

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN**

UJIAN PRAKTEK AKHIR



Disusun oleh:

Martin Simbolon

NIM

2311102269

Dosen Pengampu:

Nama Dosen

**PROGRAM STUDI TEKNIK INFORMATIKAFAKULTAS
INFORMATIKA**

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO 2024**

1. Sourcecode

```
// Martin Simbolon 2311102269
package main

import (
    "fmt"
)

func main() {
    // Input dua baris himpunan
    var set1, set2 []int

    fmt.Println("Masukkan elemen baris pertama (akhiri dengan angka duplikat):")
    set1 = readSet()

    fmt.Println("Masukkan elemen baris kedua (akhiri dengan angka duplikat):")
    set2 = readSet()

    // Temukan irisan dari kedua himpunan
    intersection := findIntersection(set1, set2)

    // Cetak hasil
    fmt.Println("Irisan dari kedua himpunan:", intersection)
}

func readSet() []int {
    var input int
    set := make(map[int]bool) // Untuk memastikan tidak ada duplikat
    var result []int

    for {
        fmt.Scan(&input)
        if set[input] {
            break // Berhenti jika elemen sudah ada dalam himpunan
        }
        set[input] = true
        result = append(result, input)
    }

    return result
}

func findIntersection(set1, set2 []int) []int {
    setMap := make(map[int]bool)
    for _, num := range set1 {
        setMap[num] = true
    }
}
```

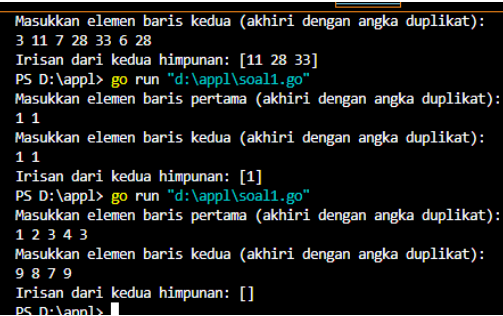
```

var intersection []int
for _, num := range set2 {
    if setMap[num] {
        intersection = append(intersection, num)
    }
}

return intersection
}

```

Screenshoot Output



```

Masukkan elemen baris kedua (akhiri dengan angka duplikat):
3 11 7 28 33 6 28
Irisan dari kedua himpunan: [11 28 33]
PS D:\appl> go run "d:\appl\soal1.go"
Masukkan elemen baris pertama (akhiri dengan angka duplikat):
1 1
Masukkan elemen baris kedua (akhiri dengan angka duplikat):
1 1
Irisan dari kedua himpunan: [1]
PS D:\appl> go run "d:\appl\soal1.go"
Masukkan elemen baris pertama (akhiri dengan angka duplikat):
1 2 3 4 3
Masukkan elemen baris kedua (akhiri dengan angka duplikat):
9 8 7 9
Irisan dari kedua himpunan: []
PS D:\appl>

```

2. Sourcecode

```

// Martin Simbolon 2311102269

package main

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

type Mahasiswa struct {
    NIM string
    Nama string
    Nilai []int
}

func main() {
    // Menerima masukan jumlah data mahasiswa
    fmt.Print("Masukkan jumlah data mahasiswa: ")
    reader := bufio.NewReader(os.Stdin)
    numInputStr, _ := reader.ReadString('\n')

