**LAPORAN PRAKTIKUM   
ALGORITMA DAN PEMR OGRAMAN 2**

**Modul 15 Test**

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

Deskripsi dibuat secara otomatis**

**Disusun Oleh :**

**Rasyid Nafsyarie / 2311102011**

**IF-11-06**

**Dosen Pengampu :**

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

Soal 1

A screenshot of a document

Description automatically generated

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  type set[2022]int  //fungsi ini untuk melakukan pengecekan apakah nilainya sudah dalam array  func exist (T set, n int, val int) bool {      for i := 0; i < n; i++ {          if T[i] == val {              return true          }      }      return false  }  //fungsi ini untuk mengisi nilai array  func inputSet (T \*set, n \*int) {      var val int      \*n = 0      for {          fmt.Scan(&val)          if exist(\*T, \*n, val) {              break          }          T[\*n] = val          (\*n)++      }  }  //fungsi ini untuk mencari nilai array yang terduplikat  func findIntersection (T1, T2 set, n, m int, T3 \*set, h \*int) {      \*h = 0      for i := 0; i < n; i++ {          if exist(T2, m, T1[i]) && !exist(\*T3, \*h, T1[i]) {              T3[\*h] = T1[i]              (\*h)++          }      }  }  //fungsi ini untuk mencetak nilai array secara horizontal  func printSet(T set, n int) {      for i := 0; i < n; i++ {          if i > 0 {              fmt.Print(" ")          }          fmt.Print(T[i])      }      fmt.Println()  }  func main () {      var s1\_2311102011, s2, s3 set      var n1, n2, n3 int      inputSet(&s1\_2311102011, &n1)      inputSet(&s2, &n2)      findIntersection(s1\_2311102011, s2,n1,n2, &s3, &n3,)      printSet(s3,n3)  } |

