

UJIAN PRAKTIKUM

Fahri Ramadhan

2311102024

IF 11 06

1.

Sourcecode

```
package main

import "fmt"

type set [2022]int

// Fungsi untuk mengecek apakah elemen sudah ada dalam array
func exist_231110204(T set, n int, val int) bool {
    for i := 0; i < n; i++ {
        if T[i] == val {
            return true
        }
    }
    return false
}

// Fungsi untuk mengisi array dengan bilangan unik
func inputSet(T *set, n *int) {
```

```

var val int

*n = 0

for {
    fmt.Scan(&val)

    if exist_231110204(*T, *n, val) {
        break
    }

    T[*n] = val

    (*n)++
}

}

// Fungsi untuk mencari irisan dari dua array
func findIntersection(T1, T2 set, n, m int, T3 *set, h *int) {
    *h = 0

    for i := 0; i < n; i++ {
        if exist_231110204(T2, m, T1[i]) && !exist_231110204(*T3, *h,
T1[i]) {
            T3[*h] = T1[i]

            (*h)++
        }
    }
}

// Fungsi untuk mencetak array secara horizontal
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() {
    var s1, s2, s3 set
    var n1, n2, n3 int

    inputSet(&s1, &n1)
    inputSet(&s2, &n2)
    findIntersection(s1, s2, n1, n2, &s3, &n3)
    printSet(s3, n3)
}

```

Output

```

PS D:\Ujian al2> go run "d:\Ujian al2\soal1.go"
11 28 33 64 95 16 100 15 64 3 11 7 28 33 6 28
11 28 33
PS D:\Ujian al2> 

```

2.

Sourcecode

```
package main

import "fmt"

const nMax = 51

type mahasiswa struct {
    NIM    string
    nama   string
    nilai  int
}

type arrayMahasiswa [nMax]mahasiswa

// Fungsi ini 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 ini 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 // Jika tidak ditemukan
}

// Fungsi ini untuk mencari nilai terbesar seorang mahasiswa berdasarkan
NIM

func cariNilaiTerbesar(T arrayMahasiswa, N int, nim string) int {
    maxNilai_2311102024 := -1
    found := false
    for i := 0; i < N; i++ {
        if T[i].NIM == nim {
            found = true
            if T[i].nilai > maxNilai_2311102024 {
                maxNilai_2311102024 = T[i].nilai
            }
        }
    }
}

```

```
        if found {
            return maxNilai_2311102024
        }

        return -1 // Jika tidak ditemukan
    }

func main() {
    var dataMahasiswa arrayMahasiswa
    var jumlahData int
    var nim string

    inputMahasiswa(&dataMahasiswa, &jumlahData)

    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)
    }

    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)  
        }  
    }  
}
```

Output

```
PS D:\Ujian al2> go run "d:\Ujian al2\soal2.go"  
Masukkan jumlah data mahasiswa: 4  
Data mahasiswa ke-1:  
Masukkan NIM: 111  
Masukkan nama: dean  
Masukkan nilai: 10  
Data mahasiswa ke-2:  
Masukkan NIM: 112  
Masukkan nama: adel  
Masukkan nilai: 70  
Data mahasiswa ke-3:  
Masukkan NIM: 111  
Masukkan nama: dean  
Masukkan nilai: 100  
Data mahasiswa ke-4:  
Masukkan NIM: 110  
Masukkan nama: kevin  
Masukkan nilai: 90  
Masukkan NIM untuk mencari nilai pertama: 111  
Nilai pertama untuk NIM 111 adalah 10  
Masukkan NIM untuk mencari nilai terbesar: 111  
Nilai terbesar untuk NIM 111 adalah 100  
PS D:\Ujian al2> █
```


Sourcecode

```
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(nama *NamaProv, pop *PopProv, tumbuh_2311102024 *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 %s: ", nama[i])
        fmt.Scanln(&pop[i])
        fmt.Printf("Masukkan angka pertumbuhan penduduk provinsi %s: ", nama[i])
        fmt.Scanln(&tumbuh_2311102024[i])
    }
}
```

```

    }

}

// Fungsi untuk mencari provinsi dengan pertumbuhan tercepat
func ProvinsiTercepat(tumbuh_2311102024 TumbuhProv) int {
    indeks := 0
    maks := tumbuh_2311102024[0]
    for i := 1; i < nProv; i++ {
        if tumbuh_2311102024[i] > maks {
            maks = tumbuh_2311102024[i]
            indeks = i
        }
    }
    return indeks
}

// Fungsi untuk mencari indeks 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
}