```

```

N, _ := strconv.Atoi(strings.TrimSpace(numInputStr))

// Buat slice untuk menyimpan data mahasiswa
dataMahasiswa := make([]Mahasiswa, 0, N)

// Menerima data mahasiswa
for i := 0; i < N; i++ {
    fmt.Printf("Masukkan NIM mahasiswa %d: ", i+1)
    nimInput, _ := reader.ReadString('\n')
    nim := strings.TrimSpace(nimInput)

    fmt.Printf("Masukkan nama mahasiswa %d: ", i+1)
    namaInput, _ := reader.ReadString('\n')
    nama := strings.TrimSpace(namaInput)

    fmt.Printf("Masukkan %d nilai mahasiswa %d (pisahkan dengan spasi): ", N, i+1)
    nilaiInput, _ := reader.ReadString('\n')
    nilaiStr := strings.Split(strings.TrimSpace(nilaiInput), " ")
    nilai := make([]int, 0, N)
    for _, v := range nilaiStr {
        v, _ := strconv.Atoi(v)
        nilai = append(nilai, v)
    }

    dataMahasiswa = append(dataMahasiswa, Mahasiswa{
        NIM: nim,
        Nama: nama,
        Nilai: nilai,
    })
}

// Mencari mahasiswa dengan nilai tertinggi di awal
mahasiswaAwal := findHighestFirstValue(dataMahasiswa)
fmt.Printf("Mahasiswa dengan nilai tertinggi di awal: %s (%s)\n", mahasiswaAwal.Nama,
mahasiswaAwal.NIM)

// Mencari mahasiswa dengan nilai tertinggi di akhir
mahasiswaAkhir := findHighestLastValue(dataMahasiswa)
fmt.Printf("Mahasiswa dengan nilai tertinggi di akhir: %s (%s)\n", mahasiswaAkhir.Nama,
mahasiswaAkhir.NIM)
}

func findHighestFirstValue(data []Mahasiswa) Mahasiswa {
    var highest Mahasiswa
    for _, m := range data {
        if len(m.Nilai) > 0 && (len(highest.Nilai) == 0 || m.Nilai[0] > highest.Nilai[0]) {
            highest = m
        }
    }
    return highest
}

```

```

func findHighestLastValue(data []Mahasiswa) Mahasiswa {
    var highest Mahasiswa
    for _, m := range data {
        if len(m.Nilai) > 0 && (len(highest.Nilai) == 0 || m.Nilai[len(m.Nilai)-1] >
highest.Nilai[len(highest.Nilai)-1]) {
            highest = m
        }
    }
    return highest
}

```

Screenshoot Output

```

PS D:\appl> go run "d:\appl\soal2.go"
Masukkan jumlah data mahasiswa: 3
Masukkan NIM mahasiswa 1: 2311102210
Masukkan nama mahasiswa 1: Timoty
Masukkan 3 nilai mahasiswa 1 (pisahkan dengan spasi): 89 90 99
Masukkan NIM mahasiswa 2: 2311102230
Masukkan nama mahasiswa 2: Kalimasada
Masukkan 3 nilai mahasiswa 2 (pisahkan dengan spasi): 90 95 100
Masukkan NIM mahasiswa 3: 2311102269
Masukkan nama mahasiswa 3: 2311102269
Masukkan 3 nilai mahasiswa 3 (pisahkan dengan spasi): 80 90 95
Mahasiswa dengan nilai tertinggi di awal: Kalimasada (2311102230)
Mahasiswa dengan nilai tertinggi di akhir: Kalimasada (2311102230)

```

3. Sourcecode

```

// Martin Simbolon 2311102269

package main

import (
    "fmt"
    "strings"
)

const maxProv = 34

func inputData() (int, [maxProv]string, [maxProv]int, [maxProv]float64) {
    var n int
    var namaProv [maxProv]string
    var popProv [maxProv]int
    var tumbuhProv [maxProv]float64

    fmt.Println("Masukkan jumlah provinsi (maksimal 34):")
    fmt.Scan(&n)
    if n > maxProv {
        fmt.Printf("Jumlah provinsi melebihi batas maksimum (%d). Menggunakan nilai
maksimum.\n", maxProv)
        n = maxProv
    } else if n <= 0 {

