

Marsep Trianto Pakondo

2311102251

Soal no 1

```
package main

import "fmt"

type set_2311102251 [2022]int

// Fungsi ini untuk mengembalikan true kalau bilangan val yang di isi di inputSet ada di dalam array T
func exist(T set_2311102251, n int, val int) bool {
    for i := 0; i < n; i++ {
        if T[i] == val {
            return true
        }
    }
    return false
}

// Prosedure ini untuk mengisi array T berisi sejumlah n bilangan bulat yang berasal dari masukan
// Lalu masukan akan berhenti kalau bilangan ada yang duplikat/sama atau array penuh
func inputSet(T *set_2311102251, n *int) {
    var x int
    *n = 0
    for {
        fmt.Scan(&x)
        if exist(*T, *n, x) {
            break
        }
        T[*n] = x
        *n++
    }
}
```

```

        *n++
    }
}

// Prosedure ini untuk menghasilkan irisan dari himpunan T1 dan T2
func findIntersection(T1,T2 set_2311102251, n,m int, T3 *set_2311102251, h *int) {
    *h = 0
    for i := 0; i < n; i++ {
        if exist(T2, m, T1[i]) {
            T3[*h] = T1[i]
            (*h)++
        }
    }
}

// Prosedure ini untuk menampilkan isi array T
func printSet(T set_2311102251, n int) {
    for i := 0; i < n; i++ {
        if i > 0 {
            fmt.Print(" ")
        }
        fmt.Print(T[i])
    }
    fmt.Println()
}

func main() {
    var s1, s2, s3 set_2311102251
    var n1, n2, n3 int

    fmt.Println("Masukkan elemen pertama:")

```

```

        inputSet(&s1, &n1) // panggil prosed inputSet untuk menginputkan bilangan bulat kedalam
        array himpunan pertama

        fmt.Println("Masukkan elemen kedua:")

        inputSet(&s2, &n2) // panggil prosed inputSet untuk menginputkan bilangan bulat kedalam
        array himpunan kedua

        findIntersection(s1, s2, n1, n2, &s3, &n3) // panggil prosed findIntersection untuk
        menghasilkan irisan

        fmt.Println("Irisan dari kedua elemen:")

        printSet(s3, n3) // panggil prosed printSet untuk menampilkan isi array
    }

```

```

PS E:\alpro 2\src> go run "e:\alpro 2\src\test_alpro\tempCodeRunnerFile.go"
Masukkan elemen pertama:
1 1
Masukkan elemen kedua:
1 1
Irisan dari kedua elemen:
1
PS E:\alpro 2\src>

```

Soal no 2

```

package main

import "fmt"

// Konstanta
const nMax int = 51

// Definisi struct

```

```

type mahasiswa_2311102251 struct {
    NIM, nama string
    nilai int
}

// Array untuk menyimpan data mahasiswa
var arrayMahasiswa [nMax]mahasiswa_2311102251

var n int // Jumlah data mahasiswa saat ini

// Fungsi untuk menambahkan data mahasiswa
func tambahMahasiswa(NIM, nama string, nilai int) {
    if n < nMax {
        arrayMahasiswa[n] = mahasiswa_2311102251{NIM: NIM, nama: nama, nilai: nilai}
        n++
    } else {
        fmt.Println("Array mahasiswa penuh!")
    }
}

// Fungsi untuk mencari nilai pertama mahasiswa dengan NIM tertentu
func cariNilaiPertama(NIM string) (int, bool) {
    for i := 0; i < n; i++ {
        if arrayMahasiswa[i].NIM == NIM {
            return arrayMahasiswa[i].nilai, true
        }
    }
    return 0, false
}

// Fungsi untuk mencari nilai terbesar mahasiswa dengan NIM tertentu

```

```

func cariNilaiTerbesar(NIM string) (int, bool) {
    ketemu := false
    maxNilai := 0
    for i := 0; i < n; i++ {
        if arrayMahasiswa[i].NIM == NIM {
            if !ketemu || arrayMahasiswa[i].nilai > maxNilai {
                maxNilai = arrayMahasiswa[i].nilai
                ketemu = true
            }
        }
    }
    return maxNilai, ketemu
}

```

```

func main() {
    var jumlah int
    fmt.Println("Masukkan jumlah data mahasiswa:")
    fmt.Scan(&jumlah)

    if jumlah > nMax {
        fmt.Printf("Jumlah data maksimal adalah %d\n", nMax)
        return
    }

    // Input data mahasiswa
    for i := 0; i < jumlah; i++ {
        var NIM, nama string
        var nilai int
        fmt.Printf("Masukkan NIM mahasiswa ke-%d : ", i+1)
        fmt.Scan(&NIM)
        fmt.Printf("Masukkan Nama mahasiswa ke-%d : ", i+1)
    }
}

