**LAPORAN PRAKTIKUM ALGORITMA**

**DAN PEMROGRAMAN 1**

**MODUL 4**

**TIPE DATA & VARIABEL**

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

Konten yang dihasilkan AI mungkin salah.**

**Disusun oleh:**

**NAMA : PRADITYA PUTRA ZAENI**

**NIM :** **109082530013**

**S1IF-13-02**

**Asisten Praktikum**

Adithana dharma putra

Alfin Ilham Berlianto

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

**LATIHAN KELAS – GUIDED**

1. **Guided 1**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {  var detik, jam, menit int  fmt.Scan(&detik)jam = detik / 3600menit = (detik % 3600) / 60detik = detik % 60fmt  Println(jam, "jam", menit, "menit dan", detik,"detik")  } |

**Screenshoot program:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Deskripsi program :**

Program Go di atas digunakan untuk **mengubah waktu dalam satuan detik menjadi jam, menit, dan detik** secara lebih mudah dipahami. Pertama, program membuat tiga variabel: detik, jam, dan menit. Pengguna memasukkan sebuah angka dalam bentuk detik melalui fmt.Scan(&detik). Setelah itu, program menghitung berapa jam yang bisa dibentuk dari jumlah detik tersebut dengan membagi detik dengan 3600, karena satu jam terdiri dari 3600 detik. Lalu, sisa detiknya dihitung lagi untuk mencari berapa menit yang terbentuk. Caranya dengan mengambil sisa dari pembagian sebelumnya (detik % 3600) lalu dibagi 60, karena satu menit adalah 60 detik. Terakhir, sisa detik yang belum terpakai dihitung dengan detik % 60. Setelah semua dihitung, hasilnya ditampilkan dalam bentuk kalimat seperti “x jam y menit dan z detik” lewat fmt.Println.

1. **Guided 2**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var angka int      fmt.Print("Masukkan angka:")      fmt.Scan(&angka)      d1 := angka / 100      angka = angka % 100      d2 := angka / 10      d3 := angka % 10      hasilnya := d1 < d2 && d2 < d3      fmt.Println(hasilnya)  } |

**Screenshoot program**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Deskripsi program**

**Program Go tersebut digunakan untuk mengecek apakah tiga digit dari sebuah angka disusun secara urut menaik. Pertama, program meminta pengguna memasukkan sebuah angka melalui fmt.Scan(&angka). Setelah angka diterima, program mengambil digit pertamanya (ratusan) dengan d1 := angka / 100. Lalu nilai angka diubah menjadi sisa dari dua digit terakhir menggunakan angka = angka % 100. Dari sisa itu, digit puluhan dihitung dengan d2 := angka / 10, dan digit satuan diambil dengan d3 := angka % 10.**

**Setelah ketiga digit diperoleh, program memeriksa apakah digit pertama lebih kecil dari digit kedua, dan digit kedua lebih kecil dari digit ketiga, dengan ekspresi d1 < d2 && d2 < d3. Hasil pengecekan tersebut berupa nilai boolean (true atau false), lalu dicetak menggunakan fmt.Println(hasilnya). Jadi, program ini menentukan apakah tiga digit dari angka yang dimasukkan tersusun secara meningkat—misalnya 123 akan menghasilkan true, sedangkan 351 atau 454 akan menghasilkan false.**

1. **Guided 3**

**Source Code**

|  |
| --- |
| package main  package main  import "fmt"    func main() {      var bmi, tinggi float64      fmt.Print("Masukkan bmi: ")      fmt.Scan(&bmi)      fmt.Print("Masukkan tinggi: ")      fmt.Scan(&tinggi)      berat := bmi \* (tinggi \* tinggi)      fmt.Printf("berat: %.0f kg\n", berat)  } |

**Screenshoot program**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**Deskripsi program**

**Program Go ini digunakan untuk menghitung nilai BMI (Body Mass Index) berdasarkan berat badan dan tinggi badan yang dimasukkan pengguna. Pertama, program mendeklarasikan dua variabel bertipe float64, yaitu berat dan tinggi. Kemudian pengguna diminta memasukkan berat badan melalui fmt.Print("Masukkan berat: ") dan nilainya disimpan menggunakan fmt.Scan(&berat). Setelah itu, pengguna juga diminta mengisi tinggi badan melalui perintah serupa dan disimpan ke variabel tinggi. Setelah kedua data tersedia, program menghitung BMI dengan rumus bmi := berat / (tinggi \* tinggi), yang berarti berat dibagi kuadrat dari tinggi badan. Terakhir, hasil BMI ditampilkan ke layar menggunakan fmt.Printf dengan format dua angka di belakang koma. Program ini membantu mengetahui indeks massa tubuh berdasarkan data yang dimasukkan.**

**TUGAS**

1. **Tugas 1**

**Source Code**

|  |
| --- |
| package main  import "fmt"  func main() {      var totalAwal, diskonPersen int      fmt.Scan(&totalAwal)      fmt.Scan(&diskonPersen)      potongan := totalAwal \* diskonPersen / 100      totalAkhir := totalAwal - potongan      fmt.Println(totalAkhir)  } |