```

```
// Prosedur untuk menampilkan prediksi populasi provinsi

func Prediksi(nama NamaProv, pop PopProv, tumbuh_2311102024 TumbuhProv)
{
    fmt.Println("Prediksi provinsi dengan pertumbuhan > 2%:")
    for i := 0; i < nProv; i++ {
        if tumbuh_2311102024[i] > 0.02 {
            prediksi := float64(pop[i]) * (1 + tumbuh_2311102024[i])
            fmt.Printf("%s: Populasi tahun depan = %.0f\n", nama[i],
prediksi)
        }
    }
}

func main() {
    var (
        nama    NamaProv
        pop      PopProv
        tumbuh_2311102024 TumbuhProv
        cari     string
    )

    InputData(&nama, &pop, &tumbuh_2311102024)

    tercepat := ProvinsiTercepat(tumbuh_2311102024)

    fmt.Printf("Provinsi    dengan    pertumbuhan    tercepat:    %s\n",
nama[tercepat])
}
```

```
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_2311102024)
}
```

Output

```
Masukkan nama provinsi yang ingin dicari:
aceh
Provinsi aceh ditemukan pada indeks 1
Prediksi provinsi dengan pertumbuhan > 2%:
klaten: Populasi tahun depan = 5010000
aceh: Populasi tahun depan = 766373132
jawa: Populasi tahun depan = 23376335863332
kalimantan: Populasi tahun depan = 8019329001216
jj: Populasi tahun depan = 11742262
g: Populasi tahun depan = 1374608
g: Populasi tahun depan = 1190
2343: Populasi tahun depan = 136976
ger: Populasi tahun depan = 124786050
rgr: Populasi tahun depan = 1171658650
gdg: Populasi tahun depan = 804405
gfe: Populasi tahun depan = 0
sdfwe: Populasi tahun depan = 1235664
fwe: Populasi tahun depan = 1405300
efew: Populasi tahun depan = 1116050
e: Populasi tahun depan = 0
er: Populasi tahun depan = 2070
gr: Populasi tahun depan = 235620
er: Populasi tahun depan = 0
few: Populasi tahun depan = 850
rg: Populasi tahun depan = 12772620
ergt: Populasi tahun depan = 7621990
egrt: Populasi tahun depan = 804405
thtrh: Populasi tahun depan = 76140
htrh: Populasi tahun depan = 106690
gerg: Populasi tahun depan = 761870
ger: Populasi tahun depan = 2056104
PS D:\Ujian al2> █
```

4.

Sourcecode

```
package main
```

```
import (
    "fmt"
    "sort"
)

// Fungsi untuk menghitung median
func median(data_2311102024 []int) float64 {
    n := len(data_2311102024)

    if n == 0 {
        return 0
    }

    // Jika jumlah data ganjil, median adalah nilai tengah
    // Jika jumlah data genap, median adalah rata-rata dari dua nilai
    // tengah
    if n%2 == 1 {
        return float64(data_2311102024[n/2])
    }

    return float64(data_2311102024[(n/2)-1]+data_2311102024[n/2]) / 2
}

func main() {
    const sentinel = -5313541 // Penanda akhir input
    var data_2311102024 []int
    var input int

    fmt.Println("Masukkan bilangan (akhiri dengan -5313541):")
}
```

```

// Membaca data
for {
    fmt.Scan(&input)

    if input == sentinel {
        break
    }

    if input == 0 {
        // Jika menemukan 0, hitung median
        sort.Ints(data_2311102024) // Mengurutkan data
        fmt.Printf("Median: %.2f\n", median(data_2311102024))
    } else {
        // Tambahkan bilangan ke dalam array
        data_2311102024 = append(data_2311102024, input)
    }
}
}

```

Output

```

PS D:\Ujian al2> go run "d:\Ujian al2\soal4.go"
Masukkan bilangan (akhiri dengan -5313541):
7 23 11 0 5 19 2 29 3 13 17 0 -5313541
Median: 11.00
Median: 12.00
PS D:\Ujian al2> 

```

5.

Sourcecode

```
package main

import
    "fmt"

const NMAX = 1000000

// Struktur data partai
type Partai struct {
    nama    int
    suara   int
}

// Tipe array untuk menyimpan data partai
type TabPartai_2311102024 [NMAX]Partai

func main() {
    var t TabPartai_2311102024
    var n int // Jumlah partai yang diproses
    var input int
```



```
// Inisialisasi jumlah partai
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 {
        // Jika partai belum ada, tambahkan partai baru
        t[n].nama = input
        t[n].suara = 1
        n++
    } else {
        // Jika partai sudah ada, tambahkan jumlah suaranya
        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_2311102024, 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_2311102024, 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
    }
}
```

Output

```
PS D:\Ujian al2> go run "d:\Ujian al2\soal5.go"
5 1 1 1 1 1 1 1 3 3 3 3 3 2 2 5 5 5 5 4 3 2 2 2 2 -1
1(7) 3(6) 2(6) 5(5) 4(1)
PS D:\Ujian al2> 
```