```

```

        fmt.Scan(&nama)
        fmt.Printf("Masukkan Nilai mahasiswa ke-%d : ", i+1)
        fmt.Scan(&nilai)
        tambahMahasiswa(NIM, nama, nilai)
    }

    // Pencarian nilai pertama
    var cariNIM string
    fmt.Println("Masukkan NIM untuk mencari nilai pertama:")
    fmt.Scan(&cariNIM)
    // Pengecekan untuk mencari nilai NIM pertama
    nilai, found := cariNilaiPertama(cariNIM)
    if found {
        fmt.Printf("Nilai pertama mahasiswa dengan NIM %s adalah %d\n", cariNIM, nilai)
    } else {
        fmt.Printf("Mahasiswa dengan NIM %s tidak ditemukan.\n", cariNIM)
    }

    // Pencarian nilai terbesar
    fmt.Println("Masukkan NIM untuk mencari nilai terbesar:")
    fmt.Scan(&cariNIM)
    // Pengecekan untuk mencari nilai terbesar menggunakan NIM
    nilai, found = cariNilaiTerbesar(cariNIM)
    if found {
        fmt.Printf("Nilai terbesar mahasiswa dengan NIM %s adalah %d\n", cariNIM, nilai)
    } else {
        fmt.Printf("Mahasiswa dengan NIM %s tidak ditemukan.\n", cariNIM)
    }
}

```

```

PS E:\alpro 2\src> go run "e:\alpro 2\src\test_alpro\no2.go"
Masukkan jumlah data mahasiswa:
5
Masukkan NIM mahasiswa ke-1 : 114
Masukkan Nama mahasiswa ke-1 : Nana
Masukkan Nilai mahasiswa ke-1 : 97
Masukkan NIM mahasiswa ke-2 : 113
Masukkan Nama mahasiswa ke-2 : Jojo
Masukkan Nilai mahasiswa ke-2 : 70
Masukkan NIM mahasiswa ke-3 : 118
Masukkan Nama mahasiswa ke-3 : Rere
Masukkan Nilai mahasiswa ke-3 : 113
Masukkan NIM mahasiswa ke-4 : 113
Masukkan Nama mahasiswa ke-4 : Jojo
Masukkan Nilai mahasiswa ke-4 : 80
Masukkan NIM mahasiswa ke-5 : 118
Masukkan Nama mahasiswa ke-5 : Rere
Masukkan Nilai mahasiswa ke-5 : 65
Masukkan NIM untuk mencari nilai pertama:
113
Nilai pertama mahasiswa dengan NIM 113 adalah 70
Masukkan NIM untuk mencari nilai terbesar:
118
Nilai terbesar mahasiswa dengan NIM 118 adalah 113
PS E:\alpro 2\src>

```

Soal no 3

```

package main

import (
    "fmt"
)

const nProv int = 34

// Definisi tipe data
type NamaProv [nProv]string
type PopProv_2311102251 [nProv]int
type TumbuhProv [nProv]float64

// Peosedure ini untuk menginputkan data nama, jumlah populasi, dan angka pertumbuhan provinsi
func InputData(prov *NamaProv, pop *PopProv_2311102251, tumbuh *TumbuhProv) {
    fmt.Println("Masukkan data provinsi (nama, populasi, pertumbuhan):")
    for i := 0; i < nProv; i++ {
        fmt.Printf("Data provinsi ke-%d: ", i+1)
        fmt.Printf("Nama provinsi : ")
        fmt.Scan(&prov[i])
        fmt.Printf("Jumlah Populasi provinsi : ")
        fmt.Scan(&pop[i])
        fmt.Printf("Angka Pertumbuhan provinsi : ")
        fmt.Scan(&tumbuh[i])
    }
}

```

```

// Fungsi mengetahui pertumbuhan provinsi tercepat
func ProvinsiTercepat(tumbuh TumbuhProv) int {
    idxMax := 0
    for i := 1; i < nProv; i++ {
        if tumbuh[i] > tumbuh[idxMax] {
            idxMax = i
        }
    }
    return idxMax
}

// Fungsi ini untuk menampilkan nama provinsi dan Prediksi jumlah populasi provinsi
func Prediksi(prov NamaProv, pop PopProv_2311102251, tumbuh TumbuhProv) {
    fmt.Println("Prediksi populasi untuk provinsi dengan pertumbuhan > 2%:")
    for i := 0; i < nProv; i++ {
        if tumbuh[i] > 2.0 {
            prediksi := float64(pop[i]) * (1 + tumbuh[i]/100)
            fmt.Printf("%s: %.0f\n", prov[i], prediksi)
        }
    }
}

// Fungsi ini untuk mencari nama Provinsi
func IndeksProvinsi(prov NamaProv, nama string) int {
    for i := 0; i < nProv; i++ {
        if prov[i] == nama {
            return i
        }
    }
    return -1
}

func main() {
    var prov NamaProv
    var pop PopProv_2311102251
    var tumbuh TumbuhProv
    var cariNama string

    // Panggil prosedur InputData untuk menginputkan data
    InputData(&prov, &pop, &tumbuh)

    // Panggil fungsi ProvinsiTercepat untuk menampilkan provinsi dengan pertumbuhan tercepat
    idxTercepat := ProvinsiTercepat(tumbuh)

