

Nama : Erwin Rivaldo Silaban

NIM : 2311102248

Kelas : IF-11- 06

1. SOAL No.1

```
package main

import (
    "fmt"
)

func main() {
    // Input dua baris himpunan
    var erwin248set1, erwin248set1set2 []int

    fmt.Println("Masukkan elemen baris pertama (akhiri dengan angka duplikat):")
    erwin248set1 = readSet()

    fmt.Println("Masukkan elemen baris kedua (akhiri dengan angka duplikat):")
    erwin248set1set2 = readSet()

    // Temukan irisan dari kedua himpunan
    intersection := findIntersection(erwin248set1, erwin248set1set2)

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

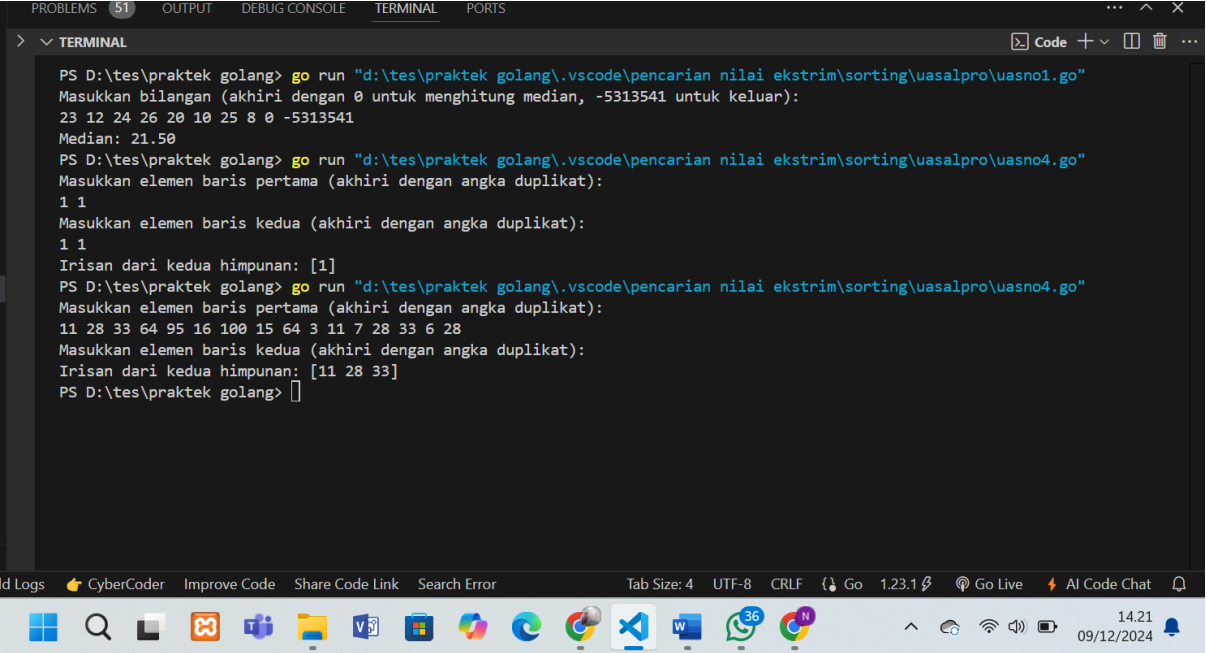
    return result
}

func findIntersection(erwin248set1, erwin248set1set2 []int) []int {
    setMap := make(map[int]bool)
    for _, num := range erwin248set1 {
        setMap[num] = true
    }

    var intersection []int
    for _, num := range erwin248set1set2 {
        if setMap[num] {
            intersection = append(intersection, num)
        }
    }

    return intersection
}
```