```

```

    fmt.Println("Jumlah provinsi harus lebih dari 0.")
    return 0, namaProv, popProv, tumbuhProv
}

for i := 0; i < n; i++ {
    fmt.Printf("Masukkan nama provinsi ke-%d: ", i+1)
    fmt.Scan(&namaProv[i])
    namaProv[i] = strings.TrimSpace(namaProv[i]) // Menghapus spasi berlebih

    fmt.Printf("Masukkan populasi provinsi ke-%d (juta): ", i+1)
    fmt.Scan(&popProv[i])
    if popProv[i] <= 0 {
        fmt.Println("Populasi harus lebih dari 0. Masukkan ulang.")
        i-- // Ulangi input untuk provinsi ini
        continue
    }

    fmt.Printf("Masukkan pertumbuhan penduduk provinsi ke-%d (%%): ", i+1)
    fmt.Scan(&tumbuhProv[i])
    if tumbuhProv[i] < 0 {
        fmt.Println("Pertumbuhan tidak boleh negatif. Masukkan ulang.")
        i-- // Ulangi input untuk provinsi ini
        continue
    }
}

return n, namaProv, popProv, tumbuhProv
}

func provinsiTerbesar(popProv [maxProv]int, n int) int {
    maxIndex := 0
    for i := 1; i < n; i++ {
        if popProv[i] > popProv[maxIndex] {
            maxIndex = i
        }
    }
    return maxIndex
}

func prediksiPenduduk(popProv [maxProv]int, tumbuhProv [maxProv]float64, n int) [maxProv]int {
    var prediksi [maxProv]int
    for i := 0; i < n; i++ {
        prediksi[i] = popProv[i] + int(float64(popProv[i])*tumbuhProv[i]/100)
    }
    return prediksi
}

func provinsiPertumbuhanTinggi(tumbuhProv [maxProv]float64, n int) []int {
    var result []int
    for i := 0; i < n; i++ {
        if tumbuhProv[i] > 2.0 {

```

```

        result = append(result, i)
    }
}
return result
}

func main() {
    n, namaProv, popProv, tumbuhProv := inputData()
    if n == 0 {
        return
    }

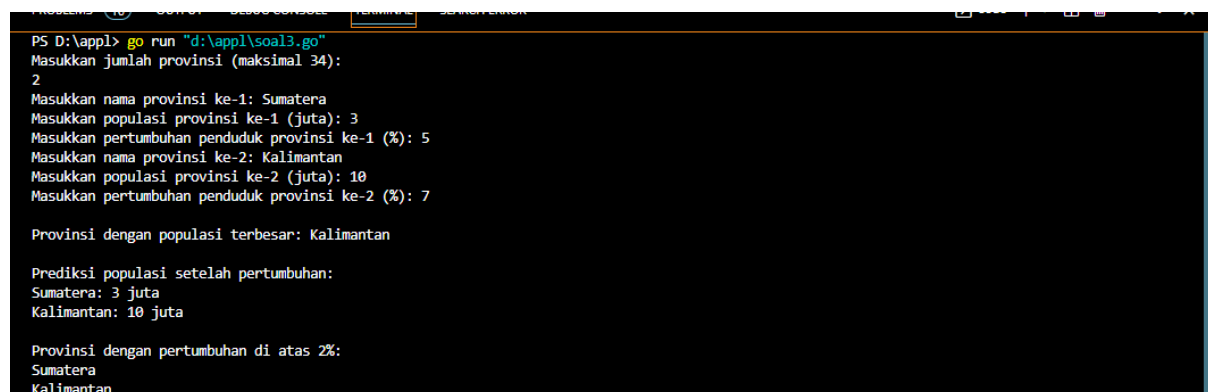
    // Provinsi dengan populasi terbesar
    maxIndex := provinsiTerbesar(popProv, n)
    fmt.Printf("\nProvinsi dengan populasi terbesar: %s\n", namaProv[maxIndex])

    // Prediksi populasi setelah pertumbuhan
    prediksi := prediksiPenduduk(popProv, tumbuhProv, n)
    fmt.Println("\nPrediksi populasi setelah pertumbuhan:")
    for i := 0; i < n; i++ {
        fmt.Printf("%s: %d juta\n", namaProv[i], prediksi[i])
    }

    // Provinsi dengan pertumbuhan di atas 2%
    tinggiPertumbuhan := provinsiPertumbuhanTinggi(tumbuhProv, n)
    fmt.Println("\nProvinsi dengan pertumbuhan di atas 2%:")
    if len(tinggiPertumbuhan) == 0 {
        fmt.Println("Tidak ada provinsi dengan pertumbuhan di atas 2%.")
    } else {
        for _, index := range tinggiPertumbuhan {
            fmt.Printf("%s\n", namaProv[index])
        }
    }
}