A screenshot of a computer

Description automatically generated

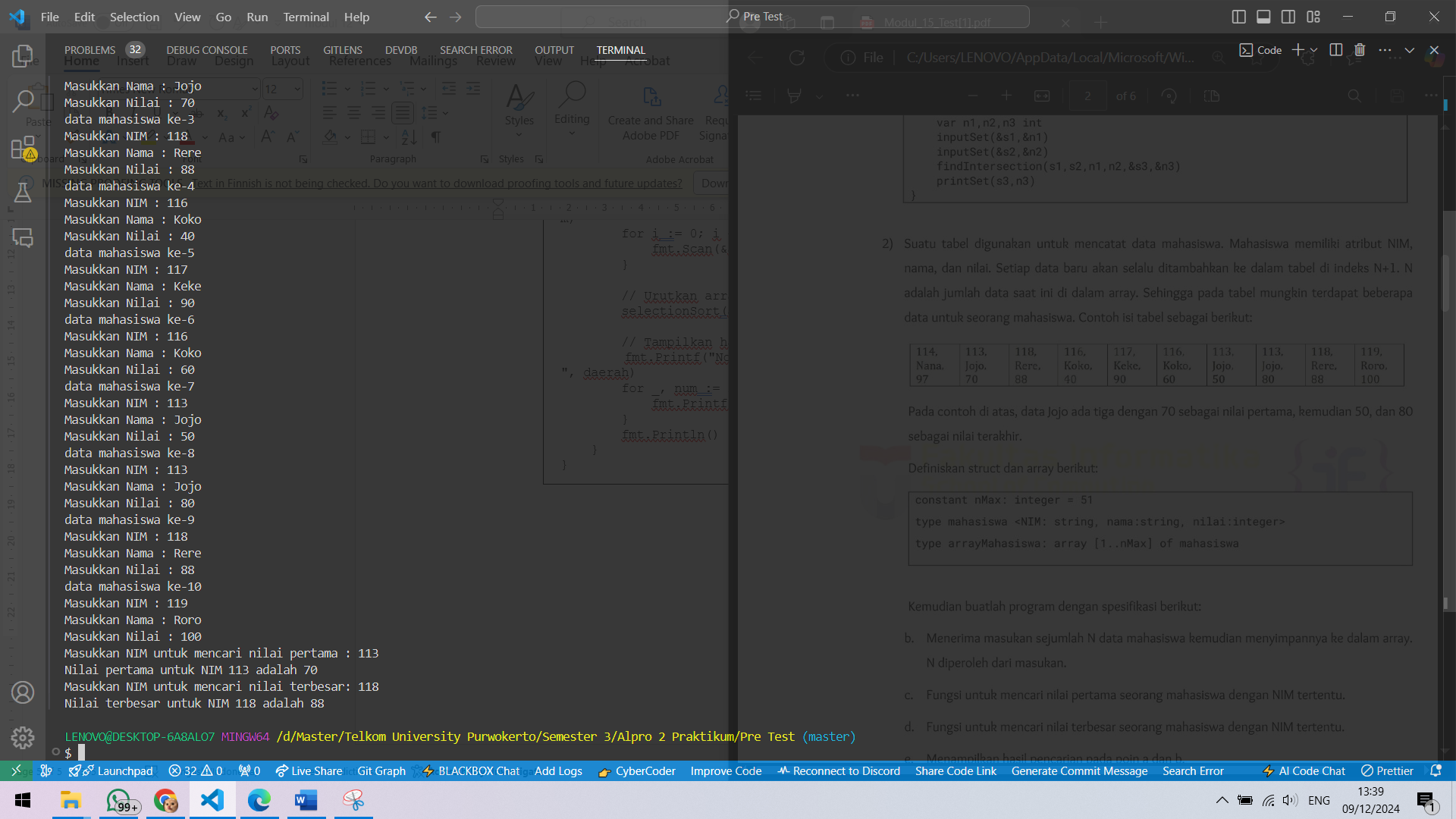
Soal 2

A screenshot of a computer

Description automatically generated

**Sourcecode**

|  |
| --- |
| package main  import "fmt"  const nMax = 51  type mahasiswa struct {      NIM string      nama string      nilai int  }  type arrayMahasiswa [nMax] mahasiswa  //fungsi ini yaitu proses input mahasiswa beserta data diri lainnya  func inputMahasiswa (T \*arrayMahasiswa, N \*int) {      fmt.Print("Masukkan Jumlah data Mahasiswa : ")      fmt.Scan(N)      for i := 0; i < \*N; i++ {          fmt.Printf("data mahasiswa ke-%d\n", i+1)          fmt.Print("Masukkan NIM : ")          fmt.Scan(&T[i].NIM)          fmt.Print("Masukkan Nama : ")          fmt.Scan(&T[i].nama)          fmt.Print("Masukkan Nilai : ")          fmt.Scan(&T[i].nilai)      }  }  //fungsi ini untuk mencari sebuah nilai pertama dari mahasiswa yang sudah diinputkan tadi  func cariNilai (T arrayMahasiswa, N int, nim\_2311102011 string) int {      for i := 0; i < N; i++ {          if T[i].NIM == nim\_2311102011 {              return T[i].nilai          }      }      return -1  }  //kemudian fungsi nilai terbesar ini untuk menemukan nilai dari inputan tadi yang paling besar menggunakan for untuk melakukan pengecekan dan percabangan  func cariNilaiTerbesar(T arrayMahasiswa, N int, nim\_2311102011 string) int{      maxNilai := -1      found := false      for i := 0; i < N; i++ {          if T[i].NIM == nim\_2311102011 {              found = true              if T[i].nilai > maxNilai{                  maxNilai = T[i].nilai              }          }      }      if found {          return maxNilai      }      return -1  }  func main() {      var dataMahasiswa arrayMahasiswa      var jumlahData int      var nim\_2311102011 string      inputMahasiswa(&dataMahasiswa, &jumlahData)      fmt.Print("Masukkan NIM untuk mencari nilai pertama : ")      fmt.Scan(&nim\_2311102011)      nilaiPertama := cariNilai(dataMahasiswa, jumlahData, nim\_2311102011)      if nilaiPertama != -1 {          fmt.Printf("Nilai pertama untuk NIM %s adalah %d\n", nim\_2311102011, nilaiPertama)      } else {          fmt.Printf("Data dengan NIM %s tidak ditemukan\n", nim\_2311102011)      }      fmt.Print("Masukkan NIM untuk mencari nilai terbesar: ")      fmt.Scan(&nim\_2311102011)      nilaiTerbesar := cariNilaiTerbesar(dataMahasiswa, jumlahData, nim\_2311102011)      if nilaiTerbesar != -1 {          fmt.Printf("Nilai terbesar untuk NIM %s adalah %d\n", nim\_2311102011, nilaiTerbesar)      } else {          fmt.Printf("Data dengan NIM %s tidak ditemukan\n", nim\_2311102011)      }  } |



