**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL II**

**REVIEW STRUKTUR CONTROL**

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

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Didik Setiawan / 2311102030**

**IF 11 06**

**Dosen Pengampu :**

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

1. **DASAR TEORI**

Dasar Teori

Go didasarkan pada paradigma pemrograman terstruktur, yang menekankan pentingnya pengorganisasian kode dalam blok-blok yang teratur. Bahasa ini mendukung elemen-elemen struktural seperti pernyataan if-else, perulangan for, dan switch, yang memungkinkan pengembang untuk mengendalikan alur program dengan lebih jelas dan sistematis.

Tipe Data dan Deklarasi Variabel

Go memiliki tipe data dasar seperti int (bilangan bulat), float64 (bilangan desimal), string (teks), dan bool (nilai boolean). Untuk mendeklarasikan variabel dalam Go, Anda perlu menyebutkan tipe data yang diinginkan, diikuti dengan nama variabel, dan nilai yang bersifat opsional.

Kesimpulan

Go adalah bahasa pemrograman yang mengutamakan struktur dalam penulisan kode, dengan penggunaan elemen-elemen seperti pernyataan if-else, perulangan for, dan switch untuk mengatur alur eksekusi program secara jelas. Selain itu, Go menyediakan tipe data dasar seperti int, float64, string, dan bool, yang dapat dideklarasikan dengan menyebutkan tipe data, nama variabel, dan nilai opsional.

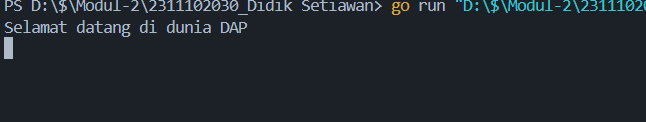
1. **GUIDED**
   * + 1. **Soal Studi Case 2.1**

Struktur Program Go

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var greetings = "Selamat datang di dunia DAP"      var a, b int      fmt.Println(greetings)      fmt.Scanln(&a, &b)      fmt.Printf("%v + %v = %v\n", a, b, a+b)  } |

**Screenshoot Output**

****

**Deskripsi Program**

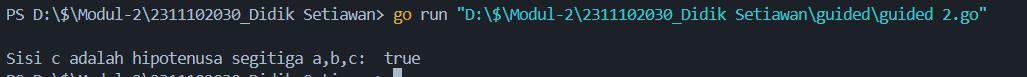
untuk menjumlahkan dua angka yang dimasukkan oleh pengguna. Setelah menampilkan pesan sambutan "Selamat datang di dunia DAP", program meminta pengguna untuk memasukkan dua angka. Kemudian, program menghitung penjumlahan kedua angka tersebut dan menampilkan hasilnya dalam format a + b = hasil.

* + - 1. **Soal Studi Case 2.2**

Tipe Data dan Instruksi Dasar **Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var a, b, c float64      var hipotenusa bool      fmt.Scanln(&a, &b, &c)      hipotenusa = (c \* c) == (a\*a + b\*b)      fmt.Println("Sisi c adalah hipotenusa segitiga a,b,c: ", hipotenusa)  } |

**Screenshoot Output**

****

**Deskripsi Program**

memeriksa apakah sisi c dari segitiga dengan sisi-sisi a, b, dan c merupakan hipotenusa. Pengguna diminta untuk memasukkan nilai untuk a, b, dan c. Setelah itu, program menghitung apakah =+ Jika kondisi tersebut terpenuhi, variabel hipotenusa akan diatur menjadi true, yang menunjukkan bahwa c adalah hipotenusa.

* + - 1. **Soal Studi Case 2.3**

Struktur Program Go

**Sourcecode**

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

**Screenshoot Output**

**A screen shot of a computer

Description automatically generated**

**Deskripsi Program**

Program ini menghasilkan deret Fibonacci hingga nilai maksimum 100. Dimulai dengan mendeklarasikan tiga variabel untuk menyimpan bilangan Fibonacci: f0 (0), f1 (1), dan f2 (1). Setelah mencetak bilangan pertama, program menggunakan perulangan untuk menghitung bilangan Fibonacci berikutnya selama bilangan tersebut tidak melebihi 100. Setiap iterasi memperbarui nilai f0, f1, dan f2, dan mencetak bilangan berikutnya dalam deret.