HASIL PROGRAM



2. SOAL No. 2

```
package main

import "fmt"

const nMax int = 51

type mahasiswa248 struct {
    NIM string
    nama string
    nilai int
}

type arraymahasiswa248 [nMax]mahasiswa

func menerimaMasukan(N int, arrayMhs *arrayMahasiswa) {
    for i := 0; i < N; i++ {
        fmt.Printf("Masukkan data mahasiswa248 ke-%d:\n", i+1)
        fmt.Print("NIM: ")
        fmt.Scanln(&arrayMhs[i].NIM)
        fmt.Print("Nama: ")
        fmt.Scanln(&arrayMhs[i].nama)
        fmt.Print("Nilai: ")
        fmt.Scanln(&arrayMhs[i].nilai)
    }
}

func mencariNilaiPertama(NIM string, arrayMhs arrayMahasiswa, N int) int {
    for i := 0; i < N; i++ {
        if arrayMhs[i].NIM == NIM {
            return arrayMhs[i].nilai
        }
    }
    return -1 // Nilai tidak ditemukan
}

func mencariNilaiTerbesar(NIM string, arrayMhs arrayMahasiswa, N int) int {
    nilaiTerbesar := -1
    for i := 0; i < N; i++ {
        if arrayMhs[i].NIM == NIM && arrayMhs[i].nilai > nilaiTerbesar {
            nilaiTerbesar = arrayMhs[i].nilai
        }
    }
    return nilaiTerbesar
}
```

```
func menampilkanHasil(NIM string, arrayMhs arrayMahasiswa, N int) {
    nilaiPertama := mencariNilaiPertama(NIM, arrayMhs, N)
    if nilaiPertama == -1 {
        fmt.Printf("mahasiswa248 dengan NIM %s tidak ditemukan.\n", NIM)
        return
    }
    nilaiTerbesar := mencariNilaiTerbesar(NIM, arrayMhs, N)
    fmt.Printf("Nilai pertama mahasiswa248 dengan NIM %s adalah %d\n", NIM,
nilaiPertama)
    fmt.Printf("Nilai terbesar mahasiswa248 dengan NIM %s adalah %d\n", NIM,
nilaiTerbesar)
}

func main() {
    var arrayMhs arrayMahasiswa
    var N int
    fmt.Print("Masukkan jumlah data mahasiswa: ")
    fmt.Scanln(&N)

    if N > nMax {
        fmt.Printf("Jumlah data tidak boleh lebih dari %d.\n", nMax)
        return
    }

    menerimaMasukan(N, &arrayMhs)

    var NIM string
    fmt.Print("Masukkan NIM mahasiswa248 yang ingin dicari: ")
    fmt.Scanln(&NIM)
    menampilkanHasil(NIM, arrayMhs, N)
}
```

Hasil Program

```

  ▾ TERMINAL
  NIM: 113
  Nilai: 60
  Masukkan data mahasiswa ke-7:
  NIM: 113
  Nilai: 60
  Masukkan data mahasiswa ke-7:
  Nilai: 60
  Masukkan data mahasiswa ke-7:
  Nilai: 60
  Masukkan data mahasiswa ke-7:
  Nilai: 60
  Masukkan data mahasiswa ke-7:
  Nilai: 60
  Masukkan data mahasiswa ke-7:
  NIM: 113
  Nama: jojo
  Nilai: 60
  Masukkan data mahasiswa ke-8:
  NIM: 113
  Nama: jojo
  Nilai: 50
  Masukkan data mahasiswa ke-9:
  NIM: 118
  Nama: rere
  Nilai: 88
  Masukkan data mahasiswa ke-10:
  NIM: 119
  Nama: roro
  Nilai: 100
  Masukkan NIM mahasiswa yang ingin dicari: 118
  Nilai pertama mahasiswa dengan NIM 118 adalah 80
  Nilai terbesar mahasiswa dengan NIM 118 adalah 88
  PS D:\tes\praktek golang> go run "d:\tes\praktek golang\.vscode\pencarian nilai ekstrim\sorting\uasalpro\uasno2.go"
```

