

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**UJIAN AKHIR ALPRO**



**Disusun Oleh :**

**Ariiq Radhitya Pradana / 2311102260**

**IF 11 06**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. JAWABAN UJIAN AKHIR ALPRO

### 1. Irisan Himpunan

```
package main

import "fmt"

// Fungsi untuk mengecek apakah elemen sudah ada di
dalam set
func exists_2311102260(set []int, n int) bool {
    for _, v := range set {
        if v == n {
            return true
        }
    }
    return false
}

// Fungsi untuk membaca input sebagai himpunan tanpa
duplikasi
func inputSet_2311102260() []int {
    var set []int
    var n int

    fmt.Println("Masukkan elemen (akhiri dengan bilangan
duplikat atau nilai yang sudah dimasukkan sebelumnya):")
    for {
        fmt.Scan(&n)
        if exists_2311102260(set, n) {
            break
        }
        set = append(set, n)
    }
    return set
}

// Fungsi untuk mencari irisan antara dua himpunan
func findIntersection_2311102260(set1, set2 []int) []int
{
    var intersection []int
    for _, v := range set1 {
        if exists_2311102260(set2, v) &&
!exists_2311102260(intersection, v) {
            intersection = append(intersection, v)
        }
    }
    return intersection
}

// Fungsi untuk mencetak himpunan
func printSet_2311102260(set []int) {
    for _, v := range set {
```

```

        fmt.Print(v, " ")
    }
    fmt.Println()
}

func main() {
    fmt.Println("Masukkan elemen untuk himpunan pertama:")
    set1 := inputSet_2311102260()

    fmt.Println("Masukkan elemen untuk himpunan kedua:")
    set2 := inputSet_2311102260()

    intersection := findIntersection_2311102260(set1, set2)

    fmt.Println("Irisan dari kedua himpunan adalah:")
    printSet_2311102260(intersection)
}

```

### Screenshoot Output

```

Masukkan elemen untuk himpunan pertama:
Masukkan elemen (akhiri dengan bilangan duplikat atau nilai yang sudah
dimasukkan sebelumnya):
1 1 1 1
Masukkan elemen untuk himpunan kedua:
Masukkan elemen (akhiri dengan bilangan duplikat atau nilai yang sudah
dimasukkan sebelumnya):
Irisan dari kedua himpunan adalah:
1

```

## 2. Data Mahasiswa

```
package main

import "fmt"

const nMax = 51

type mahasiswa struct {
    NIM    string
    nama   string
    nilai  int
}
type arrayMahasiswa [nMax]mahasiswa

// Fungsi untuk menerima masukan sejumlah N data
mahasiswa
func inputMahasiswa_2311102260(n int, arr
*arrayMahasiswa) {
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan data mahasiswa ke-%d:\n",
i+1)

        fmt.Print("NIM: ")
        fmt.Scan(&arr[i].NIM)
        fmt.Print("Nama: ")
        fmt.Scan(&arr[i].nama)
        fmt.Print("Nilai: ")
        fmt.Scan(&arr[i].nilai)
    }
}

// Fungsi untuk mencari nilai rata-rata semua mahasiswa
dengan NIM tertentu
func rataRataNilai_2311102260(n int, arr arrayMahasiswa,
targetNIM string) float64 {
    var totalNilai int
    var jumlah int

    for i := 0; i < n; i++ {
        if arr[i].NIM == targetNIM {
            totalNilai += arr[i].nilai
            jumlah++
        }
    }

    if jumlah == 0 {
        return 0 // Jika tidak ada data untuk NIM
        tertentu
    }
    return float64(totalNilai) / float64(jumlah)
}

// Fungsi untuk mencari nilai terbesar mahasiswa dengan
NIM tertentu
```

```

func nilaiTertinggi_2311102260(n int, arr
arrayMahasiswa, targetNIM string) int {
    var maxNilai = -1

    for i := 0; i < n; i++ {
        if arr[i].NIM == targetNIM {
            if arr[i].nilai > maxNilai {
                maxNilai = arr[i].nilai
            }
        }
    }

    return maxNilai
}

func main() {
    var n int
    var arr arrayMahasiswa
    var targetNIM string

    fmt.Print("Masukkan jumlah data mahasiswa (N): ")
    fmt.Scan(&n)

    inputMahasiswa_2311102260(n, &arr)

    fmt.Print("Masukkan NIM untuk pencarian: ")
    fmt.Scan(&targetNIM)

    rataRata := rataRataNilai_2311102260(n, arr,
targetNIM)
    if rataRata == 0 {
        fmt.Printf("Tidak ditemukan data dengan NIM
%s\n", targetNIM)
    } else {
        fmt.Printf("Nilai rata-rata untuk NIM %s:
%.2f\n", targetNIM, rataRata)
    }

    nilaiMax := nilaiTertinggi_2311102260(n, arr,
targetNIM)
    if nilaiMax == -1 {
        fmt.Printf("Tidak ditemukan data dengan NIM
%s\n", targetNIM)
    } else {
        fmt.Printf("Nilai tertinggi untuk NIM %s: %d\n",
targetNIM, nilaiMax)
    }
}

