**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL IV**

**PROSEDUR**

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

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Didik Setiawan**

**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

Dalam pemrograman komputer, prosedur adalah modul kode independen yang memenuhi beberapa tugas konkret dan dirujuk dalam badan kode sumber yang lebih besar.

Item kode semacam ini juga dapat disebut fungsi atau sub-rutin.

Peran mendasar dari suatu prosedur adalah untuk menawarkan satu titik referensi untuk beberapa tujuan atau tugas kecil yang dapat dipicu pengembang atau programmer dengan memanggil prosedur itu sendiri.

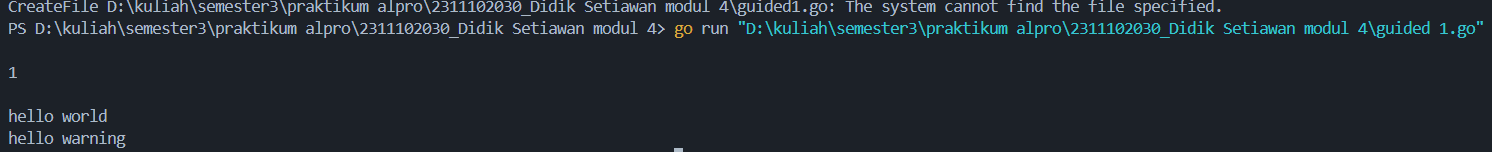
1. **GUIDED**
   * + 1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

**Soal Studi Case**

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func main() {      var bilangan int      var pesan string      fmt.Scan(&bilangan, &pesan)      cetakPesan(pesan, bilangan)  }  func cetakPesan(M string, flag int) {      var jenis string = ""      if flag == 0 {          jenis = "error"      } else if flag == 1 {          jenis = "warning"      } else if flag == 2 {          jenis = "informasi"      }      fmt.Println(M, jenis)  } |

**Screenshoot Output**

****

**Deskripsi Program**

menerima dua input dari pengguna: sebuah bilangan bulat dan sebuah pesan string. Berdasarkan nilai bilangan yang dimasukkan, program menentukan jenis pesan yang akan dicetak: "error" untuk nilai 0, "warning" untuk nilai 1, dan "informasi" untuk nilai 2. Setelah menentukan jenisnya, program akan menampilkan pesan tersebut diikuti dengan jenis pesan yang relevan

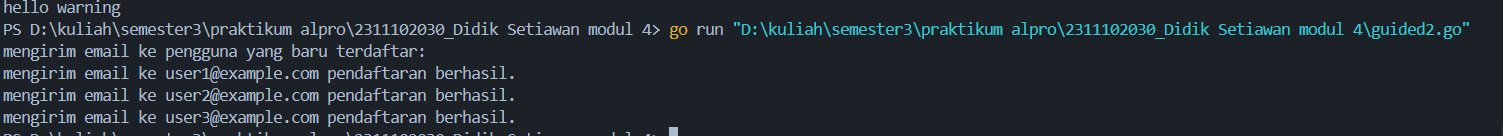
* + - 1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

**Soal Studi Case**

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func sendEmailNotification(email string) {      fmt.Printf("mengirim email ke %s pendaftaran berhasil. \n", email)  }  func main() {      emails := []string{"user1@example.com", "user2@example.com", "user3@example.com"}      fmt.Println("mengirim email ke pengguna yang baru terdaftar:")      for \_, email := range emails {          sendEmailNotification(email)      }  } |

**Screenshoot Output**

****

**Deskripsi Program**

mendefinisikan fungsi sendEmailNotification yang menerima parameter alamat email dan mencetak pesan bahwa email telah dikirim untuk mengonfirmasi pendaftaran. Di dalam fungsi main, program menginisialisasi sebuah slice yang berisi beberapa alamat email, kemudian menggunakan loop untuk mengiterasi setiap alamat email dan memanggil fungsi sendEmailNotification untuk masing-masing alamat.

