**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II**

**Review Struktur Kontrol**

**Sebuah gambar berisi logo, simbol, Grafis, Font

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Fariz Ilham / 2311102275**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. **DASAR TEORI**

Dalam pemrograman Go, setiap program utama terdiri dari dua komponen penting:

1. **Package Main**: Menandakan bahwa file ini adalah program utama.
2. **Fungsi Main**: Berisi kode eksekusi utama dari program Go.

Selain itu, komentar dapat ditambahkan di mana saja dalam kode. Komentar ini berguna untuk penjelasan dan tidak mempengaruhi eksekusi program. Terdapat dua jenis komentar:

* Komentar satu baris yang diawali dengan //.
* Komentar multi-baris yang dibuka dengan /\* dan ditutup dengan \*/.

### Koding, Kompilasi, dan Eksekusi

#### Koding

#### Dalam Go, program ditulis menggunakan penyunting teks dan disimpan dalam format file dengan ekstensi .go. Setiap program disimpan dalam folder terpisah, dan nama folder tersebut sebaiknya mencerminkan nama program. Meskipun program dapat dibagi dalam beberapa file dengan ekstensi .go, semuanya harus berada dalam folder yang sama.

#### Kompilasi

Go adalah bahasa pemrograman yang menggunakan proses kompilasi. Berbeda dengan bahasa pemrograman yang diinterpretasikan, kompilator Go memeriksa keseluruhan kode sumber dan mengubahnya menjadi program yang dapat dieksekusi. Proses ini memastikan bahwa kesalahan sintaksis terdeteksi sebelum program dieksekusi, dan optimasi dapat dilakukan untuk meningkatkan efisiensi program.

1. **GUIDED**

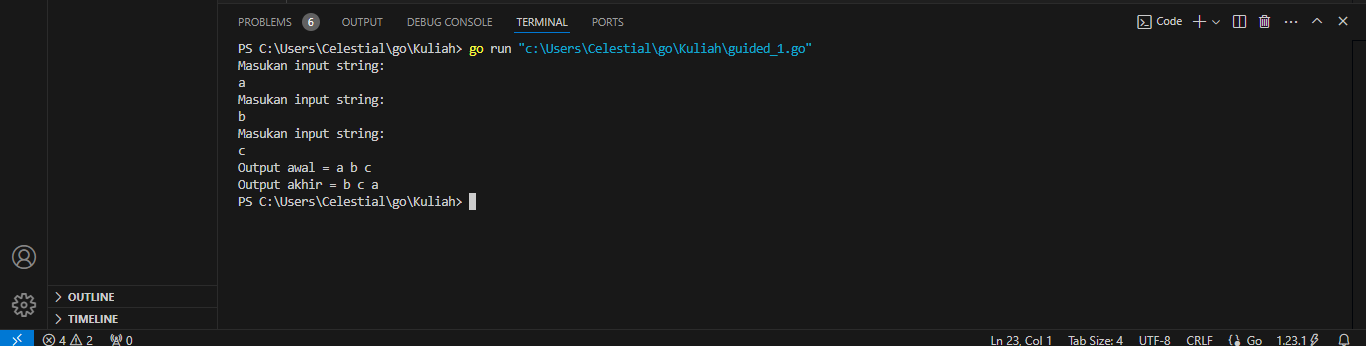
**Soal Studi Case**

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

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var (          satu, dua, tiga string          temp string      )      fmt.Println("Masukan input string: ")      fmt.Scanln(&satu)      fmt.Println("Masukan input string: ")      fmt.Scanln(&dua)      fmt.Println("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**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menerima tiga input string, kemudian menukar urutan string sebelum menampilkan hasilnya.

* **Algoritma**

1. Inisialisasi Variabel:

* Tiga variabel string untuk menyimpan input dan satu variabel tambahan untuk membantu dalam proses pertukaran.

2. Pengambilan Input:

* Program meminta pengguna untuk memasukkan tiga string secara berurutan dan input akan disimpan ke dalam variabel yang telah disiapkan.

3. Menampilkan Output Awal:

* Program menampilkan output awal yang merupakan ketiga string yang telah dimasukkan.

4. Pertukaran Nilai:

* Menggunakan variabel tambahan, program menukar urutan nilai dari ketiga string tersebut.

5. Menampilkan Output Akhir:

* Program menampilkan output akhir setelah nilai string ditukar.
* **Cara Kerja Program**

1. Input Pengguna:

* Pengguna diminta untuk memasukkan tiga string secara berurutan.

2. Menampilkan Output Awal:

* Program menampilkan ketiga string yang telah diinputkan.

3. Proses Pertukaran:

* Program menukar urutan nilai string, sehingga urutan nilai string berubah.