```

### Screenshoot Output

```
Masukkan jumlah data mahasiswa (N): 2
Masukkan data mahasiswa ke-1:
NIM: 2311102260
Nama: Ariiq
Nilai: 90
Masukkan data mahasiswa ke-2:
NIM: 2311102270
Nama: Vyn
Nilai: 80
Masukkan NIM untuk pencarian: 2311102260
Nilai rata-rata untuk NIM 2311102260: 90.00
Nilai tertinggi untuk NIM 2311102260: 90
```

### 3. Data Provinsi

```
package main

import (
    "fmt"
)

const nProv = 34

// Struktur data untuk menyimpan nama, populasi, dan
// angka pertumbuhan provinsi
type Provinsi struct {
    Nama      string
    Populasi  int
    Pertumbuhan float64
}

// Fungsi untuk menginput data provinsi
func inputData_2311102260(provinsi []Provinsi) {
    for i := 0; i < nProv; i++ {
        fmt.Printf("Masukkan data provinsi ke-%d:\n",
i+1)

        fmt.Print("Nama Provinsi: ")
        fmt.Scan(&provinsi[i].Nama)
        fmt.Print("Jumlah Populasi: ")
        fmt.Scan(&provinsi[i].Populasi)
        fmt.Print("Angka Pertumbuhan (%): ")
        fmt.Scan(&provinsi[i].Pertumbuhan)
    }
}

// Fungsi untuk mencari provinsi dengan angka
// pertumbuhan tercepat
func provinsiTercepat_2311102260(provinsi []Provinsi)
string {
    indexTercepat := 0
    for i := 1; i < nProv; i++ {
        if provinsi[i].Pertumbuhan >
provinsi[indexTercepat].Pertumbuhan {
            indexTercepat = i
        }
    }
    return provinsi[indexTercepat].Nama
}

// Fungsi untuk mencari indeks provinsi berdasarkan nama
func indeksProvinsi_2311102260(provinsi []Provinsi, nama
string) int {
    for i := 0; i < nProv; i++ {
        if provinsi[i].Nama == nama {
            return i
        }
    }
}
```

```

    }
    return -1 // Jika provinsi tidak ditemukan
}

// Fungsi untuk menampilkan prediksi populasi
berdasarkan pertumbuhan > 2%
func prediksiPopulasi_2311102260(provinsi []Provinsi) {
    fmt.Println("Prediksi populasi untuk provinsi dengan
    pertumbuhan > 2%:")
    for i := 0; i < nProv; i++ {
        if provinsi[i].Pertumbuhan > 2 {
            prediksi :=
            float64(provinsi[i].Populasi) * (1 +
            provinsi[i].Pertumbuhan/100)
            fmt.Printf("%s: Populasi saat ini = %d,
            Prediksi populasi = %.2f\n", provinsi[i].Nama,
            provinsi[i].Populasi, prediksi)
        }
    }
}

func main() {
    var provinsi [nProv]Provinsi
    var namaCari string

    // Input data
    fmt.Println("Masukkan data untuk 34 provinsi:")
    inputData_2311102260(provinsi[:])

    // Provinsi dengan pertumbuhan tercepat
    namaTercepat :=
    provinsiTercepat_2311102260(provinsi[:])
    fmt.Printf("Provinsi dengan angka pertumbuhan
    tercepat: %s\n", namaTercepat)

    // Cari provinsi berdasarkan nama
    fmt.Print("Masukkan nama provinsi yang ingin dicari:
    ")
    fmt.Scan(&namaCari)
    index := indeksProvinsi_2311102260(provinsi[:],
    namaCari)
    if index == -1 {
        fmt.Printf("Provinsi dengan nama '%s' tidak
        ditemukan.\n", namaCari)
    } else {
        fmt.Printf("Provinsi %s memiliki populasi %d
        dan pertumbuhan %.2f%%\n", provinsi[index].Nama,
        provinsi[index].Populasi, provinsi[index].Pertumbuhan)
    }

    // Prediksi populasi
    prediksiPopulasi_2311102260(provinsi[:])
}

```



```
}
```

### Screenshoot Output

Masukkan data untuk 34 provinsi:

Masukkan data provinsi ke-1:

Nama Provinsi: jakarta

Jumlah Populasi: 60

Angka Pertumbuhan (%): 70%

Masukkan data provinsi ke-2:

Nama Provinsi: Bandung

Jumlah Populasi: 60

Angka Pertumbuhan (%): 80%

Masukkan data provinsi ke-3:

Nama Provinsi: |

#### 4.Kompetisi pemrograman

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

const NMAX = 1000000 // Batas maksimum array
type arrInt [NMAX]int

// Fungsi untuk mengurutkan array menggunakan algoritma
selection sort
func sorting_2311102260(T *arrInt, n int) {
    for i := 0; i < n-1; i++ {
        minIdx := i
        for j := i + 1; j < n; j++ {
            if T[j] < T[minIdx] {
                minIdx = j
            }
        }
        // Menukar elemen
        T[i], T[minIdx] = T[minIdx], T[i]
    }
}

// Fungsi untuk menghitung median dari array yang sudah
diurutkan
func median_2311102260(T arrInt, n int) float64 {
    if n%2 == 0 {
        return float64(T[n/2-1]+T[n/2]) / 2.0
    }
    return float64(T[n/2])
}

func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan data (pisahkan dengan spasi,
akhiri dengan -5313541):")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)
    strNums := strings.Split(input, " ")

    var T arrInt
    n := 0
    step := 1

    fmt.Println()
    for _, str := range strNums {
```

```

num, _ := strconv.Atoi(str)
if num == 0 {
    // Jika input adalah 0, cetak hasil saat ini
    fmt.Printf("Sampai bilangan 0 yang ke-%d,
data terbaca adalah ", step)
    step++
    fmt.Print(T[:n], ", setelah tersusun: ")
    sorting_2311102260(&T, n)
    fmt.Print(T[:n])
    fmt.Printf(", maka median saat itu adalah
%.1f\n", median_2311102260(T, n))
} else if num == -5313541 {
    break // Penanda akhir input
} else {
    // Tambahkan angka ke array
    T[n] = num
    n++
}
}
}

```

### Screenshoot Output

Masukkan data (pisahkan dengan spasi, akhiri dengan -5313541):

7 23 11 0 5 19 2 29 3 13 17 0 -5313541

Sampai bilangan 0 yang ke-1, data terbaca adalah [7 23 11], setelah tersusun: [7 11 23], maka median saat itu adalah 11.0

Sampai bilangan 0 yang ke-2, data terbaca adalah [7 11 23 5 19 2 29 3 13 17], setelah tersusun: [2 3 5 7 11 13 17 19 23 29], maka median saat itu adalah 12.0

## 5.Perolehan suara

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "sort"
    "strconv"
    "strings"
)

const NMAX = 1000000

// Struct untuk menyimpan nama partai dan jumlah suara
type Partai struct {
    nama    int
    suara   int
}

// Fungsi untuk mencari indeks partai di dalam array
func posisi_2311102260(tabPartai []Partai, n int, nama
int) int {
    for i := 0; i < n; i++ {
        if tabPartai[i].nama == nama {
            return i
        }
    }
    return -1
}

func main() {
    var partai []Partai // Array untuk menyimpan data
partai
    var n int           // Jumlah partai yang ditemukan

    // Membaca input pengguna
    fmt.Println("Masukkan data suara (akhiri dengan -
1):")
    scanner := bufio.NewScanner(os.Stdin)
    scanner.Scan()
    input := scanner.Text()

    // Memproses input
    data := strings.Split(input, " ")
    for _, val := range data {
        suara, err := strconv.Atoi(val)
        if err != nil {
            fmt.Println("Input tidak valid:", val)
            return
        }
        if suara == -1 {
```

```

        break
    }
    idx := posisi_2311102260(partai, n, suara)
    if idx == -1 {
        // Jika partai belum ada di array, tambahkan
partai baru
        partai = append(partai, Partai{nama: suara,
suara: 1})
        n++
    } else {
        // Jika partai sudah ada, tambahkan jumlah
suaranya
        partai[idx].suara++
    }
}

// Sort array berdasarkan suara (descending) dan
nama (ascending jika suara sama)
sort.Slice(partai, func(i, j int) bool {
    if partai[i].suara == partai[j].suara {
        return partai[i].nama < partai[j].nama
    }
    return partai[i].suara > partai[j].suara
}))

// Output hasil
fmt.Println("Hasil perolehan suara:")
for _, p := range partai {
    fmt.Printf("%d(%d) ", p.nama, p.suara)
}
fmt.Println()
}

```

### Screenshoot Output

```

Masukkan data suara (akhiri dengan -1):
5 1 1 1 1 1 1 1 3 3 3 3 3 2 2 5 5 5 5 5 4 3 2 2 2 2 -1
Hasil perolehan suara:
1(7) 2(6) 3(6) 5(6) 4(1)

```