K = 10
Nilai akar 2 11.3870201908
Nilai K = 100
Nilai akar 2 101.3920802736
Nilai K = 100
Nilai akar 2 101.3920802736
Nilai K =
```


Deskripsi Program

Fungsi `f(k int) float64` menghitung nilai elemen deret berdasarkan parameter `k` dengan rumus: $f(k) = (4k + 2)^2 / ((4k + 1)(4k + 3))$. Hasilnya dikembalikan sebagai `float64` untuk mempertahankan presisi. Fungsi `sqrt2Approximation(k int) float64` menghitung perkiraan $\sqrt{2}$ dengan menjumlahkan elemen deret dari fungsi `f(k)` untuk `i` dari 0 hingga `k`. Semakin besar `k`, semakin akurat perkiraannya. Inisialisasi `approx` dengan nilai 0 sebagai tempat penjumlahan. Dilakukan iterasi `for` dari `i = 0` hingga `k` untuk menjumlahkan elemen deret dengan fungsi `f(i)`, dan hasilnya disimpan dalam variabel `approx` sebagai perkiraan nilai $\sqrt{2}$. Fungsi utama meminta pengguna memasukkan nilai `k` untuk menentukan jumlah elemen deret dalam estimasi. Setiap nilai `k` yang dimasukkan memicu pemanggilan fungsi `sqrt2Approximation(k)` untuk menghitung perkiraan $\sqrt{2}$ berdasarkan `k` elemen deret. Setelah perhitungan selesai, program mencetak hasil dengan 10 angka desimal. Program akan terus meminta input hingga pengguna memberikan input tidak valid (seperti tombol non-integer atau EOF). Tentu! Silakan berikan teks yang ingin Anda ringkas, dan saya akan membantu mempersingkatnya.

UNGUIDED

Modul 2C

Soal Studi Case

1. PT POS membutuhkan aplikasi perhitungan biaya kirim berdasarkan berat parcel. Maka, buatlah program `BlayaPos` untuk menghitung biaya pengiriman tersebut dengan ketentuan sebagai berikut! Dari berat parcel (dalam gram), harus dihitung total berat dalam kg dan sisanya (dalam gram). Biaya jasa pengiriman adalah Rp. 10.000,- per kg. Jika sisa berat tidak kurang dari 500 gram, maka tambahan biaya kirim hanya Rp. 5,- per gram saja. Tetapi jika kurang dari 500 gram, maka tambahan biaya akan dibebankan sebesar Rp. 15,- per gram. Sisa berat (yang kurang dari 1kg) digratiskan biayanya apabila total berat ternyata lebih dari 10kg.

Sourcecode


```
unguidedmodul2B 3.go 1  -go unguided2C 1.go 1 X  -go unguidedmodul2B 4.go 1  -go hello 2.g

modul 2 > -go unguided2C 1.go > ...
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      var beratParcel int
9      fmt.Print("Masukkan berat parsel dalam gram: ")
10     fmt.Scan(&beratParcel)
11
12     totalBeratKg := beratParcel / 1000
13     sisaBeratGram := beratParcel % 1000
14
15     biayaPengiriman := totalBeratKg * 10000
16
17     if totalBeratKg > 10 {
18         sisaBeratGram = 0
19     } else if sisaBeratGram >= 500 {
20         biayaPengiriman += sisaBeratGram * 5
21     } else {
22         biayaPengiriman += sisaBeratGram * 15
23     }
24
25     fmt.Printf("Total biaya pengiriman: Rp. %d\n", biayaPengiriman)
26 }
```

Screenshoot Output