```



```

    fmt.Printf("Provinsi dengan pertumbuhan tercepat: %s\n", prov[idxTercepat])

    // Mencari nama provinsi
    fmt.Println("Masukkan nama provinsi yang ingin dicari:")
    fmt.Scan(&cariNama)
    idx := IndeksProvinsi(prov, cariNama)
    if idx != -1 {
        fmt.Printf("Indeks provinsi %s adalah %d\n", cariNama, idx)
    } else {
        fmt.Printf("Provinsi %s tidak ditemukan.\n", cariNama)
    }

    // Menampilkan prediksi populasi dengan memanggil prosedur Prediksi
    Prediksi(prov, pop, tumbuh)
}

```

```

PS E:\alpro 2\src> go run "e:\alpro 2\src\test_alpro\n03.go"
Masukkan data provinsi (nama, populasi, pertumbuhan):
Data provinsi ke-1: Nama provinsi : SulSel
Jumlah Populasi provinsi : 2000
Angka Pertumbuhan provinsi : 3
Data provinsi ke-2: Nama provinsi : Jakbar
Jumlah Populasi provinsi : 1500
Angka Pertumbuhan provinsi : 2
Data provinsi ke-3: Nama provinsi : Sumut
Jumlah Populasi provinsi : 5000
Angka Pertumbuhan provinsi : 4
Provinsi dengan pertumbuhan tercepat: Sumut
Masukkan nama provinsi yang ingin dicari:
Jakbar
Indeks provinsi Jakbar adalah 1
Prediksi populasi untuk provinsi dengan pertumbuhan > 2%:
SulSel: 2060
Sumut: 5200
PS E:\alpro 2\src>

```

Soal no 4

```

package main

import (
    "fmt"
)

const nMax int = 100000

```

```

// Definisi array bilangan bulat
type arrInt [nMax]int

// prosedur ini untuk sorting array secara ascending menggunakan Insertion Sort
func sorting(T *arrInt, n int) {
    for i := 1; i < n; i++ {
        key := T[i]
        j := i - 1
        for j >= 0 && T[j] > key {
            T[j+1] = T[j]
            j--
        }
        T[j+1] = key
    }
}

// fungsi ini untuk mencari median dari array yang telah diurutkan
func findMedian(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() {
    var arrBilBul arrInt
    var n int
    var median []float64

    fmt.Println("Masukkan bilangan bulat secara berurutan:")
    for {
        var input int
        fmt.Scan(&input)

        // Berhenti jika input negatif
        if input < 0 {
            break
        }

        // Jika input 0, cari median dari bilangan yang ada
        if input == 0 {
            if n == 0 {
                fmt.Println("Median: 0.00")
            }
        }
    }
}

```

```

        } else {
            sorting(&arrBilBul, n) // Urutkan array sebelum mencari median
            medianValue := findMedian(arrBilBul, n)
            median = append(median, medianValue)
            fmt.Printf("Median saat ini: %.2f\n", medianValue)
        }
        continue
    }

    // Tambahkan bilangan ke array
    if n < nMax {
        arrBilBul[n] = input
        n++
    } else {
        fmt.Println("Maksimum kapasitas array tercapai.")
    }
}

// Cetak semua median yang telah ditemukan
fmt.Println("\nDaftar Median:")
for i, m := range median {
    fmt.Printf("Median %d: %.2f\n", i+1, m)
}
}

```

```

PS E:\alpro 2\src> go run "e:\alpro 2\src\test_alpro\n04.go"
Masukkan bilangan bulat secara berurutan:
7
23
11
0
Median saat ini: 11.00
5
19
2
29
3
13
17
0
Median saat ini: 12.00
-5343

Daftar Median:
Median 1: 11.00
Median 2: 12.00
PS E:\alpro 2\src>

```

Soal no 5

```
package main

import "fmt"

const NMAX = 1000000

// Struktur untuk merepresentasikan partai
type Partai struct {
    nama, suara int
}

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

// Fungsi utama
func main() {
    var suara int
    tabPartai := make([]Partai, 0, NMAX)
    n := 0

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

        pos := posisi(tabPartai, n, suara)
        if pos == -1 {
            // Jika partai belum ada, tambahkan ke array
            tabPartai = append(tabPartai, Partai{nama: suara, suara: 1})
            n++
        } else {
            // Jika partai sudah ada, tambahkan suaranya
            tabPartai[pos].suara++
        }
    }
}
```

```

    }

    // Sorting dengan insertion sort secara descending berdasarkan suara
    for i := 1; i < n; i++ {
        key := tabPartai[i]
        j := i - 1
        for j >= 0 && (tabPartai[j].suara < key.suara || (tabPartai[j].suara == key.suara &&
tabPartai[j].nama > key.nama)) {
            tabPartai[j+1] = tabPartai[j]
            j--
        }
        tabPartai[j+1] = key
    }

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

```

```

● PS E:\alpro 2\src> go run "e:\alpro 2\src\test_alpro\no5.go"
5 1 1 1 1 1 1 3 3 3 3 2 2 5 5 5 5 4 3 2 2 2 2 -1
1(7) 2(6) 3(6) 5(6) 4(1)
○ PS E:\alpro 2\src>

```