**Screenshoot program**

**A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Deskripsi program :** **Program Go ini berfungsi untuk menghitung total belanja setelah mendapatkan potongan diskon. Pertama, program mendeklarasikan dua variabel bertipe integer yaitu totalAwal untuk menampung total belanja sebelum diskon dan diskonPersen untuk menampung persentase diskon. Pengguna menginput kedua nilai tersebut melalui fmt.Scan. Setelah nilai diterima, program menghitung besar potongan diskon menggunakan rumus potongan := totalAwal \* diskonPersen / 100. Potongan tersebut kemudian dikurangi dari total awal, sehingga diperoleh nilai totalAkhir := totalAwal - potongan. Terakhir, program menampilkan hasil akhir belanja setelah diskon menggunakan fmt.Println(totalAkhir). Dengan demikian, program ini menyederhanakan proses perhitungan potongan harga berdasarkan persen diskon yang diberikan.**

**Tugas 2**

**Source code**

|  |
| --- |
| package main  import "fmt"    func main() {      var bmi, tinggi float64      fmt.Print("Masukkan bmi: ")      fmt.Scan(&bmi)      fmt.Print("Masukkan tinggi: ")      fmt.Scan(&tinggi)      berat := bmi \* (tinggi \* tinggi)      fmt.Printf("berat: %.0f kg\n", berat)  } |

**Screenshot Program:**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Deskripsi program :**

**Program Go tersebut digunakan untuk menghitung berat badan seseorang berdasarkan nilai BMI dan tinggi badan. Pertama, program mendeklarasikan dua variabel bertipe float64, yaitu bmi dan tinggi. Pengguna diminta memasukkan nilai BMI melalui perintah fmt.Print("Masukkan bmi: ") dan nilainya dibaca dengan fmt.Scan(&bmi). Selanjutnya, tinggi badan dalam meter diminta melalui fmt.Print("Masukkan tinggi: ") dan juga dibaca menggunakan fmt.Scan(&tinggi). Setelah kedua input diterima, program menghitung berat badan dengan rumus berat := bmi \* (tinggi \* tinggi), karena BMI merupakan hasil pembagian berat badan dengan kuadrat tinggi badan. Hasil perhitungan berat kemudian ditampilkan menggunakan fmt.Printf("berat: %.0f kg\n", berat), yang membulatkan angka ke bilangan bulat tanpa angka desimal. Jadi, program ini membantu menentukan berat badan berdasarkan tinggi dan BMI yang diberikan pengguna.**

**Tugas 3**

**Source code**

|  |
| --- |
| package main  import (      "fmt"      "math"  )  func main() {      var x1, y1, x2, y2, x3, y3 float64      fmt.Scan(&x1, &y1)      fmt.Scan(&x2, &y2)      fmt.Scan(&x3, &y3)      AB := math.Sqrt(math.Pow(x2-x1, 2) + math.Pow(y2-y1, 2))      BC := math.Sqrt(math.Pow(x3-x2, 2) + math.Pow(y3-y2, 2))      CA := math.Sqrt(math.Pow(x1-x3, 2) + math.Pow(y1-y3, 2))      terpanjang := AB      if BC > terpanjang {          terpanjang = BC      }      if CA > terpanjang {          terpanjang = CA      }      fmt.Printf("%.2f\n", terpanjang)  } |

**Screenshoot program : A screen shot of a computer

AI-generated content may be incorrect.**

**Deskripsi program :**

**Program Go ini digunakan untuk menghitung panjang sisi-sisi segitiga yang dibentuk oleh tiga titik koordinat, lalu menentukan sisi yang paling panjang. Pertama, program mendeklarasikan enam variabel bertipe float64 untuk menyimpan koordinat titik A (x1, y1), titik B (x2, y2), dan titik C (x3, y3). Nilai koordinat dibaca dari input menggunakan tiga perintah fmt.Scan. Setelah itu, program menghitung panjang masing-masing sisi segitiga menggunakan rumus jarak antar dua titik dengan teorema Pythagoras. Jarak antara titik A dan B disimpan dalam variabel AB, antara titik B dan C dalam BC, dan antara titik C dan A dalam CA. Perhitungan dilakukan dengan fungsi math.Sqrt untuk akar kuadrat dan math.Pow untuk pangkat dua.**

**Selanjutnya, program membandingkan ketiga panjang sisi tersebut untuk mencari nilai terbesar. Variabel terpanjang awalnya diisi dengan nilai AB, kemudian dibandingkan dengan BC dan CA. Jika salah satunya lebih besar, maka nilainya menggantikan variabel terpanjang. Terakhir, hasil sisi terpanjang ditampilkan menggunakan fmt.Printf dengan format dua angka di belakang koma. Program ini membantu mengidentifikasi panjang sisi terbesar dari sebuah segitiga berdasarkan koordinat kartesius.**