4. Menampilkan Output Akhir:

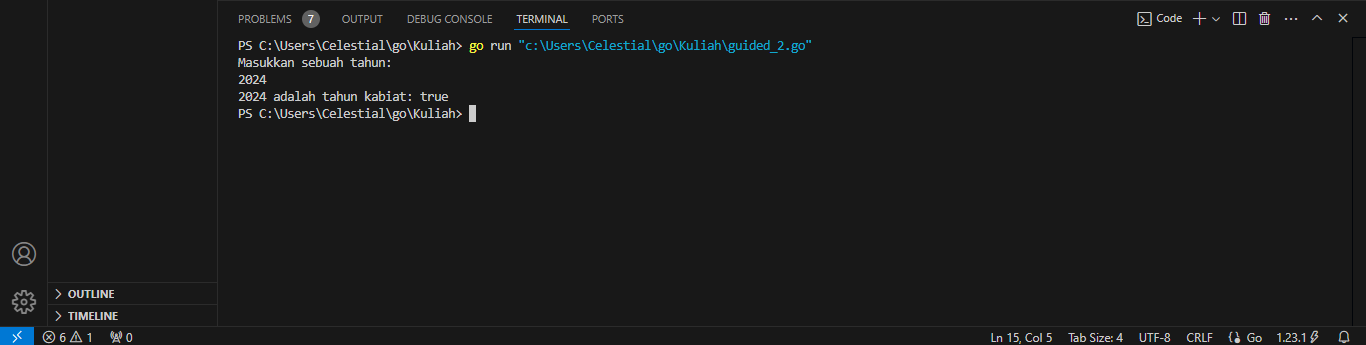
* Program menampilkan hasil setelah urutan string ditukar.

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

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var tahun int      fmt.Println("Masukkan sebuah tahun: ")      fmt.Scanln(&tahun)      if (tahun%400 == 0) || (tahun%4 == 0 && tahun%100 != 0) {          fmt.Println(tahun, "adalah tahun kabiat: true")      } else {          fmt.Println(tahun, "bukan tahun kabisat: false")      }  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menentukan apakah sebuah tahun merupakan tahun kabisat atau bukan. Tahun kabisat adalah tahun yang dapat habis bila dibagi dengan 4 atau habis di bagi 400 tetapi tidak habis dibagi dengan 100.

* **Algoritma**

1. Inisialisasi Variabel:

* Variabel integer untuk menyimpan input dari pengguna.

2. Pengambilan Input:

* Program meminta pengguna untuk memasukkan sebuah tahun.

3. Pemeriksaan Tahun Kabisat:

* Program memeriksa apakah tahun tersebut memenuhi kondisi sebagai tahun kabisat:
* Tahun kabisat jika tahun tersebut dapat dibagi 400 menghasilkan nilai 0, atau
* Tahun kabisat jika tahun tersebut dapat dibagi 4 menghasilkan nilai 0 dan dapat dibagi 100 tetapi tidak menghasilkan nilai 0.

4. Menampilkan Hasil:

* Program menampilkan hasil dari pemeriksaan, apakah tahun tersebut kabisat atau bukan.
* **Cara Kerja Program**

1. Input Pengguna:

* Pengguna diminta untuk memasukkan sebuah tahun.

2. Pemeriksaan Kriteria Tahun Kabisat:

* Program melakukan pemeriksaan terhadap tahun yang dimasukkan berdasarkan kriteria tahun kabisat.

3. Menampilkan Hasil:

* Jika tahun tersebut memenuhi kriteria, program akan menampilkan bahwa tahun tersebut adalah tahun kabisat dengan kalimat “true”.
* Jika tidak memenuhi kriteria, program akan menampilkan bahwa tahun tersebut bukan tahun kabisat dengan kalimat “false”.

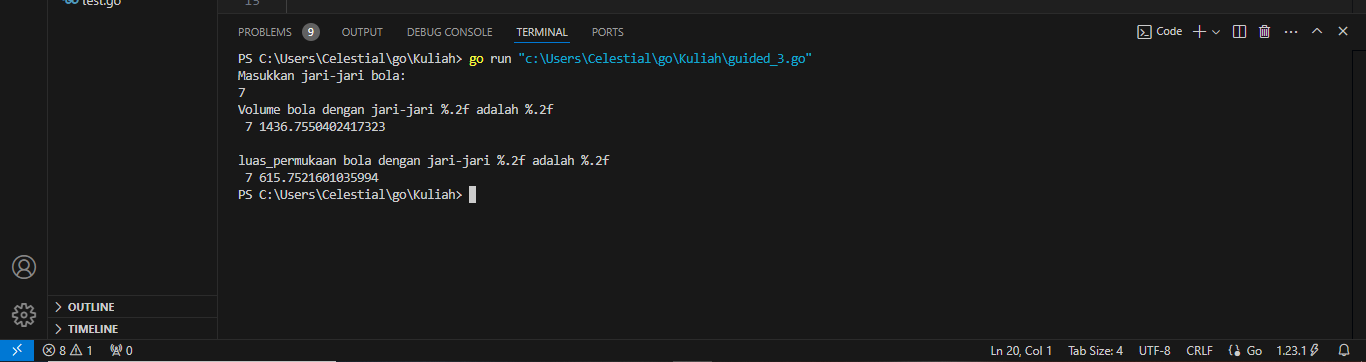
1. Buat program Bola yang menerima input jari-jari suatu bola

(bilangan bulat). Tampilkan Volume dan Luas kulit bola. volumebola = r³ dan luasbola = 4πr² (π = 3.1415926535).