Soal 3

A screenshot of a cell phone

Description automatically generated

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "strings"  )  const nProv = 34  type (      NamaProv [nProv]string      PopProv [nProv]int      TumbuhProv [nProv]float64  )  // Fungsi untuk menginputkan data provinsi  func InputData(nama \*NamaProv, pop \*PopProv, tumbuh \*TumbuhProv) {      for i := 0; i < nProv; i++ {          fmt.Printf("Masukkan nama provinsi ke-%d: ", i+1)          fmt.Scanln(&nama[i])          fmt.Printf("Masukkan populasi provinsi %s: ", nama[i])          fmt.Scanln(&pop[i])          fmt.Printf("Masukkan angka pertumbuhan penduduk provinsi %s: ", nama[i])          fmt.Scanln(&tumbuh[i])      }  }  // Fungsi untuk mencari provinsi dengan pertumbuhan tercepat  func ProvinsiTercepat(tumbuh TumbuhProv) int {      indeks := 0      maks := tumbuh[0]      for i := 1; i < nProv; i++ {          if tumbuh[i] > maks {              maks = tumbuh[i]              indeks = i          }      }      return indeks  }  // Fungsi untuk mencari indeks provinsi berdasarkan nama provinsi yang telah diinputkan  func IndeksProvinsi(nama NamaProv, target string) int {      for i := 0; i < nProv; i++ {          if strings.EqualFold(nama[i], target) {              return i          }      }      return -1  }  // fungsi untuk menampilkan prediksi populasi provinsi  func Prediksi(nama NamaProv, pop PopProv, tumbuh TumbuhProv) {      fmt.Println("Prediksi provinsi dengan pertumbuhan > 2%:")      for i := 0; i < nProv; i++ {          if tumbuh[i] > 0.02 {              prediksi := float64(pop[i]) \* (1 + tumbuh[i])              fmt.Printf("%s: Populasi tahun depan = %.0f\n", nama[i], prediksi)          }      }  }  func main() {      var (          nama\_2311102011   NamaProv          pop    PopProv          tumbuh TumbuhProv          cari   string      )      InputData(&nama\_2311102011, &pop, &tumbuh)      tercepat := ProvinsiTercepat(tumbuh)      fmt.Printf("Provinsi dengan pertumbuhan tercepat: %s\n", nama\_2311102011[tercepat])      fmt.Println("Masukkan nama provinsi yang ingin dicari:")      fmt.Scanln(&cari)        indeks := IndeksProvinsi(nama\_2311102011, cari)      if indeks != -1 {          fmt.Printf("Provinsi %s ditemukan pada indeks %d\n", cari, indeks)      } else {          fmt.Printf("Provinsi %s tidak ditemukan\n", cari)      }      Prediksi(nama\_2311102011, pop,tumbuh)  } |

A screenshot of a computer

Description automatically generated

Soal 4

A screenshot of a screenshot of a document

Description automatically generated

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"      "sort"  )  // Fungsi untuk menghitung median, ketika jumlah datanya ganjil maka median adalah nilai tengahnya, namun ketika jumlah genap median adalah rata rata dari nilai tengah  func median(data []int) float64 {      n := len(data)      if n == 0 {          return 0      }      if n%2 == 1 {          return float64(data[n/2])      }      return float64(data[(n/2)-1]+data[n/2]) / 2  }  func main() {      const sentinel = -5313541      var data\_2311102011 []int      var input int      fmt.Println("Masukkan bilangan (akhiri dengan -5313541):")      for {          fmt.Scan(&input)          if input == sentinel {              break          }          if input == 0 {              sort.Ints(data\_2311102011)              fmt.Printf("Median: %.2f\n", median(data\_2311102011))          } else {              data\_2311102011 = append(data\_2311102011, input)          }      }  } |

A screenshot of a computer

Description automatically generated

Soal 5

A paper with text and numbers

Description automatically generated

**Sourcecode**

|  |
| --- |
| package main  import (      "fmt"  )  const NMAX = 1000000  type Partai struct {      nama  int      suara int  }  type TabPartai [NMAX]Partai  // Fungsi untuk mencari indeks partai berdasarkan nama  func posisi(t TabPartai, n int, nama int) int {      for i := 0; i < n; i++ {          if t[i].nama == nama {              return i          }      }      return -1  }  // Fungsi untuk mengurutkan array partai secara descending berdasarkan jumlah suara, untuk memindahkan elemen yang lebih kecil menjadi ke kanan  func insertionSortDescending(t \*TabPartai, n int) {      for i := 1; i < n; i++ {          key := t[i]          j := i - 1          for j >= 0 && t[j].suara < key.suara {              t[j+1] = t[j]              j--          }          t[j+1] = key      }  }  func main() {      var t\_2311102011 TabPartai      var n int      var input int      n = 0      for {          fmt.Scan(&input)          if input == -1 {              break          }          pos := posisi(t\_2311102011, n, input)          if pos == -1 {              t\_2311102011[n].nama = input              t\_2311102011[n].suara = 1              n++          } else {              t\_2311102011[pos].suara++          }      }      insertionSortDescending(&t\_2311102011, n)      for i := 0; i < n; i++ {          fmt.Printf("%d(%d) ", t\_2311102011[i].nama, t\_2311102011[i].suara)      }      fmt.Println()  } |

A screenshot of a computer

Description automatically generated