* + - 1. **Soal Studi Case 2 A 1**

Telusuri program berikut dengan cara mengkompilasi dan mengeksekusi program. Silakan masukan data yang sesuai sesuai sebanyak sebanyak yang diminta program. Perhatikan keluaran yang diperoleh. Coba terangkan apa sebenarnya yang dilakukan program tersebut?

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var (          satu, dua311102030, tiga string          temp                     string      )      fmt.Print("Masukan input string: ")      fmt.Scanln(&satu)      fmt.Print("Masukan input string: ")      fmt.Scanln(&dua311102030)      fmt.Print("Masukan input string: ")      fmt.Scanln(&tiga)      fmt.Println("Output awal = " + satu + " " + dua311102030 + " " + tiga)      temp = satu      satu = dua311102030      dua311102030 = tiga      tiga = temp      fmt.Println("Output akhir = " + satu + " " + dua311102030 + " " + tiga)  } |

**Screenshoot Output**

**A screen shot of a computer

Description automatically generated**

**Deskripsi Program**

mencetak output awal yang berisi ketiga string tersebut. Setelah itu, program melakukan pertukaran nilai string: nilai dari variabel pertama (satu) akan dipindahkan ke variabel kedua (dua311102030), nilai dari variabel kedua akan dipindahkan ke variabel ketiga (tiga), dan nilai dari variabel ketiga akan disimpan kembali ke variabel pertama. Setelah pertukaran, program mencetak output akhir yang menunjukkan string yang telah diubah urutannya.

* + - 1. **Soal Studi Case 2 A 2**

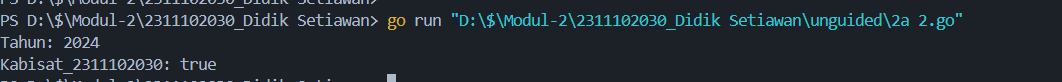
Tahun kabisat adalah tahun yang habis dibagi 400 atau habis dibagi 4 tetapi tidak habis dibagi 100. Buatlah sebuah program yang menerima input sebuah bilangan bulat dan memeriksa apakah bilangan tersebut

merupakan tahun kbisat (true) atau bukan (false).

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var tahun int      var kabisat\_2311102030 bool      fmt.Print("Tahun: ")      fmt.Scanln(&tahun)      if tahun%400 == 0 {          kabisat\_2311102030 = true      } else if tahun%100 == 0 {          kabisat\_2311102030 = false      } else if tahun%4 == 0 {          kabisat\_2311102030 = true      } else {          kabisat\_2311102030 = false      }      fmt.Println("Kabisat\_2311102030:", kabisat\_2311102030)  } |

**Screenshoot Output**

****

**Deskripsi Program**

menentukan apakah tahun yang dimasukkan oleh pengguna adalah tahun kabisat. Setelah meminta pengguna untuk memasukkan sebuah tahun, program menggunakan serangkaian pernyataan kondisi untuk memeriksa apakah tahun tersebut memenuhi kriteria tahun kabisat. Kriteria tersebut adalah: tahun yang dapat dibagi oleh 400 adalah tahun kabisat tahun yang dapat dibagi oleh 100 tetapi tidak oleh 400 bukan tahun kabisat

* + - 1. **Soal Studi Case 2 A 3**

Buat program Bola yang menerima input jari-jari suatu bola (bilangan bulat). Tampilkan Volume dan Luas kulit bola. volumebola ar³ dan luasbola = 4πr² (π 4 3 3.1415926535).

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func main() {      var jari2 float64      const pi = 3.1415926535      fmt.Print("Jejari = ")      fmt.Scanln(&jari2)      volume := (4.0 / 3.0) \* pi \* math.Pow(jari2, 3)      luas := 4 \* pi \* math.Pow(jari2, 2)      fmt.Printf("Bola dengan jari2 %.0f mempunyai volume %.4f dan luas kulit %.4f\n", jari2, volume, luas)  } |

**Screenshoot Output**

****

**Deskripsi Program**

Program ini menghitung volume dan luas permukaan bola berdasarkan jejari yang dimasukkan oleh pengguna. Setelah meminta pengguna untuk memasukkan nilai jejari, program menghitung volume menggunakan rumus dan menghitung luas permukaan 4πr²