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func main() {      var jarijari float64      fmt.Println("Masukkan jari-jari bola: ")      fmt.Scanln(&jarijari)  volume := (4.0 / 3.0) \* math.Pi \* math.Pow(jarijari, 3)  luas\_permukaan := 4 \* math.Pi \* math.Pow(jarijari, 2)  fmt.Println("Volume bola dengan jari-jari %.2f adalah %.2f\n", jarijari, volume)      fmt.Println()      fmt.Println("luas\_permukaan bola dengan jari-jari %.2f adalah %.2f\n", jarijari, luas\_permukaan)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menghitung volume dan luas permukaan sebuah bola berdasarkan jari-jari yang dimasukkan oleh pengguna. Volume bola dapat dihitung menggunakan rumus “ (4.0 / 3.0) \* math.Pi \* math.Pow(jarijari, 3) ”, dan luas permukaan bola dapat dihitung dengan rumus “ 4 \* math.Pi \* math.Pow(jarijari, 2) “.

* **Algoritma**

1. Inisialisasi Variabel:

* Variabel bertipe float64 untuk menyimpan input jari-jari bola dari pengguna.

2. Pengambilan Input:

* Program meminta pengguna untuk memasukkan nilai jari-jari bola.

3. Perhitungan Volume dan Luas Permukaan:

* Menghitung volume bola menggunakan rumus “ (4.0 / 3.0) \* math.Pi \* math.Pow(jarijari, 3) ”.
* Menghitung luas permukaan bola menggunakan rumus “ 4 \* math.Pi \* math.Pow(jarijari, 2) “.

4. Menampilkan Hasil:

* Program menampilkan hasil dari perhitungan volume dan luas permukaan bola.
* **Cara Kerja Program**

1. Input Pengguna:

* Pengguna diminta untuk memasukkan jari-jari bola.

2. Perhitungan:

* Setelah mendapatkan input, program menghitung volume dan luas permukaan bola berdasarkan jari-jari yang dimasukkan.

3. Menampilkan Hasil:

* Program menampilkan volume dan luas permukaan bola dalam format yang sudah ditentukan.

1. **UNGUIDED**

**Soal Studi Case**

1. Di baca nilai temperatur dalam derajat Celsius. Nyatakan temperatur tersebut dalam Fahrenheit.

Fahrenheit = ( Celsius 9/5) + 32

Reamur = Celsius 4/5

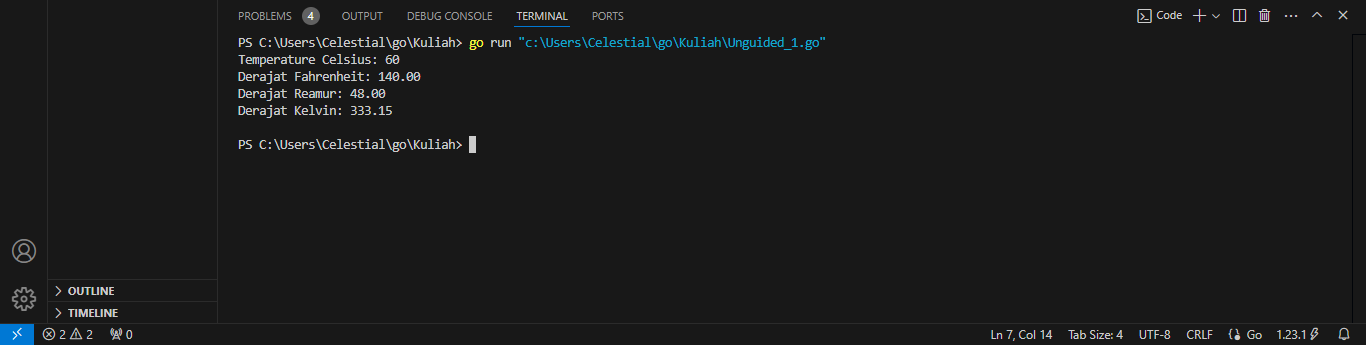
Kelvin = (Fahrenheit +459.67) 5/9

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

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"  )  func main() {  var celsius float64      fmt.Print("Temperatur Celsius: ")      \_, err := fmt.Scan(&celsius)      if err != nil {          fmt.Println("Input tidak valid. Harap masukkan angka.")          return      }      fahrenheit := (celsius \* 9 / 5) + 32      reamur := celsius \* 4 / 5      kelvin := celsius + 273.15      fmt.Printf("Derajat Fahrenheit: %.2f\n", fahrenheit)      fmt.Printf("Derajat Reamur: %.2f\n", reamur)      fmt.Printf("Derajat Kelvin: %.2f\n", kelvin)      fmt.Println()  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk mengonversi suhu dari satuan derajat Celsius ke tiga satuan lainnya: Fahrenheit, Reamur, dan Kelvin. Pengguna akan diminta untuk memasukkan suhu dalam Celsius, dan program akan mengonversi serta menampilkan hasil konversi dari ketiga satuan tersebut.