3. SOAL No. 3

```
package main

import "fmt"

const nProv int = 34

// Tipe data array untuk menyimpan nama provinsi, populasi, dan angka pertumbuhan
type namaprov248 [nProv]string
type PopProv [nProv]int
type TumbuhProv [nProv]float64

// Fungsi untuk memasukkan data provinsi, populasi, dan angka pertumbuhan
func InputData(prov *namaprov248, pop *PopProv, tumbuh *TumbuhProv) {
```

```

    fmt.Println("Masukkan 34 nama provinsi, populasi, dan angka pertumbuhan:")
    var i int
    for i = 0; i < nProv; i++ {
        fmt.Scan(&prov[i], &pop[i], &tumbuh[i])
    }
}

// Fungsi untuk mencari provinsi dengan pertumbuhan tercepat
func ProvinsiTercepat(tumbuh TumbuhProv) int {
    var idx int = 0
    var i int
    for i = 1; i < nProv; i++ {
        if tumbuh[idx] < tumbuh[i] {
            idx = i
        }
    }
    return idx
}

// Fungsi untuk menampilkan prediksi jumlah penduduk provinsi dengan pertumbuhan > 2%
func Prediksi(prov namaprov248, pop PopProv, tumbuh TumbuhProv) {
    fmt.Println("Prediksi jumlah penduduk tahun depan untuk provinsi dengan pertumbuhan di atas 2%:")
    var i int
    var result float64
    for i = 0; i < nProv; i++ {
        if tumbuh[i] > 0.02 {
            result = (1 + tumbuh[i]) * float64(pop[i])
            fmt.Printf("%s %.0f\n", prov[i], result) // Format lebih rapi
        }
    }
}

// Fungsi untuk mencari indeks provinsi berdasarkan nama
func IndeksProvinsi(prov namaprov248, nama string) int {
    var found int = -1
    var i int = 0
    for i < nProv && found == -1 {
        if prov[i] == nama {
            found = i
        }
        i++
    }
    return found
}

func main() {
    var TProvinsi namaprov248
    var TPopulasi PopProv
    var TPertumbuhan TumbuhProv
    var cari string
    var idxTercepat, idxProvinsi int

    // Memasukkan data provinsi
    InputData(&TProvinsi, &TPopulasi, &TPertumbuhan)

    // Menerima nama provinsi yang ingin dicari
    fmt.Println("Masukkan nama provinsi yang ingin dicari:")
    fmt.Scan(&cari)

    // Mencari provinsi dengan pertumbuhan tercepat dan menampilkan nama provinsi
    idxTercepat = ProvinsiTercepat(TPertumbuhan)
    fmt.Printf("Nama provinsi dengan angka pertumbuhan tercepat: %s\n",
    TProvinsi[idxTercepat])

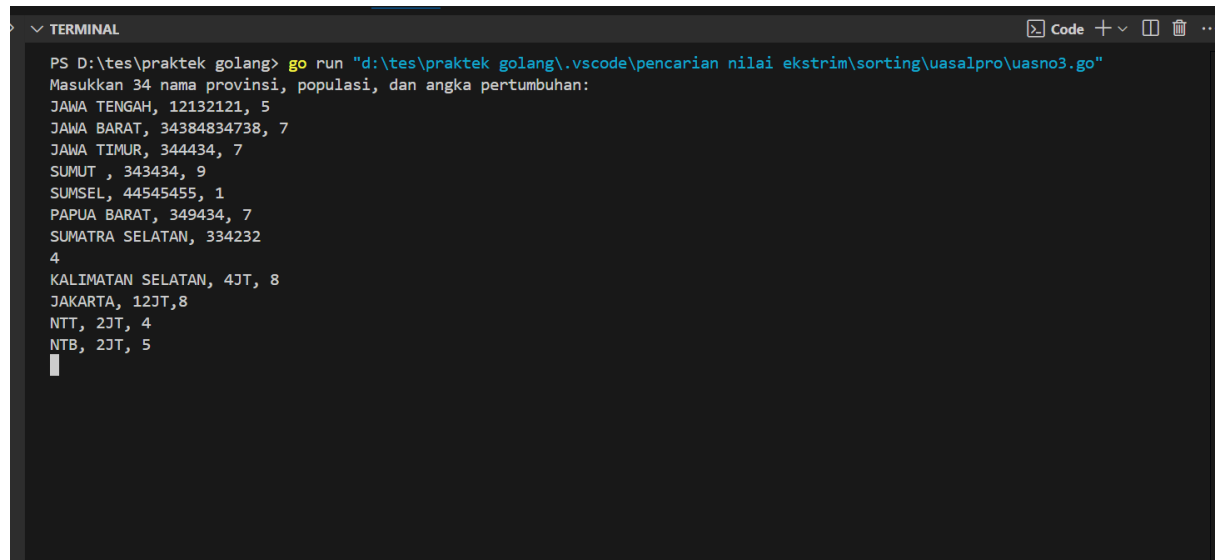
    // Mencari indeks provinsi yang dicari dan menampilkan nama provinsi
    idxProvinsi = IndeksProvinsi(TProvinsi, cari)
    if idxProvinsi != -1 {
        fmt.Printf("Indeks provinsi yang dicari sesuai nama: %d\n", idxProvinsi)
    } else {
        fmt.Println("Provinsi tidak ditemukan.")
    }
}

```

```
// Menampilkan prediksi jumlah penduduk provinsi dengan pertumbuhan lebih dari 2%
Prediksi(TProvinsi, TPopulasi, TPertumbuhan)
}
```

HASIL

Untuk hasil dari program ini harus mengisi 34 baris dulu baru ada
gk sempat isi di kejar waktu



```
PS D:\tes\praktek golang> go run "d:\tes\praktek golang\.vscode\pencarian nilai ekstrim\sorting\uasalpro\uasno3.go"
Masukkan 34 nama provinsi, populasi, dan angka pertumbuhan:
JAWA TENGAH, 12132121, 5
JAWA BARAT, 34384834738, 7
JAWA TIMUR, 344434, 7
SUMUT , 343434, 9
SUMSEL, 44545455, 1
PAPUA BARAT, 349434, 7
SUMATRA SELATAN, 334232
4
KALIMATAN SELATAN, 4JT, 8
JAKARTA, 12JT,8
NTT, 2JT, 4
NTB, 2JT, 5
█
```