```
PROBLEMS 26 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\1-Priesty AM\alpro 3 semester
3> go run "d:\1-Priesty AM\alpro 3
Masukkan berat parsel dalam gram: 8
500
Total biaya pengiriman: Rp. 82500
PS D:\1-Priesty AM\alpro 3 semester
3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguided2C 1.go"
Masukkan berat parsel dalam gram: 9250
Total biaya pengiriman: Rp. 93750
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguided2C 1.go"
Masukkan berat parsel dalam gram: 11750
Total biaya pengiriman: Rp. 110000
PS D:\1-Priesty AM\alpro 3 semester 3> |
```

Deskripsi Program

Input berat parsel: Pengguna diminta memasukkan berat parsel dalam gram. Konversi: Berat dipecah menjadi kilogram (totalBeratKg) dan sisa gram (sisaBeratGram). totalBeratKg diperoleh dengan membagi berat parsel dalam gram dengan 1000, sedangkan sisaBeratGram adalah modulus dari berat parsel dengan 1000, menunjukkan sisa berat yang

kurang dari satu kilogram. Menghitung biaya pengiriman: Biaya per kilogram dihitung dengan mengalikan berat total (kg) dengan Rp. 10. Setiap kilogram dikenakan biaya Rp. Teks Anda terlalu singkat untuk diringkas. Mohon berikan teks yang lebih panjang atau penjelasan lebih lanjut tentang apa yang ingin Anda ringkas. Logika tambahan sisa berat: Jika berat total parsel > 10 kg, sisa berat gram diset menjadi 0, dan biaya pengiriman dihitung hanya dari kilogram penuh. Jika sisa berat gram \geq 500 gram, biaya ekstra ditambahkan dengan mengalikan sisa berat gram dengan Rp. Biaya tambahan dihitung dengan mengalikan sisa berat yang kurang dari 500 gram dengan Rp. 5 per gram. 15 rupiah per gram. Tentu, silakan berikan teks yang ingin Anda pendekkan.

UNGUIDED

Soal Studi Case

2. Diberikan sebuah nilai akhir mata kuliah (NAM) [0..100] dan standar penilaian nilai mata kuliah (NMK) sebagai berikut:

NAM	NMK
$NAM > 80$	A
$72.5 < NAM \leq 80$	AB
$65 < NAM \leq 72.5$	B
$57.5 < NAM \leq 65$	BC
$50 < NAM \leq 57.5$	C
$40 < NAM \leq 50$	D
$NAM \leq 40$	E

Jawablah pertanyaan-pertanyaan berikut:

- Jika nam diberikan adalah 80.1, apa keluaran dari program tersebut? Apakah eksekusi program tersebut sesuai spesifikasi soal?
=> nilai indeks untuk nilai 80.10 adalah A
- Apa saja kesalahan dari program tersebut? Mengapa demikian? Jelaskan alur program seharusnya!
=> A. Kriteria Penentuan Nilai Huruf:

- Dalam program tersebut, huruf "C" ditulis dengan huruf kecil ("c"), sementara huruf "E" dan "F" juga ditulis dalam huruf kecil ("E" dan "F"). Ini tidak konsisten.

B. Logika Penentuan Nilai:

- Tidak ada penanganan untuk input di bawah 0 atau di atas 100. Program seharusnya dapat menangani nilai-nilai di luar rentang ini.

C. Alur Program Seharusnya:

1. Meminta pengguna untuk memasukkan nilai.
 2. Memeriksa nilai yang dimasukkan.
 3. Menentukan nilai huruf berdasarkan kriteria yang sudah ditentukan.
 4. Menampilkan hasilnya kepada pengguna.
- c. Perbaiki program tersebut! Ujilah dengan masukan: 93.5; 70.6; dan 49.5. Seharusnya keluaran yang diperoleh adalah 'A', 'B', dan 'D'.

=>

```
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-
masukkan niali: 93.5
nilai indeks untuk nilai 93.50 adalah A
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguided2C 2.go"
masukkan niali: 70.6
nilai indeks untuk nilai 70.60 adalah c
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguided2C 2.go"
masukkan niali: 49.5
nilai indeks untuk nilai 49.50 adalah E
PS D:\1-Priesty AM\alpro 3 semester 3> 
```

Sourcecode