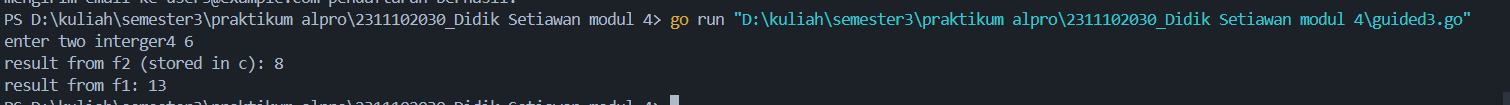
* + - 1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

**Soal Studi Case**

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  func f1(x, y int) float64 {      var hasil float64      hasil = float64(2\*x) - 0.5\*float64(y) + 3.0      return hasil  }  func f2(x, y int, hasil \*float64) {      \*hasil = float64(2\*x) - 0.5\*float64(y) + 3.0  }  func main() {      var a, b int      var c float64      fmt.Print("enter two interger")      fmt.Scan(&a, &b)      f2(a, b, &c)      fmt.Println("result from f2 (stored in c):", c)      resultF1 := f1(b, a)      fmt.Println("result from f1:", resultF1)  } |

**Screenshoot Output**

****

**Deskripsi Program**

mendefinisikan dua fungsi: f1, yang menerima dua parameter dan mengembalikan hasil perhitungan sebagai tipe float64, dan f2, yang menggunakan pointer untuk menyimpan hasil perhitungan langsung ke variabel yang diberikan

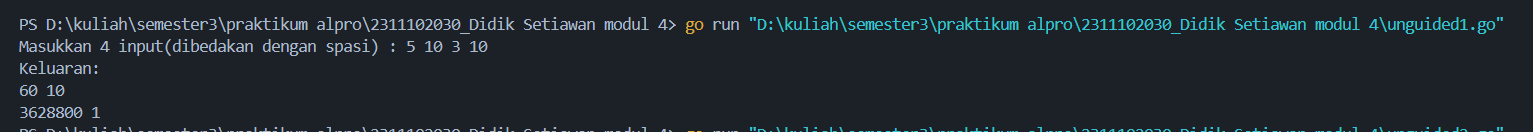
1. **UNGUIDED**
   * + 1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

**Soal Studi Case**

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"  )  func factorial(n int) int {      if n == 0 {          return 1      }      result := 1      for i := 2; i <= n; i++ {          result \*= i      }      return result  }  func permutasi(n, r int) int {      return factorial(n) / factorial(n-r)  }  func kombinasi(n, r int) int {      return factorial(n) / (factorial(r) \* factorial(n-r))  }  func main() {      var a, b, c, d int      fmt.Print("Masukkan 4 input(dibedakan dengan spasi) : ")      fmt.Scan(&a, &b, &c, &d)      if a >= c && b >= d {          permutasiA := permutasi(a, c)          kombinasiA := kombinasi(a, c)          permutasiB := permutasi(b, d)          kombinasiB := kombinasi(b, d)          fmt.Println("Keluaran:")          fmt.Println(permutasiA, kombinasiA)          fmt.Println(permutasiB, kombinasiB)      } else {          fmt.Println("Syarat tidak terpenuhi")      }  } |

**Screenshoot Output**

****

**Deskripsi Program**

mendefinisikan fungsi factorial untuk menghitung faktorial dari suatu bilangan, permutasi untuk menghitung nilai permutasi berdasarkan dua argumen, dan kombinasi untuk menghitung nilai kombinasi. Dalam fungsi main, program meminta pengguna untuk memasukkan empat bilangan bulat, kemudian memeriksa apakah dua bilangan pertama (a dan b) lebih besar atau sama dengan dua bilangan terakhir (c dan d). Jika syarat tersebut terpenuhi, program akan menghitung dan mencetak nilai permutasi dan kombinasi untuk kedua pasangan bilangan (a, c) dan (b, d). Jika syarat tidak terpenuhi, program akan menampilkan pesan bahwa syarat tidak terpenuhi

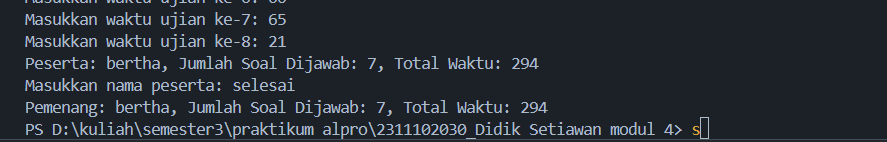
2. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

