

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2  
UJIAN AKHIR ALPRO 2**



**Disusun Oleh :**

**M. Haidar Akhbiyani / 2311102276**

**S1-IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO**

1.

```
2. package main
3.
4. import (
5.     "fmt"
6. )
7.
8. type set [2022]int
9.
10. // Memeriksa apakah nilai sudah ada di array
11. func exist(T set, n int, val int) bool {
12.     for i := 0; i < n; i++ {
13.         if T[i] == val {
14.             return true
15.         }
16.     }
17.     return false
18. }
19.
20. // Memasukkan nilai ke dalam array himpunan
21. func inputSet(T *set, n *int) {
22.     var val int
23.     *n = 0 // Inisialisasi jumlah elemen
24.     fmt.Println("Masukkan elemen himpunan (berhenti jika duplikat
        ditemukan):")
25.     for {
26.         fmt.Scan(&val)
27.         if exist(*T, *n, val) { // Jika duplikat ditemukan,
            berhenti
28.             fmt.Println("Duplikat ditemukan, input dihentikan.")
29.             break
30.         }
31.         T[*n] = val
32.         *n++
33.     }
34. }
35.
36. // Mencari irisan dua himpunan
37. func findIntersection(T1, T2 set, n1, n2 int, T3 *set, n3 *int) {
38.     *n3 = 0 // Inisialisasi jumlah elemen irisan
39.     for i := 0; i < n1; i++ {
40.         if exist(T2, n2, T1[i]) && !exist(*T3, *n3, T1[i]) {
41.             T3[*n3] = T1[i]
42.             (*n3)++
43.         }
44.     }
45. }
```

```

44.     }
45.}
46.
47.// Menampilkan elemen-elemen dalam himpunan
48.func printSet(T set, n int) {
49.    for i := 0; i < n; i++ {
50.        fmt.Print(T[i], " ")
51.    }
52.    fmt.Println()
53.}
54.
55.func main() {
56.    var s1, s2, s3_2311102276 set // Variabel s3 diberi nama
    sesuai NIM
57.    var n1, n2, n3 int
58.
59.    // Input himpunan pertama
60.    fmt.Println("Input himpunan pertama:")
61.    inputSet(&s1, &n1)
62.
63.    // Input himpunan kedua
64.    fmt.Println("Input himpunan kedua:")
65.    inputSet(&s2, &n2)
66.
67.    // Cari irisan
68.    findIntersection(s1, s2, n1, n2, &s3_2311102276, &n3)
69.
70.    // Cetak hasil irisan
71.    fmt.Println("Irisan himpunan:")
72.    printSet(s3_2311102276, n3)
73.}
74.
75.

```

Output

```

PS C:\Users\Lenovo> go run "e:\Ujian Alpro\Ujian 1.go"
Input himpunan pertama:
Masukkan elemen himpunan (berhenti jika duplikat ditemukan):
11 28 33 64 28
Duplikat ditemukan, input dihentikan.
Input himpunan kedua:
Masukkan elemen himpunan (berhenti jika duplikat ditemukan):
2 5
6 2
Duplikat ditemukan, input dihentikan.
Irisan himpunan:

```

2.

```
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
}

```

#### Output

```

PS C:\Users\Lenovo> go run "e:\Ujian Alpro\Ujian 2.go"
Masukkan jumlah data mahasiswa: 3
Masukkan NIM mahasiswa 1: 2311102276
Masukkan nama mahasiswa 1: Haidar
Masukkan 3 nilai mahasiswa 1 (pisahkan dengan spasi): 99 100 98
Masukkan NIM mahasiswa 2: 23111445
Masukkan nama mahasiswa 2: Ucup
Masukkan 3 nilai mahasiswa 2 (pisahkan dengan spasi): 50 80 88
Masukkan NIM mahasiswa 3: 123321123
Masukkan nama mahasiswa 3: Warno
Masukkan 3 nilai mahasiswa 3 (pisahkan dengan spasi): 66 77 88
Mahasiswa dengan nilai tertinggi di awal: Haidar (2311102276)
Mahasiswa dengan nilai tertinggi di akhir: Haidar (2311102276)

```

3.

```
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])
        }
    }
}
}3
```

#### Output

```
PS C:\Users\Lenovo> go run "e:\Ujian Alpro\Ujian 3.go"
Masukkan jumlah provinsi (maksimal 34):
3
Masukkan nama provinsi ke-1: Jawa
Masukkan populasi provinsi ke-1 (juta): 20
Masukkan pertumbuhan penduduk provinsi ke-1 (%): 20
Masukkan nama provinsi ke-2: Sumatra
Masukkan populasi provinsi ke-2 (juta): 15
Masukkan pertumbuhan penduduk provinsi ke-2 (%): 18
Masukkan nama provinsi ke-3: Sunda
Masukkan populasi provinsi ke-3 (juta): 22
Masukkan pertumbuhan penduduk provinsi ke-3 (%): 20

Provinsi dengan populasi terbesar: Sunda

Prediksi populasi setelah pertumbuhan:
Jawa: 24 juta
Sumatra: 17 juta
Sunda: 26 juta

Provinsi dengan pertumbuhan di atas 2%:
Jawa
Sumatra
Sunda
```



4.

```
package main

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

// Program ini dibuat oleh [NIM 2311102276]
func main() {
    fmt.Println("Program Median - Dibuat oleh NIM 2311102276")

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

    fmt.Println("Masukkan angka (masukkan -5313541 untuk berhenti):")
    for 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)
    }
}

// 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  
}
```

Output

```
PS C:\Users\Lenovo> go run "e:\Ujian Alpro\Ujian 4.go"  
Program Median - Dibuat oleh NIM 2311102276  
Masukkan angka (masukkan -5313541 untuk berhenti):  
10 5 8 12 20 -5313541  
Masukkan angka yang valid!  
10  
Median saat ini: 10.00  
5  
Median saat ini: 7.50  
8  
Median saat ini: 8.00  
12  
Median saat ini: 9.00  
20  
Median saat ini: 10.00  
-53113541  
Median saat ini: 9.00  
-5313541
```

5.

```
package main

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

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

// Program ini dibuat oleh [NIM 2311102276]
func main() {
    fmt.Println("Program Pemrosesan Suara Partai - Dibuat oleh NIM 2311102276")

    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
    })
}
```

```

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

    // Menampilkan output
    fmt.Println("Hasil perhitungan suara:")
    for _, p := range partaiList {
        fmt.Printf("%d(%d) ", p.ID, p.Suara)
    }
    fmt.Println() // Tambahkan baris baru setelah output
}

```

Output

```

PS C:\Users\Lenovo> go run "e:\Ujian Alpro\Ujian 5.go"
Program Pemrosesan Suara Partai - Dibuat oleh NIM 2311102276
Masukkan data (akhiri dengan -1):
1 2 1 3 2 1 3 3 -1
Hasil perhitungan suara:
1(3) 3(3) 2(2)
PS C:\Users\Lenovo>

```