```

1 package main
2
3 import "fmt"
4
5 func main () {
6     var nam float32
7     var nmk string
8
9     // meminta input nilai
10    fmt.Print("masukkan niali: ")
11    fmt.Scan(&nam)
12
13    // logika penentuan nilai huruf berdasarkan nilai numerik
14    if nam > 80 {
15        nmk = "A"
16    } else if nam > 75.5 {
17        nmk = "B"
18    } else if nam > 65 {
19        nmk = "C"
20    } else if nam > 50 {
21        nmk = "D"
22    } else if nam > 40 {
23        nmk = "E"
24    } else {
25        nmk = "F"
26    }
27
28    // menampilkan hasil
29    fmt.Printf("nilai indeks untuk nilai %.2f adalah %s\n", nam, nmk)
30 }

```

Screenshoot Output

```

PROBLEMS 22 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\hello 4.go"
masukkan niali: 80.1
nilai indeks untuk nilai 80.10 adalah A
PS D:\1-Priesty AM\alpro 3 semester 3>

```

Deskripsi Program

Program meminta pengguna memasukkan nilai numerik (float32) dan menyimpannya dalam variabel 'nam'. Menentukan nilai huruf: Program menggunakan pernyataan if-else if untuk menentukan nilai huruf. Jika nilai > 80, maka nilai hurufnya adalah "A". Jika nilai > 75.5, maka hurufnya "B". Jika nilai > 65, huruf "C"; jika nilai > 50, huruf "D". Nilai huruf: "E" untuk nilai > 40, "F" untuk nilai ≤ 40. Menampilkan hasil: Setelah nilai huruf (nmk) ditentukan, program menampilkan hasilnya

dengan `fmt.Printf`, mencetak input dan nilai huruf dalam format dua desimal. Output:

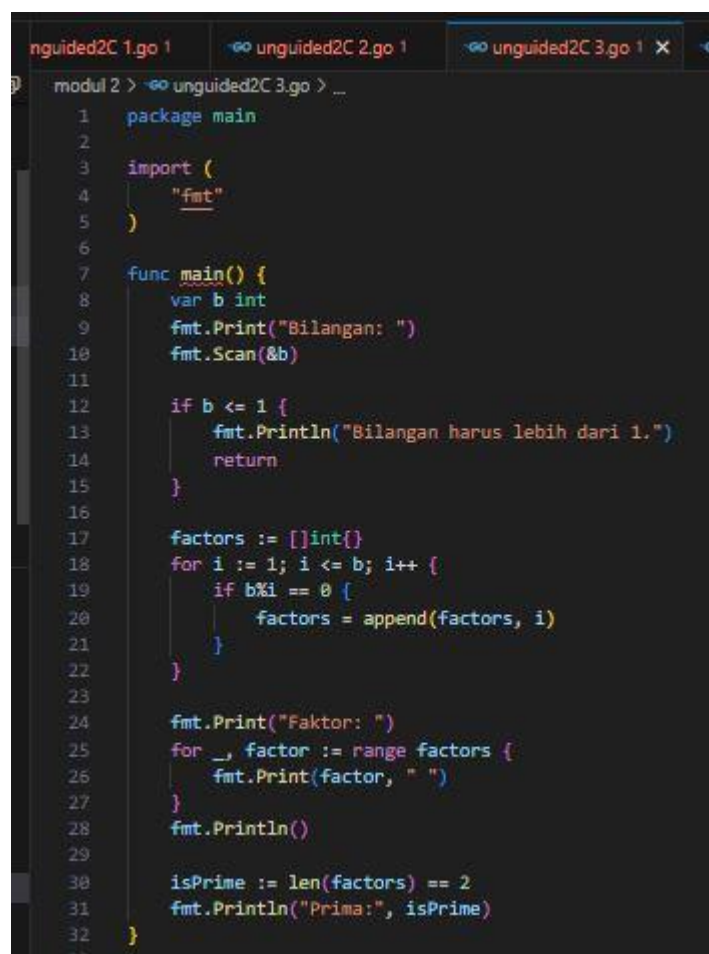
- Nilai 80 = A
- Nilai 73 = C

UNGUIDED

Soal Studi Case

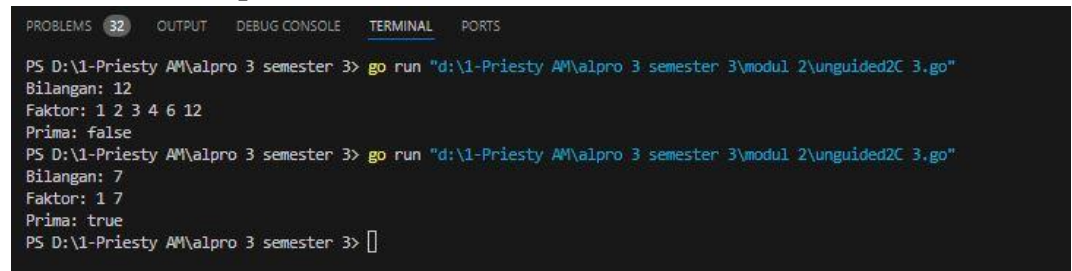
3. Sebuah bilangan bulat b memiliki faktor bilangan $f > 0$ jika f habis membagi b . Contoh: 2 merupakan faktor dari bilangan 6 karena 6 habis dibagi 2. Buatlah program yang menerima input sebuah bilangan bulat b dan $b > 1$. Program harus dapat mencari dan menampilkan semua faktor dari bilangan tersebut!

Sourcecode