4. SOAL No. 4

```
package main

import "fmt"

// NMAX = batas maksimum jumlah data yang dapat disimpan
const NMAX = 1000000
type arrInt [NMAX]int
// I.S. T adalah array dengan n elemen yang belum t
// F.S. T menjadi array dengan elemen-elemen terurut secara menaik menggunakan
algoritma selection sort
func sorting(T *arrInt, n int) {
    for erwin248 := 0; erwin248 < n-1; erwin248++ {
        idxMin := erwin248
        for i := erwin248 + 1; i < n; i++ {
            if T[idxMin] > T[i] {
                idxMin = i
            }
        }
        // Tukar elemen
        T[erwin248], T[idxMin] = T[idxMin], T[erwin248]
    }
}

// Fungsi median
// Mengembalikan median dari array T yang sudah terurut berisi n elemen
func median(T arrInt, n int) float64 {
    mid := n / 2 // Indeks tengah array
    if n%2 == 0 {
        // Jika jumlah elemen genap, median adalah rata-rata dua elemen tengah
        return float64(T[mid-1]+T[mid]) / 2.0
    }
    // Jika jumlah elemen ganjil, median adalah elemen tengah
    return float64(T[mid])
}

// Fungsi utama (main)
func main() {
    var A arrInt
    var x, n int
    n = 0
```

```

    fmt.Println("Masukkan bilangan (akhiri dengan 0 untuk menghitung median, -5313541
untuk keluar):")
    fmt.Scan(&x)

    // Proses setiap bilangan sampai menemukan marker -5313541 atau mencapai batas
    NMAX
    for x != -5313541 && n < NMAX {
        if x == 0 {
            if n > 0 {
                sorting(&A, n)
                fmt.Printf("Median: %.2f\n", median(A, n))
            } else {
                fmt.Println("Tidak ada data untuk dihitung median.")
            }
        } else {
            // Tambahkan bilangan ke array A
            A[n] = x
            n++
        }
        // Baca bilangan berikutnya
        fmt.Scan(&x)
    }
}

```

HASIL PROGRAM

```

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\.vscode\pencarian nilai ekstrim\sorting\uasalpro\uasno1.go"
Masukkan bilangan (akhiri dengan 0 untuk menghitung median, -5313541 untuk keluar):
23 12 24 26 20 10 25 8 0 -5313541
Median: 21.50
PS D:\tes\praktek golang>

```

5. SOAL No. 5

```

package main

import (
    "fmt"
    "sort"
)

const NMAX = 1000000

type arrInt []int

// Mengembalikan panjang
func (erwin248 arrInt) Len() int { return len(erwin248) }

// Menukar elemen
func (erwin248 arrInt) Swap(i, j int) { erwin248[i], erwin248[j] = erwin248[j], erwin248[i] }

// Membandingkan elemen untuk sorting
func (erwin248 arrInt) Less(i, j int) bool { return erwin248[i] < erwin248[j] }

// Mengurutkan elemen dalam array
func sorting(arr arrInt) {
    sort.Sort(arr)
}

// Menghitung frekuensi setiap elemen dalam array
func countFrequency(arr arrInt) map[int]int {

```

```

    freq := make(map[int]int)
    for _, v := range arr {
        freq[v]++
    }
    return freq
}

func main() {
    var masukan arrInt
    var keluaran []string
    var input int

    fmt.Println("Masukkan elemen (akhiri dengan -1):")
    for {
        fmt.Scan(&input)
        if input == -1 {
            break
        }
        masukan = append(masukan, input)
    }

    // Hitung frekuensi elemen
    freq := countFrequency(masukan)

    // Format keluaran sesuai spesifikasi
    for num, count := range freq {
        keluaran = append(keluaran, fmt.Sprintf("%d(%d)", num, count))
    }

    // Urutkan keluaran berdasarkan angka
    sort.Strings(keluaran)

    // Tampilkan hasil
    fmt.Println("Keluaran:", keluaran)
}

```

OUTPUT

```

PS D:\tes\praktek golang> go run "d:\tes\praktek golang\.vscode\pencarian nilai ekstrim\sorting\uasalpro\uasno5.go"
Masukkan elemen (akhiri dengan -1):
5 1 1 1 1 1 1 1 3 3 3 3 3 2 2 5 5 5 5 4 3 2 2 2 -1
Keluaran: [1(7) 2(5) 3(7) 4(1) 5(5)]
PS D:\tes\praktek golang> go run "d:\tes\praktek golang\.vscode\pencarian nilai ekstrim\sorting\uasalpro\uasno5.go"
Masukkan elemen (akhiri dengan -1):
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 -1
Keluaran: [8(21)]
PS D:\tes\praktek golang> 

```