1. **UNGUIDED**

**Soal Studi Case 2A 4**

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

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var celsius float64      // Input suhu dalam Celsius      fmt.Print("Masukkan suhu dalam derajat Celsius: ")      fmt.Scanf("%f", &celsius)      // Hitung suhu dalam berbagai satuan      fmt.Printf("Temperatur Celsius: %.2f\n", celsius)      fmt.Printf("Derajat Reaumur: %.2f\n", (4.0/5.0)\*celsius)      fmt.Printf("Derajat Fahrenheit: %.2f\n", (celsius\*9.0/5.0)+32.0)      fmt.Printf("Derajat Kelvin: %.2f\n", celsius+273.15)  } |

**Screenshoot Output**

**A screenshot of a computer screen

Description automatically generated**

**Deskripsi Program**

Program ini mengonversi suhu dari derajat Celsius ke berbagai satuan suhu lainnya. Setelah meminta pengguna untuk memasukkan nilai suhu dalam Celsius, program menghitung dan mencetak suhu yang setara dalam derajat Reaumur, Fahrenheit, dan Kelvin. Konversi dilakukan dengan menggunakan rumus yang sesuai, dan hasilnya ditampilkan dengan format dua angka desimal untuk memudahkan pembacaan

**Soal Studi Case 2A 5**

Berikut adalah salinan teks dari gambar: Tipe karakter sebenarnya hanya apa yang tampak dalam tampilan. Di dalamnya tersimpan dalam bentuk biner 8 bit (byte) atau 32 bit (rune) saja. Buat program ASCII yang akan membaca 5 buah data integer dan mencetaknya dalam format karakter. Kemudian membaca 3 buah data karakter dan mencetak 3 buah karakter setelah karakter tersebut (menurut tabel ASCII). Masukan terdiri dari dua baris. Baris pertama berisi 5 buah data integer. Data integer mempunyai nilai antara 32 s.d. 127. Baris kedua berisi 3 buah karakter yang berdampingan satu dengan yang lain (tanpa dipisahkan spasi).

**Sourcecode**

|  |
| --- |
| package main  import (      "bufio"      "fmt"      "os"  )  func main() {      var nums [5]int      reader := bufio.NewReader(os.Stdin)      // Input angka dari 32 hingga 127 dan cetak sebagai karakter      fmt.Println("Masukkan 5 buah angka (nilai antara 32 hingga 127):")      fmt.Scanf("%d %d %d %d %d", &nums[0], &nums[1], &nums[2], &nums[3], &nums[4])      fmt.Print("Hasil karakter dari angka yang diinput: ")      for \_, num := range nums {          fmt.Printf("%c", num)      }      fmt.Println()      // Input string karakter dan geser setiap karakter      fmt.Println("Masukkan karakter (contoh input: 123):")      inputStr, \_ := reader.ReadString('\n')      fmt.Print("Hasil karakter dari inputan angka yang digeser: ")      for \_, char := range inputStr {          if char != '\n' && char != '\r' {              fmt.Printf("%c", char+1)          }      }      fmt.Println()  } |

**Screenshoot Output**

**A computer screen with blue text

Description automatically generated**

**Deskripsi Program**

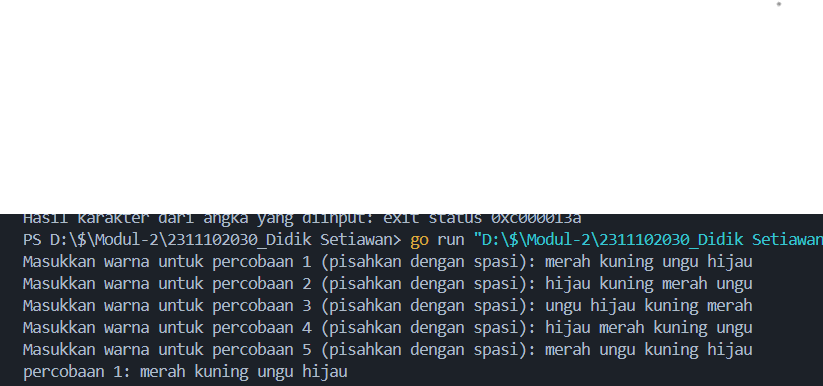
melakukan dua fungsi utama: pertama, meminta pengguna untuk memasukkan lima angka dalam rentang 32 hingga 127 dan mencetak karakter yang sesuai dengan angka-angka tersebut. Setelah mencetak karakter yang dihasilkan, program kemudian meminta pengguna untuk memasukkan sebuah string. Setiap karakter dalam string yang dimasukkan akan digeser satu posisi ke depan dalam tabel ASCII, dan hasilnya dicetak ke layar. Program ini menggunakan bufio untuk membaca input string dari pengguna dan menampilkan karakter hasil geseran.