* **Algoritma**

1. Inisialisasi Variabel:

* Variabel bertipe `float64` untuk menyimpan input suhu dalam derajat Celsius.

2. Pengambilan Input:

* Program meminta pengguna untuk memasukkan suhu dalam Celsius.

3. Perhitungan Konversi:

* Mengonversi suhu dari Celsius ke Fahrenheit dengan rumus “ (celsius \* 9 / 5) + 32 “.
* Mengonversi suhu dari Celsius ke Reamur dengan rumus “ celsius \* 4 / 5 “.
* Mengonversi suhu dari Celsius ke Kelvin dengan rumus “ celsius + 273.15 “.

4. Menampilkan Hasil:

* Program menampilkan hasil konversi suhu dalam tiga satuan: Fahrenheit, Reamur, dan Kelvin.
* **Cara Kerja Program**

1. Input Pengguna:

* Pengguna diminta untuk memasukkan nilai suhu dalam Celsius.

2. Perhitungan:

* Program melakukan perhitungan konversi dari suhu Celsius ke Fahrenheit, Reamur, dan Kelvin berdasarkan rumus yang telah ditentukan.

3. Menampilkan Hasil:

* Program menampilkan suhu yang telah dikonversi ke dalam ketiga satuan tersebut.

1. 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)

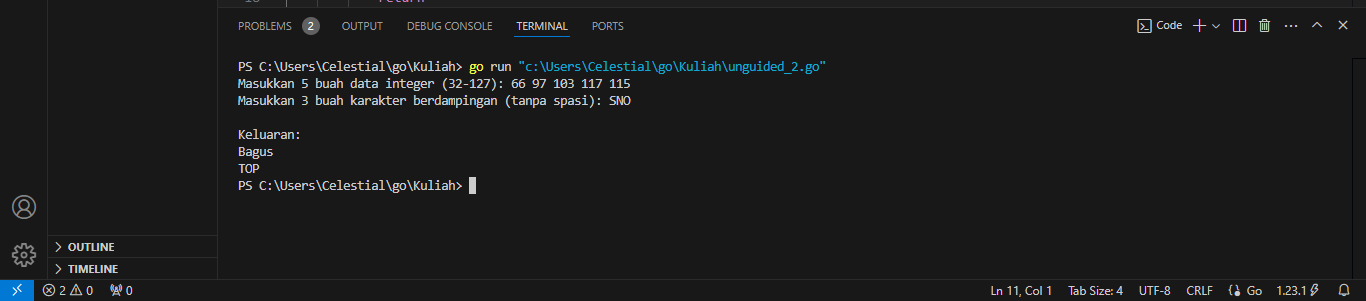
Masukan terdiri dari dua baris. Baris pertarna berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

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 (      "fmt"      "unicode"  )  func main() {      var intArr [5]int  var charArr [3]rune      fmt.Print("Masukkan 5 buah data integer (32-127): ")      for i := 0; i < 5; i++ {          fmt.Scan(&intArr[i])          if intArr[i] < 32 || intArr[i] > 127 {              fmt.Println("Input harus berupa bilangan antara 32 sampai 127.")              return          }  }      fmt.Print("Masukkan 3 buah karakter berdampingan (tanpa spasi): ")      var input string  fmt.Scan(&input)      if len(input) != 3 {          fmt.Println("Input harus terdiri dari 3 karakter.")          return      }      for \_, char := range input {          if unicode.IsDigit(char) {              fmt.Println("Input harus berupa karakter, bukan angka.")              return          }      }      for i := 0; i < 3; i++ {          charArr[i] = rune(input[i])      }      fmt.Println()      fmt.Print("Keluaran:\n")      for i := 0; i < 5; i++ {          fmt.Print(string(intArr[i]))      }  fmt.Println()      for i := 0; i < 3; i++ {          fmt.Print(string(charArr[i] + 1))      }      fmt.Println()  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menerima input berupa 5 bilangan bulat (integer) dan 3 karakter. Program akan meminta pengguna untuk memasukkan 5 angka dalam rentang 32 hingga 127 dan 3 karakter berdampingan tanpa spasi. Setelah menerima input, program akan menampilkan karakter ASCII yang sesuai dengan angka yang dimasukkan dan karakter yang dimasukkan ditambah satu (berdasarkan kode ASCII).

* **Algoritma**

1. Inisialisasi Variabel:

* Satu array integer dengan ukuran 5 untuk menyimpan input angka.
* Satu array rune dengan ukuran 3 untuk menyimpan karakter.

2. Pengambilan Input Angka:

* Program meminta pengguna untuk memasukkan 5 angka.
* Memeriksa apakah setiap angka berada dalam rentang 32 hingga 127. Jika tidak, program akan menampilkan pesan kesalahan dan keluar.

3. Pengambilan Input Karakter:

* Program meminta pengguna untuk memasukkan 3 karakter berdampingan (tanpa spasi) dan memeriksa apakah input terdiri dari 3 karakter. Jika tidak, program akan menampilkan pesan kesalahan dan keluar.