**Soal Studi Case**

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "strings"  )  func hitungSkor(namaPeserta \*string, jumlahSoalDijawab \*int, totalWaktu \*int) {      var waktuUjianPerSoal [8]int      \*jumlahSoalDijawab = 0      \*totalWaktu = 0      fmt.Print("Masukkan nama peserta: ")      fmt.Scan(namaPeserta)      if strings.ToLower(\*namaPeserta) == "selesai" {          return      }      for i := 0; i < 8; i++ {          fmt.Printf("Masukkan waktu ujian ke-%d: ", i+1)          fmt.Scan(&waktuUjianPerSoal[i])          if waktuUjianPerSoal[i] != 301 {              (\*jumlahSoalDijawab)++              \*totalWaktu += waktuUjianPerSoal[i]          }      }  }  func main() {      var namaPemenang string      var soalTerbanyak, waktuTercepat int      for {          var namaPeserta string          var jumlahSoalDijawab, totalWaktu int          hitungSkor(&namaPeserta, &jumlahSoalDijawab, &totalWaktu)          // Cek apakah nama peserta adalah "selesai"          if strings.ToLower(namaPeserta) == "selesai" {              break          }          fmt.Printf("Peserta: %s, Jumlah Soal Dijawab: %d, Total Waktu: %d\n", namaPeserta, jumlahSoalDijawab, totalWaktu)          if jumlahSoalDijawab > soalTerbanyak || (jumlahSoalDijawab == soalTerbanyak && totalWaktu < waktuTercepat) {              namaPemenang = namaPeserta              soalTerbanyak = jumlahSoalDijawab              waktuTercepat = totalWaktu          }      }      fmt.Printf("Pemenang: %s, Jumlah Soal Dijawab: %d, Total Waktu: %d\n", namaPemenang, soalTerbanyak, waktuTercepat)  } |

**Screenshoot Output**

****

**Deskripsi Program**

mendefinisikan fungsi hitungSkor, yang meminta pengguna untuk memasukkan nama peserta dan waktu yang dihabiskan untuk setiap soal. Waktu ujian yang dimasukkan ke dalam array hanya dihitung jika tidak sama dengan 301 (yang tampaknya merupakan indikator soal yang tidak dijawab). Program kemudian menghitung jumlah soal yang dijawab dan total waktu yang dihabiskan

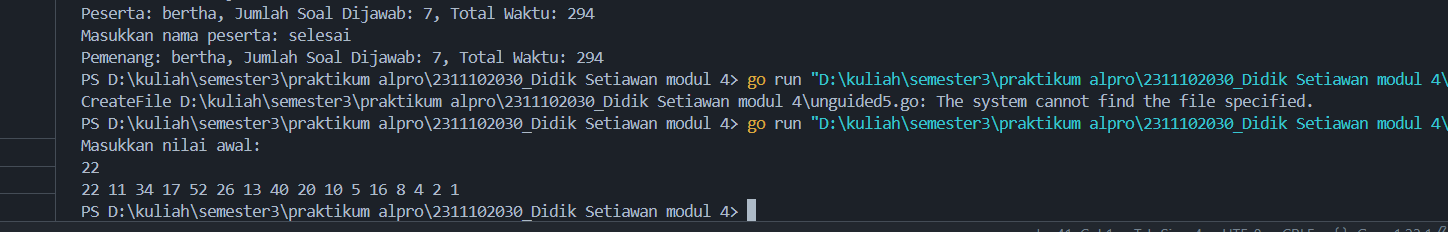
1. Berisi source code dan output dari kegiatan praktikum yang telah dilaksanakan. Source Code diberi penjelasan maka akan menjadi nilai ++

**Soal Studi Case**

**Sourcecode**

|  |
| --- |
| // 2311102037 BRIAN FARREL EVANDHIKA IF 11 06  package main  import (      "fmt"  )  // cetakDeret mencetak setiap elemen dari deret berdasarkan aturan yang diberikan.  func cetakDeret(angka int) {      for angka != 1 {          // Mencetak elemen saat ini          fmt.Printf("%d ", angka)          // Menghitung elemen berikutnya          if angka%2 == 0 {              angka = angka / 2          } else {              angka = 3\*angka + 1          }      }      // Mencetak angka 1 sebagai elemen terakhir      fmt.Println(1)  }  func main() {      var nilaiAwal int      // Meminta input dari pengguna      fmt.Print("Masukkan nilai awal: ")      fmt.Scan(&nilaiAwal)      // Memastikan nilaiAwal valid (kurang dari 1.000.000)      if nilaiAwal < 1 || nilaiAwal >= 1000000 {          fmt.Println("Nilai harus lebih dari 0 dan kurang dari 1.000.000")          return      }      // Memanggil prosedur cetakDeret untuk mencetak deret      cetakDeret(nilaiAwal)  } |

**Screenshoot Output**

****

**Deskripsi Program**

mencetak deret angka berdasarkan aturan yang dikenal sebagai "Collatz Conjecture" atau "Conjecture Hailstone." Program ini mendefinisikan fungsi cetakDeret yang mengambil sebuah angka sebagai parameter dan mencetak deret angka hingga mencapai 1. Aturan untuk menghasilkan elemen berikutnya dalam deret adalah: jika angka saat ini genap, angka tersebut dibagi dua; jika ganjil, angka tersebut dikalikan tiga dan ditambahkan satu.