```
modul 2 > unguided2C 3.go > ...
1  package main
2
3  import (
4      "fmt"
5  )
6
7  func main() {
8      var b int
9      fmt.Print("Bilangan: ")
10     fmt.Scan(&b)
11
12     if b <= 1 {
13         fmt.Println("Bilangan harus lebih dari 1.")
14         return
15     }
16
17     factors := []int{}
18     for i := 1; i <= b; i++ {
19         if b%i == 0 {
20             factors = append(factors, i)
21         }
22     }
23
24     fmt.Print("Faktor: ")
25     for _, factor := range factors {
26         fmt.Print(factor, " ")
27     }
28     fmt.Println()
29
30     isPrime := len(factors) == 2
31     fmt.Println("Prima:", isPrime)
32 }
```

Screenshot Output



```
PROBLEMS 32 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguided2C 3.go"
Bilangan: 12
Faktor: 1 2 3 4 6 12
Prima: false
PS D:\1-Priesty AM\alpro 3 semester 3> go run "d:\1-Priesty AM\alpro 3 semester 3\modul 2\unguided2C 3.go"
Bilangan: 7
Faktor: 1 7
Prima: true
PS D:\1-Priesty AM\alpro 3 semester 3> 
```

Deskripsi Program

Program meminta pengguna untuk memasukkan bilangan bulat dengan `fmt.Scan()`. Bilangan disimpan dalam variabel `b` dan divalidasi untuk memastikan nilainya lebih dari 1. Jika bilangan tidak memenuhi syarat, program akan menampilkan pesan kesalahan dan berhenti. Selanjutnya, program menginisialisasi slice factors untuk menyimpan semua faktor dari bilangan `b`. Program menggunakan perulangan `for` untuk memeriksa setiap bilangan `i` dari 1 hingga `b`, dan menambahkan `i` ke dalam slice factors jika `b` dapat dibagi habis oleh `i` (sisa pembagian `b` oleh `i` adalah 0).
Menampilkan faktor: Program mencetak semua faktor dari slice factors.
Memeriksa bilangan prima: Program memeriksa apakah jumlah faktor sama dengan 2. Jika ya, maka `b` bilangan prima.