4. Menyimpan Input Karakter:

* Jika semua pemeriksaan berhasil, program akan menyimpan karakter ke dalam array rune.

5. Menampilkan Keluaran:

* Program menampilkan karakter ASCII yang sesuai dengan angka dalam array integer.
* Program menampilkan karakter yang dimasukkan oleh pengguna, tetapi setiap karakter ditambah satu dalam kode ASCII sebelum ditampilkan.
* **Cara Kerja Program**

1. Input Pengguna:

* Pengguna diminta untuk memasukkan 5 angka dalam rentang 32 hingga 127. Setelah itu, pengguna diminta untuk memasukkan 3 karakter tanpa spasi.

2. Validasi Input:

* Program memeriksa validitas input angka dan karakter. Jika terdapat kesalahan, program akan memberi tahu pengguna dan keluar.

3. Konversi dan Menampilkan Hasil:

* Program mengonversi setiap angka menjadi karakter ASCII dan menampilkannya.
* Program juga menampilkan karakter yang dimasukkan oleh pengguna, tetapi setiap karakter ditambah satu berdasarkan kode ASCII sebelum ditampilkan.

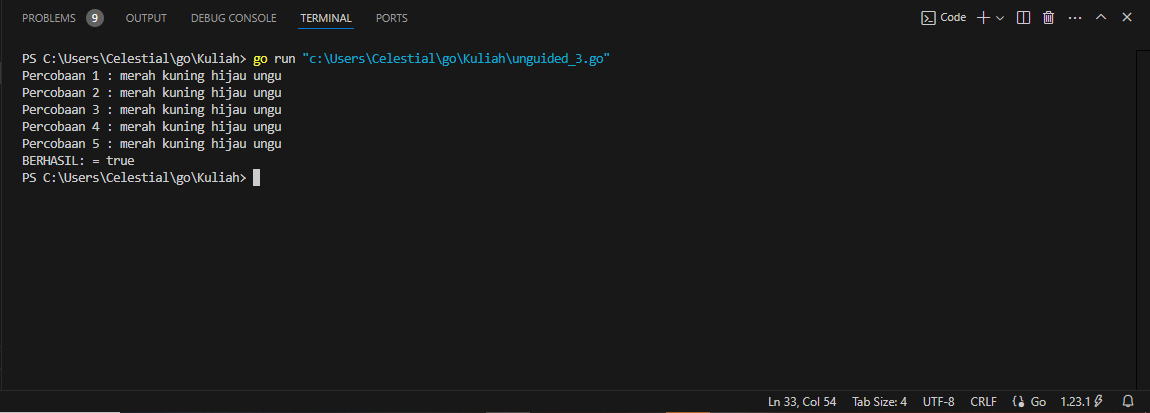
1. Siswa kelas IPA di salah satu sekolah menengah atas di Indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. Percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara berturutan 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 (      "bufio"      "fmt"      "os"      "strings"  )  func main() {      var percobaan [5][4]string  expectedColors := []string{"merah", "kuning", "hijau", "ungu"}      reader := bufio.NewReader(os.Stdin)      for i := 0; i < 5; i++ {          fmt.Printf("Percobaan %d : ", i+1)          input, \_ := reader.ReadString('\n')          input = strings.TrimSpace(input)          colors := strings.Fields(input)          if len(colors) != 4 {              fmt.Println("Input harus terdiri dari 4 warna. Silakan coba lagi.")              i--              continue          }          for j := 0; j < 4; j++ {              percobaan[i][j] = strings.ToLower(colors[j])          }      }      warna := true      for i := 0; i < 5; i++ {          for j := 0; j < 4; j++ {              if percobaan[i][j] != expectedColors[j] {                  warna = false                  break              }          }          if !warna {              break          }      }      fmt.Printf("BERHASIL: = %t\n", warna)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk menerima input warna dari pengguna dalam lima percobaan. Setiap percobaan harus berisi empat warna. Program akan memeriksa apakah setiap input warna sesuai dengan urutan warna yang diharapkan ("merah", "kuning", "hijau", "ungu"). Di akhir, program akan menampilkan apakah semua percobaan berhasil atau tidak.

* **Algoritma**

1. Inisialisasi Variabel:

* Satu array dua dimensi untuk menyimpan input warna dari lima percobaan.
* Satu slice untuk menyimpan warna yang diharapkan.

2. Pengambilan Input:

* Menggunakan `bufio.NewReader` untuk membaca input dari pengguna.
* Dalam loop, program meminta pengguna untuk memasukkan empat warna untuk setiap percobaan.

3. Validasi Input:

* Program memeriksa apakah jumlah warna yang dimasukkan tepat empat. Jika tidak, program akan meminta pengguna untuk mencoba lagi.
* Warna yang dimasukkan akan diubah menjadi huruf kecil untuk memastikan konsistensi.

4. Pemeriksaan Kesamaan Warna:

* Program membandingkan setiap warna yang dimasukkan ke dalam setiap percobaan dengan warna yang diharapkan.
* Jika ada satu warna yang tidak cocok, program akan mengubah status `warna` menjadi `false` dan keluar dari loop.