```

Screenshoot Output



```

PS D:\appl> go run "d:\appl\soal3.go"
Masukkan jumlah provinsi (maksimal 34):
2
Masukkan nama provinsi ke-1: Sumatera
Masukkan populasi provinsi ke-1 (juta): 3
Masukkan pertumbuhan penduduk provinsi ke-1 (%): 5
Masukkan nama provinsi ke-2: Kalimantan
Masukkan populasi provinsi ke-2 (juta): 10
Masukkan pertumbuhan penduduk provinsi ke-2 (%): 7

Provinsi dengan populasi terbesar: Kalimantan

Prediksi populasi setelah pertumbuhan:
Sumatera: 3 juta
Kalimantan: 10 juta

Provinsi dengan pertumbuhan di atas 2%:
Sumatera
Kalimantan

```

4. Soucecode

```
// 2311102269 Martin Simbolon

package main

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

func main() {
    scanner := bufio.NewScanner(os.Stdin)
    var numbers []int

    fmt.Println("Masukkan angka satu per satu. Ketik '-5313541' untuk berhenti:")

    for {
        fmt.Print("Masukkan angka: ")
        if scanner.Scan() {
            input := scanner.Text()

            // Konversi input ke integer
            number, err := strconv.Atoi(input)
            if err != nil {
                fmt.Println("Masukkan angka yang valid!")
                continue
            }

            // Jika input adalah -5313541, hentikan masukan
            if number == -5313541 {
                break
            }

            // Tambahkan angka ke daftar
            numbers = append(numbers, number)

            // Hitung median
            median := calculateMedian(numbers)
            fmt.Printf("Median saat ini: %.2f\n", median)
        } else {
            fmt.Println("Terjadi kesalahan input. Silakan coba lagi.")
            break
        }
    }
}
```



```

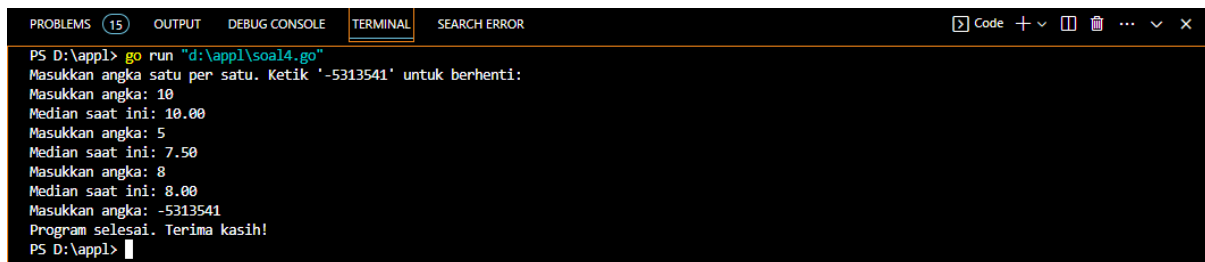
    }
    fmt.Println("Program selesai. Terima kasih!")
}

// Fungsi untuk menghitung median
func calculateMedian(numbers []int) float64 {
    sort.Ints(numbers) // Urutkan data

    n := len(numbers)
    if n%2 == 1 {
        // Jika jumlah data ganjil, median adalah elemen tengah
        return float64(numbers[n/2])
    }
    // Jika jumlah data genap, median adalah rata-rata dua elemen tengah
    return float64(numbers[n/2-1]+numbers[n/2]) / 2
}