**Soal Studi Case 2b 1**

Siswa kelas IPA disalah satu sekolah menengah atas di indonesia sedang mengadakan praktikum kimia. Di setiap percobaan akan menggunakan 4 tabung reaksi, yang mana susunan warna cairan di setiap tabung akan menentukan hasil percobaan. Siswa diminta untuk mencatat hasil percobaan tersebut. percobaan dikatakan berhasil apabila susunan warna zat cair pada gelas 1 hingga gelas 4 secara 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 (      "fmt"  )  func main() {      var warna [5][4]string      urutanBenar := [4]string{"merah", "kuning", "hijau", "ungu"}      berhasil := true      for i := 0; i < 5; i++ {          fmt.Printf("Masukkan warna untuk percobaan %d (pisahkan dengan spasi): ", i+1)          fmt.Scan(&warna[i][0], &warna[i][1], &warna[i][2], &warna[i][3])      }      for i := 0; i < 5; i++ {          fmt.Printf("percobaan %d: %s %s %s %s\n", i+1, warna[i][0], warna[i][1], warna[i][2], warna[i][3])          for j := 0; j < 4; j++ {              if warna[i][j] != urutanBenar[j] {                  berhasil = false                  break              }          }          if !berhasil {              break          }      }      fmt.Printf("berhasil: %t\n", berhasil)  } |

**Screenshoot Output**

****

**Deskripsi Program**

memasukkan warna dalam lima percobaan, masing-masing dengan empat warna yang dipisahkan oleh spasi. Setelah semua warna dimasukkan, program mencetak hasil percobaan dan membandingkan setiap set warna yang dimasukkan dengan urutan yang benar, yaitu "merah", "kuning", "hijau", dan "ungu". Jika semua warna dalam percobaan tertentu sesuai dengan urutan yang benar, program menandai hasilnya sebagai berhasil. Jika ada ketidaksesuaian, hasilnya akan ditandai sebagai gagal, dan program akan menghentikan pemeriksaan lebih lanjut. Akhirnya, program mencetak status keberhasilan berdasarkan hasil pemeriksaan.

**Soal Studi Case 2b 2**

Suatu pita(string) berisi kumpulan nama-nama bunga yang dipisahkan oleh spasi dan '\_', contoh pita diilustrasikan seperti berikut ini Pita: mawar=melati-tulip-teratai-kamboja-anggrek Buatlah sebuah program yang menerima input sebuah bilangan bulat positif(dan tidak nol) N, kemudian program akan meminta input berupa nama bunga secara berulang sebanyak N kali dan nama tersebut disimpan ke dalam pita. Tampilkan isi pita setelah proses input selesai. Perhatikan contoh sesi interaksi program seperti di bawah ini (teks bergaris bawah adalah input/read) Modifikasi program sebelumnya, proses input akan berhenti apabila user mengetiikan 'SELESAI'. Kemudian tampilkan isi pita beserta banyaknya bunga yang ada di dalam pita perhatikan contoh sesi interaksi program seperti dibawah ini (teks bergaris bawah adalah input/read):

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "strings"  )  func main() {      var N int      fmt.Print("Masukkan jumlah bunga (N): ")      fmt.Scan(&N)      var pita strings.Builder      var bunga string      for i := 1; i <= N; i++ {          fmt.Printf("Bunga %d: ", i)          fmt.Scan(&bunga)          // Jika input adalah "SELESAI", hentikan input          if strings.ToUpper(bunga) == "SELESAI" {              break          }          // Tambahkan tanda "-" sebelum bunga, kecuali bunga pertama          if pita.Len() > 0 {              pita.WriteString(" - ")          }          pita.WriteString(bunga)      }      // Output hasil jika ada bunga yang dimasukkan      if pita.Len() > 0 {          fmt.Printf("Pita: %s\n", pita.String())          fmt.Printf("Bunga: %d\n", strings.Count(pita.String(), " - ")+1)      } else {          fmt.Println("Tidak ada bunga yang dimasukkan.")      }  } |