5. Menampilkan Hasil:

* Program menampilkan apakah semua percobaan berhasil atau tidak berdasarkan kesesuaian warna.
* **Cara Kerja Program**

1. Input Pengguna:

* Program meminta pengguna untuk memasukkan warna untuk lima percobaan, dengan masing-masing percobaan terdiri dari empat warna.

2. Validasi dan Penyimpanan:

* Program memvalidasi input untuk memastikan bahwa setiap percobaan berisi tepat empat warna dan menyimpannya dalam array.

3. Pemeriksaan Kesesuaian:

* Program memeriksa kesesuaian warna dari setiap percobaan dengan warna yang diharapkan.

4. Menampilkan Hasil Akhir:

* Program menampilkan hasil akhir dalam bentuk boolean, menunjukkan apakah semua percobaan berhasil.

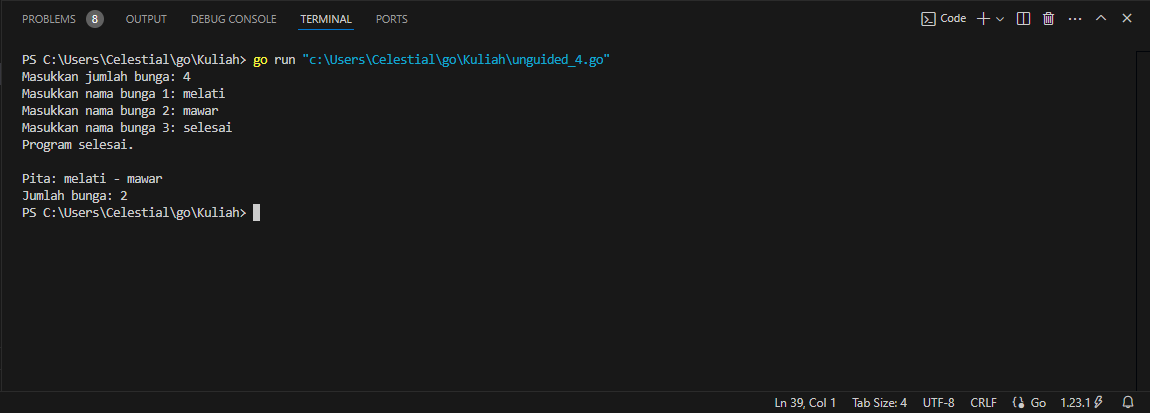
1. Buatlah sebuah program yang menerima input sebuah bilangan bulat positif (dan tidak nol) N, kemudian program, akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita.

Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetikkan 'selesai'. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "strings"  )  func main() {      fmt.Print("Masukkan jumlah bunga: ")      var n int  fmt.Scanln(&n)  var flowers []string      for i := 0; i < n; i++ {          var flower string          fmt.Printf("Masukkan nama bunga %d: ", i+1)          fmt.Scanln(&flower)          if strings.ToUpper(flower) == "SELESAI" {              fmt.Println("Program selesai.")              break          }          flowers = append(flowers, flower)  }  fmt.Println()      pita := strings.Join(flowers, " - ")      fmt.Println("Pita:", pita)      fmt.Printf("Jumlah bunga: %d\n", len(flowers))  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk meminta pengguna memasukkan nama bunga hingga jumlah yang ditentukan. Program akan menyimpan nama-nama bunga yang dimasukkan dan dapat menghentikan proses input jika pengguna mengetik "SELESAI". Di akhir, program akan menampilkan daftar bunga yang telah dimasukkan dalam bentuk string yang dipisahkan dengan tanda " - " serta jumlah bunga yang telah dimasukkan.

* **Algoritma**

1. Inisialisasi Variabel:

* Program meminta pengguna untuk memasukkan jumlah bunga yang ingin dimasukkan.
* Buat slice untuk menyimpan nama-nama bunga.

2. Pengambilan Input:

* Dalam loop, program meminta pengguna untuk memasukkan nama bunga satu per satu.
* Setelah setiap masukan, program memeriksa apakah masukan adalah "SELESAI". Jika ya, program akan menghentikan pengambilan input.

3. Penyimpanan Data:

* Nama bunga akan disimpan dalam slice `flowers`.

4. Menampilkan Hasil:

* Setelah proses input selesai, program akan menggabungkan semua nama bunga yang telah dimasukkan menjadi satu string yang dipisahkan oleh " - ".
* Program juga akan menampilkan jumlah bunga yang telah dimasukkan.
* **Cara Kerja Program**

1. Input Pengguna:

* Program meminta pengguna untuk memasukkan jumlah bunga yang ingin diinputkan.

2. Input Nama Bunga:

* Program meminta pengguna untuk memasukkan nama bunga satu per satu.
* Jika pengguna mengetik "SELESAI", program akan menghentikan proses input.

3. Penyimpanan dan Pengolahan Data:

* Setiap nama bunga akan disimpan dalam slice.
* Setelah input selesai, semua nama bunga dalam slice digabung menjadi satu string dengan " - ".