```

Screenshoot Output



The screenshot shows a terminal window with the following output:

```

PS D:\appl> go run "d:\appl\soal4.go"
Masukkan angka satu per satu. Ketik '-5313541' untuk berhenti:
Masukkan angka: 10
Median saat ini: 10.00
Masukkan angka: 5
Median saat ini: 7.50
Masukkan angka: 8
Median saat ini: 8.00
Masukkan angka: -5313541
Program selesai. Terima kasih!
PS D:\appl>

```

5. Sourcecode

```
package main

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

// Struct untuk menyimpan partai dan suaranya
type Partai struct {
    ID    int
    Suara int
}

func main() {
    reader := bufio.NewReader(os.Stdin)
    fmt.Println("Masukkan data (akhiri dengan -1):")
    input, _ := reader.ReadString('\n')
    input = strings.TrimSpace(input)

    // Memproses input
    data := strings.Split(input, " ")
    counter := make(map[int]int)

    // Menghitung suara per partai
    for _, s := range data {
        num, _ := strconv.Atoi(s)
        if num == -1 {
            break
        }
        counter[num]++
    }

    // Memasukkan data ke dalam slice untuk diurutkan
    var partaiList []Partai
    for id, suara := range counter {
        partaiList = append(partaiList, Partai{ID: id, Suara: suara})
    }

    // Mengurutkan berdasarkan jumlah suara (descending), lalu ID partai (ascending)
    sort.Slice(partaiList, func(i, j int) bool {
        if partaiList[i].Suara == partaiList[j].Suara {
```

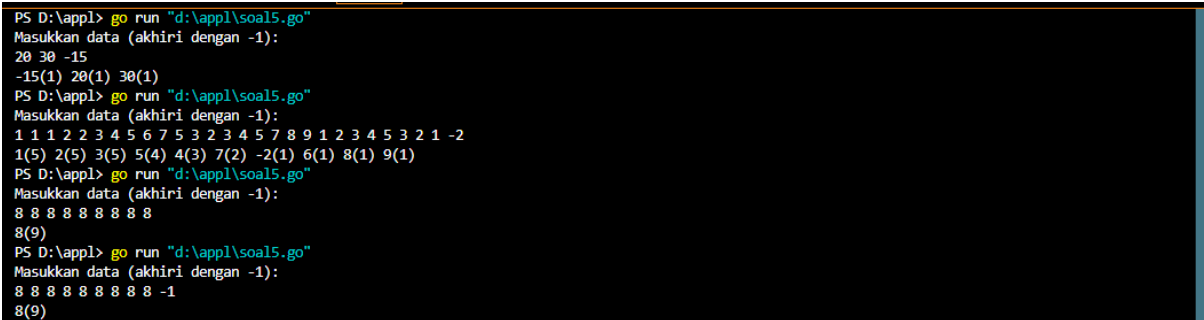
```

        return partaiList[i].ID < partaiList[j].ID
    }
    return partaiList[i].Suara > partaiList[j].Suara
})

// Menampilkan output
for _, p := range partaiList {
    fmt.Printf("%d(%d) ", p.ID, p.Suara)
}
}

```

Screenshot Output



```

PS D:\appl> go run "d:\appl\soal5.go"
Masukkan data (akhiri dengan -1):
20 30 -15
-15(1) 20(1) 30(1)
PS D:\appl> go run "d:\appl\soal5.go"
Masukkan data (akhiri dengan -1):
1 1 1 2 2 3 4 5 6 7 5 3 2 3 4 5 7 8 9 1 2 3 4 5 3 2 1 -2
1(5) 2(5) 3(5) 5(4) 4(3) 7(2) -2(1) 6(1) 8(1) 9(1)
PS D:\appl> go run "d:\appl\soal5.go"
Masukkan data (akhiri dengan -1):
8 8 8 8 8 8 8 8
8(9)
PS D:\appl> go run "d:\appl\soal5.go"
Masukkan data (akhiri dengan -1):
8 8 8 8 8 8 8 8 -1
8(9)

```