**Screenshoot Output**

**A screenshot of a computer

Description automatically generated**

**Deskripsi Program**

memasukkan jumlah bunga yang diinginkan (N). Setelah itu, pengguna akan diminta untuk memasukkan nama setiap bunga satu per satu. Jika pengguna memasukkan "SELESAI", program akan menghentikan input. Setiap bunga yang dimasukkan akan ditambahkan ke dalam string pita, yang dipisahkan oleh tanda "-".

**Soal Studi Case 2b 3**

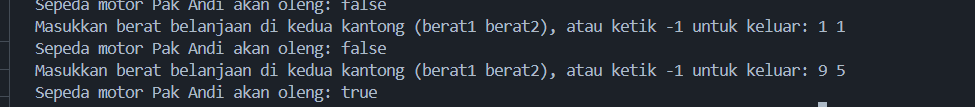
Buatlah program Pak Andi yang menerima input dua buah bilangan real positif yang menyatakan berat total masing-masing isi kantong terpal. Program akan terus meminta input bilangan tersebut hingga salah satu kantong terpal berisi 9 kg atau lebih.

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

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"  )  func main() {      var berat1, berat2 float64      for {          fmt.Print("Masukkan berat belanjaan di kedua kantong (berat1 berat2), atau ketik -1 untuk keluar: ")          fmt.Scan(&berat1, &berat2)          // Berikan opsi untuk keluar dari loop          if berat1 == -1 || berat2 == -1 {              fmt.Println("Program selesai.")              break          }          // Cek apakah salah satu kantong lebih dari atau sama dengan 9 kg          if berat1 >= 9 || berat2 >= 9 {              fmt.Println("Sepeda motor Pak Andi akan oleng: true")          } else {              fmt.Println("Sepeda motor Pak Andi akan oleng: false")          }      }  } |

**Screenshoot Output**

****

**Deskripsi Program**

Jika berat salah satu kantong melebihi atau sama dengan 9 kg, program akan mencetak bahwa sepeda motor akan oleng (true), sebaliknya jika keduanya di bawah 9 kg, motor dianggap stabil (false).

**Soal Studi Case 2b 4**

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

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  // Fungsi untuk menghitung nilai f(k)  func hitungF(k int) float64 {      pembilang := math.Pow(float64(4\*k+2), 2)      penyebut := float64((4\*k + 1) \* (4\*k + 3))      return pembilang / penyebut  }  // Fungsi untuk menghitung nilai perkiraan akar 2 dengan iterasi hingga k  func hitungAkarDua(k int) float64 {      hasil := 1.0      for i := 0; i <= k; i++ {          hasil \*= hitungF(i)      }      return hasil  }  func main() {      var k int      fmt.Print("Masukkan nilai K (>= 0): ")      fmt.Scan(&k)      // Validasi nilai k agar tidak negatif      if k < 0 {          fmt.Println("Nilai K harus lebih besar atau sama dengan 0.")          return      }      nilaiF := hitungF(k)      fmt.Printf("Nilai f(%d) = %.10f\n", k, nilaiF)      akarDua := hitungAkarDua(k)      fmt.Printf("Nilai perkiraan akar 2 untuk K = %d: %.10f\n", k, akarDua)  } |

**Screenshoot Output**

**A screenshot of a computer

Description automatically generated**

**Deskripsi Program**

menerima nilai k sebagai input, yang menentukan jumlah iterasi dalam perhitungan. Fungsi hitungF(k) digunakan untuk menghitung nilai tertentu berdasarkan rumus matematika, sementara fungsi hitungAkarDua(k) menghitung perkiraan akar kuadrat dari 2 dengan mengalikan hasil dari beberapa iterasi fungsi hitungF(k). Jika pengguna memasukkan nilai k yang negatif, program akan memberi tahu bahwa nilai k harus lebih besar atau sama dengan 0.