4. Menampilkan Hasil:

* Program menampilkan string hasil penggabungan dan jumlah bunga yang telah dimasukkan.

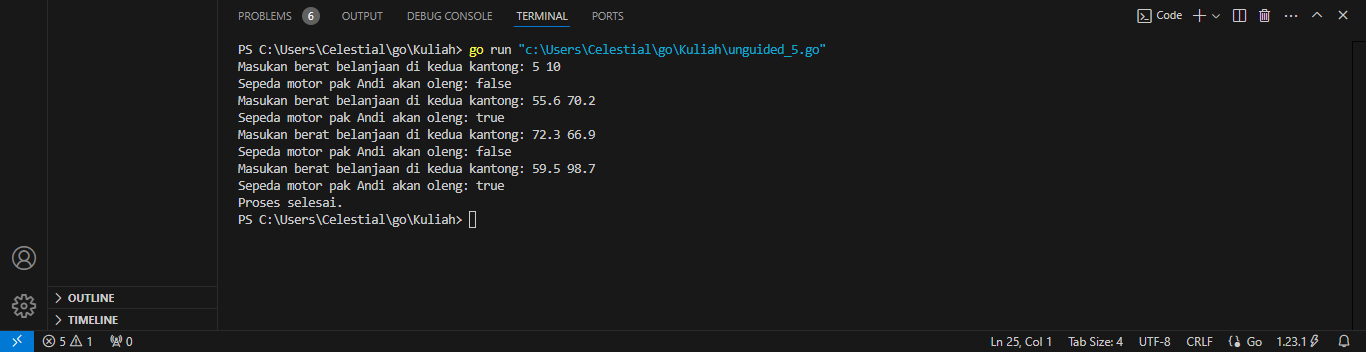
1. Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebüt hingga salah satu kantong terpal berisi 9 lig atau lebih.

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

**Sourcecode**

|  |
| --- |
| package main  import (      "bufio"      "fmt"      "os"      "strconv"      "strings"  )  func main() {      scanner := bufio.NewScanner(os.Stdin)      for {          fmt.Print("Masukan berat belanjaan di kedua kantong: ")          if !scanner.Scan() {              break          }          input := scanner.Text()          weights := strings.Fields(input)          if len(weights) != 2 {              fmt.Println("Input tidak valid.")              continue          }          weight1, err1 := strconv.ParseFloat(weights[0], 64)          weight2, err2 := strconv.ParseFloat(weights[1], 64)          if err1 != nil || err2 != nil {              fmt.Println("Input tidak valid.")              continue          }          fmt.Print("Sepeda motor pak Andi akan oleng: ")          difference := weight1 - weight2          if difference < 0 {              difference = -difference          }          if difference >= 9 {              fmt.Println("true")          } else {              fmt.Println("false")          }          totalWeight := weight1 + weight2          if totalWeight > 150 {              fmt.Println("Proses selesai.")              break          }      }  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa pemrograman Go dan berfungsi untuk memeriksa berat belanjaan dalam dua kantong. Program akan meminta pengguna untuk memasukkan berat kedua kantong secara berulang hingga total berat melebihi 150. Program juga akan menentukan apakah sepeda motor pak Andi akan oleng berdasarkan perbedaan berat antara kedua kantong tersebut.

* **Algoritma**

1. Inisialisasi Scanner:

* Menggunakan `bufio.Scanner` untuk membaca input dari pengguna.

2. Loop Input:

* Program berjalan dalam loop tanpa batas hingga pengguna memasukkan input yang valid atau total berat melebihi 150.

3. Validasi Input:

* Memeriksa apakah input terdiri dari dua bagian. Jika tidak, tampilkan pesan kesalahan dan ulangi permintaan input.
* Mengonversi kedua input ke tipe `float64`. Jika konversi gagal, tampilkan pesan kesalahan dan ulangi permintaan input.

4. Perhitungan dan Keluaran:

* Menghitung selisih antara kedua berat dan menampilkan, apakah sepeda motor akan oleng:
* Jika selisih berat lebih besar dari atau sama dengan 9, tampilkan `true`; jika tidak, tampilkan `false`.
* Menghitung total berat kedua kantong. Jika total berat lebih dari 150, program menampilkan "Proses selesai." dan keluar dari loop.
* **Cara Kerja Program**

1. Input Pengguna:

* Program meminta pengguna untuk memasukkan berat kedua kantong berulang kali.

2. Validasi dan Penyimpanan:

* Input diperiksa untuk memastikan bahwa itu valid (dua angka).
* Input yang valid diubah menjadi `float64` untuk keperluan perhitungan.

3. Logika Penilaian:

* Program menghitung perbedaan berat antara kedua kantong dan menentukan apakah sepeda motor akan oleng berdasarkan nilai tersebut.
* Program juga menghitung total berat dan mengecek apakah total tersebut melebihi 150.

4. Menampilkan Hasil:

* Hasil dari perhitungan ditampilkan kepada pengguna, dengan indikasi apakah sepeda motor akan oleng.

