

Nama : Tegar Aji Pangestu

NIM : 2311102021

Kelas : II IF A

1.

```
package main

import "fmt"

// Konstanta untuk jumlah maksimal elemen
const maxElements = 2022

// Tipe data untuk himpunan
type set [maxElements]int

// Fungsi untuk memeriksa apakah nilai sudah ada dalam himpunan
func exist(T set, n int, val int) bool {
    for i := 0; i < n; i++ {
        if T[i] == val {
            return true
        }
    }
    return false
}

// Fungsi untuk menginput himpunan dari pengguna
func inputSet_2311102021(T *set, n *int) {
    var val int
    *n = 0
    for {
        fmt.Scan(&val)
        if exist(*T, *n, val) {
            break
        }
        T[*n] = val
        (*n)++
    }
}

// Fungsi untuk mencari irisan dari dua himpunan
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 untuk mencetak elemen-elemen dalam himpunan
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() {
    fmt.Printf("NIM: %s\n")
    var s1, s2, s3 set
    var n1, n2, n3 int
    fmt.Println("Masukkan elemen-elemen himpunan pertama (akhiri dengan elemen duplikat):")
    inputSet_2311102021(&s1, &n1)
    fmt.Println("Masukkan elemen-elemen himpunan kedua (akhiri dengan elemen duplikat):")
    inputSet_2311102021(&s2, &n2)
    findIntersection(s1, s2, n1, n2, &s3, &n3)
    fmt.Println("Irisan dari kedua himpunan adalah:")
    printSet(s3, n3)
}

```

```

Masukkan elemen-elemen himpunan pertama (akhiri dengan elemen duplikat):
11 28 33 64 95 16 100 15 64 3 11 7 28 33 6 28
Masukkan elemen-elemen himpunan kedua (akhiri dengan elemen duplikat):
Irisan dari kedua himpunan adalah:
11 28 33
PS C:\Users\reza\OneDrive\Dokumen\quiz praktikum alpro>

```

2.

```

package main

import "fmt"

const nMax = 51

// untuk menyimpan data mahasiswa
type mahasiswa_2311102021 struct {
    NIM    string
    nama   string
    nilai  int
}

// array untuk menyimpan data mahasiswa
type arrayMahasiswa [nMax]mahasiswa_2311102021

// fungsi untuk menerima input data mahasiswa
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 untuk mencari nilai pertama seorang mahasiswa berdasarkan NIM
func cariNilaiPertama(T arrayMahasiswa, N int, nim string) int {
    for i := 0; i < N; i++ {
        if T[i].NIM == nim {
            return T[i].nilai
        }
    }
    return -1
}

// fungsi untuk mencari nilai terbesar seorang mahasiswa berdasarkan NIM
func cariNilaiTerbesar(T arrayMahasiswa, N int, nim string) int {
    maxNilai := -1
    found := false
    for i := 0; i < N; i++ {
        if T[i].NIM == nim {
            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 string

    inputMahasiswa(&dataMahasiswa, &jumlahData)

    // fungsi mencari nilai pertama berdasarkan NIM
    fmt.Print("Masukkan NIM untuk mencari nilai pertama: ")
    fmt.Scan(&nim)
    nilaiPertama := cariNilaiPertama(dataMahasiswa, jumlahData, nim)
    if nilaiPertama != -1 {
        fmt.Printf("Nilai pertama untuk NIM %s adalah %d\n", nim,
nilaiPertama)
    } else {
        fmt.Printf("Data dengan NIM %s tidak ditemukan\n", nim)
    }

    // fungsi mencari nilai terbesar berdasarkan NIM
    fmt.Print("Masukkan NIM untuk mencari nilai terbesar: ")
    fmt.Scan(&nim)
    nilaiTerbesar := cariNilaiTerbesar(dataMahasiswa, jumlahData, nim)
    if nilaiTerbesar != -1 {
        fmt.Printf("Nilai terbesar untuk NIM %s adalah %d\n", nim,
nilaiTerbesar)
    } else {
        fmt.Printf("Data dengan NIM %s tidak ditemukan\n", nim)
    }
}

```

```

Masukkan jumlah data mahasiswa: 3
Data mahasiswa ke-1:
Masukkan NIM: 2311102021
Masukkan nama: aji
Masukkan nilai: 90
Data mahasiswa ke-2:
Masukkan NIM: 2311103030
Masukkan nama: aryo
Masukkan nilai: 70
Data mahasiswa ke-3:
Masukkan NIM: 2311102222
Masukkan nama: agus
Masukkan nilai: 50
Masukkan NIM untuk mencari nilai pertama: 2311102021
Nilai pertama untuk NIM 2311102021 adalah 90
Masukkan NIM untuk mencari nilai terbesar: 2311103030
Nilai terbesar untuk NIM 2311103030 adalah 70