1. Diberikan sebuah persamaan sebagai berikut ini.

f(k) = (4k+2)² /(4k+1)(4k+3)

Buatlah sebuah program yang menerima input sebuah bilangan sebagai K, hemudian menghitüng dan menampilkan nilai f(K) sesuai persamaan di atas.

√2 merupakan bilangan irasional. Meskipun demikian, nilai tersebut dapat dihampin dengan rumus berikut:

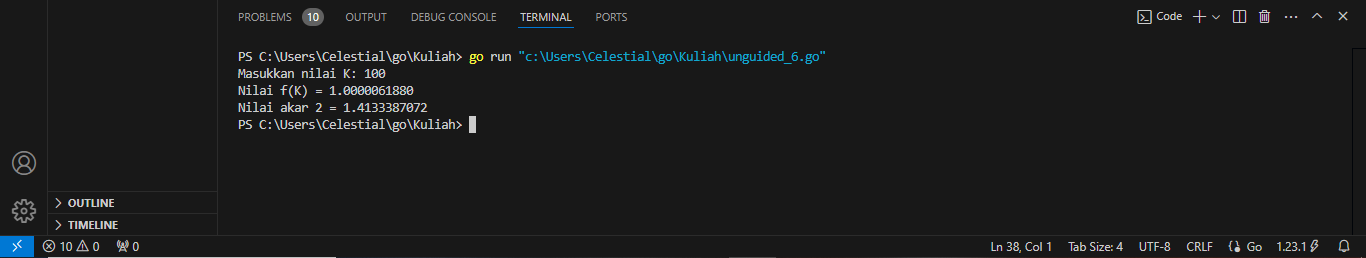
√Σ-Π П (4+2)² (4k+1)(4k+3)

Modifikası program sebelumnya yang menerima input integer K dan menghitung √2 untuk K tersebut. Hampiran v2 dituliskan dalam ketelitian 10 angka di belakang koma.

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"  )  func calculateF(k int) float64 {      numerator := (4\*float64(k) + 2) \* (4\*float64(k) + 2)      denominator := (4\*float64(k) + 1) \* (4\*float64(k) + 3)      return numerator / denominator  }  func calculateSqrt2(k int) float64 {      sqrt2 := 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)          sqrt2 \*= numerator / denominator      }      return sqrt2  }  func main() {      var k int      fmt.Print("Masukkan nilai K: ")      fmt.Scan(&k)        fK := calculateF(k)      fmt.Printf("Nilai f(K) = %.10f\n", fK)      sqrt2 := calculateSqrt2(k)      fmt.Printf("Nilai akar 2 = %.10f\n", sqrt2)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini ditulis dalam bahasa Go dan digunakan untuk menghitung dua nilai: fungsi `f(K)` dan estimasi dari nilai akar 2. Fungsi `f(K)` dihitung berdasarkan suatu formula dengan nilai `K` yang diinput oleh pengguna, dan nilai estimasi akar 2 dihitung menggunakan perhitungan berbasis nilai `K`.

* **Algoritma**

1. Fungsi `calculateF(k)`:

* Fungsi ini menghitung nilai `f(K)` berdasarkan rumus:

(4k+2)² /(4k+1)(4k+3)

* `k` adalah bilangan bulat yang diinput oleh pengguna. Fungsi ini akan mengembalikan nilai float hasil perhitungan tersebut.

1. Fungsi `calculateSqrt2(k)`:

* Fungsi ini menghitung nilai estimasi akar 2 menggunakan produk dari serangkaian perhitungan yang melibatkan variabel `i` dari 0 hingga `K`. Rumusnya sama dengan `f(K)`, namun dilakukan dalam loop: √Σ-Π П (4+2)² (4k+1)(4k+3)
* Nilai `sqrt2` diinisialisasi dengan 1 dan terus dikalikan dengan hasil dari perhitungan tersebut untuk setiap iterasi hingga mencapai `K`.

1. Fungsi `main()`:

* Program utama meminta input pengguna berupa nilai `K`.
* Menghitung `f(K)` menggunakan fungsi `calculateF(k)`.
* Menghitung nilai estimasi akar 2 menggunakan fungsi `calculateSqrt2(k)`.
* Menampilkan hasil perhitungan `f(K)` dan estimasi akar 2 dalam format desimal dengan 10 angka di belakang koma.
* Cara Kerja Program

1. Input Pengguna:

* Program meminta pengguna memasukkan nilai `K` yang merupakan bilangan bulat positif.

1. Perhitungan:

* Fungsi `calculateF(k)` menghitung dan mengembalikan nilai `f(K)` berdasarkan formula yang sudah ditentukan.
* Fungsi `calculateSqrt2(k)` melakukan loop dari 0 hingga `K` untuk menghitung estimasi akar 2 dengan cara mengalikan hasil perhitungan setiap iterasi ke variabel `sqrt2`.

1. Output:

* Program menampilkan nilai `f(K)` dan estimasi akar 2 dengan presisi 10 angka di belakang koma.