```

3.

```

package main

import (
    "fmt"
    "strings"
)

const nProv = 34

type (
    NamaProv    [nProv]string
    PopProv     [nProv]int
    TumbuhProv  [nProv]float64
)

// fungsi untuk menginput data provinsi
func InputData_2311102021(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 : ")
        fmt.Scanln(&pop[i])
        fmt.Printf("Masukkan angka pertumbuhan penduduk provinsi : ")
        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 provinsi berdasarkan nama
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    NamaProv
        pop      PopProv
        tumbuh  TumbuhProv
        cari     string
    )

    InputData_2311102021(&nama, &pop, &tumbuh)

    tercepat := ProvinsiTercepat(tumbuh)
    fmt.Printf("Provinsi dengan pertumbuhan tercepat: %s\n",
nama[tercepat])

    // mencari provinsi berdasarkan nama
    fmt.Println("Masukkan nama provinsi yang ingin dicari:")
    fmt.Scanln(&cari)
    indeks := IndeksProvinsi(nama, 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, pop, tumbuh)
}

```

```

Masukkan nama provinsi ke-32: banten
Masukkan populasi provinsi : 7286863
Masukkan angka pertumbuhan penduduk provinsi : 62663
Masukkan nama provinsi ke-33: kepri
Masukkan populasi provinsi : 7387872
Masukkan angka pertumbuhan penduduk provinsi : 7234
Masukkan nama provinsi ke-34: sulteng
Masukkan populasi provinsi : 72676763
Masukkan angka pertumbuhan penduduk provinsi : 6767

```

```
Masukkan nama provinsi yang ingin dicari:  
jateng  
Provinsi jateng ditemukan pada indeks 2
```

```
Prediksi provinsi dengan pertumbuhan > 2%:  
sulawesi: Populasi tahun depan = 402000000  
sumatra: Populasi tahun depan = 900300000  
jateng: Populasi tahun depan = 1604000000  
jabar: Populasi tahun depan = 4602300000  
jatim: Populasi tahun depan = 2401200000  
jakarta: Populasi tahun depan = 96008000000  
sumut: Populasi tahun depan = 2580238703  
sumbar: Populasi tahun depan = 394976000  
sumsel: Populasi tahun depan = 5790030215919  
bali: Populasi tahun depan = 45893933240990  
kalbar: Populasi tahun depan = 3844850052428  
kaltim: Populasi tahun depan = 21971754728  
kalteng: Populasi tahun depan = 5227247015856  
kalsel: Populasi tahun depan = 464984842500  
ntb: Populasi tahun depan = 45622088736  
ulteng: Populasi tahun depan = 564000642  
sulsel: Populasi tahun depan = 62055844291168  
sulbar: Populasi tahun depan = 68853401607  
aceh: Populasi tahun depan = 53717490126  
riau: Populasi tahun depan = 31435741455833  
jambi: Populasi tahun depan = 57409541697078  
bengkulu: Populasi tahun depan = 6885284591042  
lampung: Populasi tahun depan = 6537517828032  
jogja: Populasi tahun depan = 6833337060588  
gorontalo: Populasi tahun depan = 6467177457482  
maluku: Populasi tahun depan = 5311860978399  
malukuutara: Populasi tahun depan = 3872203401  
papua: Populasi tahun depan = 555786788569892  
at: Populasi tahun depan = 323351846048  
banten: Populasi tahun depan = 456623983032  
kepri: Populasi tahun depan = 53451253920  
sulteng: Populasi tahun depan = 491876331984
```

4.

```
package main  
  
import (  
    "fmt"  
    "sort"  
)  
  
// fungsi untuk menghitung median  
func median_2311102021(data []int) float64 {  
    n := len(data)  
    if n == 0 {  
        return 0  
    }  
    // Jika jumlah data ganjil, median adalah nilai tengah  
    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 []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)
        fmt.Printf("Median: %.2f\n", median_2311102021(data))
    } else {
        // Tambahkan bilangan ke dalam array
        data = append(data, input)
    }
}
}

```

```

Masukkan bilangan (akhiri dengan -5313541):
2 3 1 0 8 9 0 -5313541
Median: 2.00
Median: 3.00

```

5.

```

package main

import (
    "fmt"
)

const NMAX = 1000000

type Partai_2311102021 struct {
    nama  int
    suara int
}

// Tipe array untuk menyimpan data partai
type TabPartai [NMAX]Partai_2311102021

func main() {
    var t TabPartai
    var n int
    var input int

    n = 0

    // Input suara partai
    for {
        fmt.Scan(&input)
        if input == -1 {
            break
        }
    }
}

```

```

        // Cari posisi partai berdasarkan nama
        pos := posisi(t, n, input)

        if pos == -1 {
            t[n].nama = input
            t[n].suara = 1
            n++
        } else {
            t[pos].suara++
        }
    }

    // Pengurutan secara descending berdasarkan jumlah suara
    insertionSortDescending(&t, n)

    // Tampilkan hasil
    for i := 0; i < n; i++ {
        fmt.Printf("%d(%d) ", t[i].nama, t[i].suara)
    }
    fmt.Println()
}

// 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
func insertionSortDescending(t *TabPartai, n int) {
    for i := 1; i < n; i++ {
        key := t[i]
        j := i - 1

        // Pindahkan elemen yang lebih kecil ke kanan
        for j >= 0 && t[j].suara < key.suara {
            t[j+1] = t[j]
            j--
        }
        t[j+1] = key
    }
}

```