

TEST ALPRO

Nama : Maulana Ghani Rolanda

NIM : 2311102012

Kelas : IF-11-06

1. SOURCE CODE

```
package main

import "fmt"

type set [2022]int

// Fungsi untuk mengecek apakah elemen sudah ada dalam array
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 mengisi array dengan bilangan unik
func inputSet(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 array
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 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)
}

```

SCREENSHOT OUTPUT

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> go run "c:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro\test1.go"
11 28 33 64 95 16 100 15 64
3 11 7 28 33 6 28
11 28 33
PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro>

```

2. SOURCE CODE

```

//Mahasiswa adalah struct yang berisi NIM, nama, dan array nilai (3 nilai).
//ArrayMahasiswa adalah array statis dengan kapasitas maksimum nMax (51
mahasiswa).

package main

import (
    "fmt"
)

const nMax = 51

type Mahasiswa struct {
    NIM    string

```

```

    Nama string
    Nilai [3]int
}

type ArrayMahasiswa [nMax]Mahasiswa

func main() {
    var dataMahasiswa ArrayMahasiswa
    var n int

    //Input data mahasiswa
    fmt.Print("Masukkan jumlah mahasiswa (maks 51): ")
    fmt.Scan(&n)

    if n > nMax {
        fmt.Println("Jumlah mahasiswa melebihi kapasitas!")
        return
    }

    for i := 0; i < n; i++ {
        fmt.Printf("\nMahasiswa ke-%d:\n", i+1)
        fmt.Print("Masukkan NIM: ")
        fmt.Scan(&dataMahasiswa[i].NIM)
        fmt.Print("Masukkan Nama: ")
        fmt.Scan(&dataMahasiswa[i].Nama)
        fmt.Println("Masukkan 3 nilai mahasiswa:")
        for j := 0; j < 3; j++ {
            fmt.Printf("Nilai %d: ", j+1)
            fmt.Scan(&dataMahasiswa[i].Nilai[j])
        }
    }

    //mencari nilai pertama berdasarkan NIM
    var cariNIM string
    fmt.Print("\nMasukkan NIM untuk mencari nilai pertama: ")
    fmt.Scan(&cariNIM)

    found := false
    for i := 0; i < n; i++ {
        if dataMahasiswa[i].NIM == cariNIM {
            fmt.Printf("Nilai pertama mahasiswa dengan NIM %s adalah %d\n",
cariNIM, dataMahasiswa[i].Nilai[0])
            found = true
            break
        }
    }
    if !found {
        fmt.Println("Mahasiswa dengan NIM tersebut tidak ditemukan.")
    }
}

```

```

    }

    //Mencari nilai terbesar berdasarkan NIM
    fmt.Print("\nMasukkan NIM untuk mencari nilai terbesar: ")
    fmt.Scan(&carinIM)

    found = false
    for i := 0; i < n; i++ {
        if dataMahasiswa[i].NIM == carinIM {
            maks := dataMahasiswa[i].Nilai[0]
            for _, nilai := range dataMahasiswa[i].Nilai {
                if nilai > maks {
                    maks = nilai
                }
            }
            fmt.Printf("Nilai terbesar mahasiswa dengan NIM %s adalah %d\n",
carinIM, maks)
            found = true
            break
        }
    }
    if !found {
        fmt.Println("Mahasiswa dengan NIM tersebut tidak ditemukan.")
    }

    //Tampilkan hasil pencarian
    fmt.Println("\nData Mahasiswa:")
    for i := 0; i < n; i++ {
        fmt.Printf("NIM: %s, Nama: %s, Nilai: %d, %d, %d\n",
            dataMahasiswa[i].NIM,
            dataMahasiswa[i].Nama,
            dataMahasiswa[i].Nilai[0],
            dataMahasiswa[i].Nilai[1],
            dataMahasiswa[i].Nilai[2],
        )
    }
}

```

SCREENSHOT OUTPUT

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> go run "c:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro\test2.go"
Masukkan jumlah mahasiswa (maks 51): 3

Mahasiswa ke-1:
Masukkan NIM: 001
Masukkan Nama: alex
Masukkan 3 nilai mahasiswa:
Nilai 1: 100
Nilai 2: 90
Nilai 3: 70

Mahasiswa ke-2:
Masukkan NIM: 002
Masukkan Nama: ryan
Masukkan 3 nilai mahasiswa:
Nilai 1: 90
Nilai 2: 60
Nilai 3: 50

Mahasiswa ke-3:
Masukkan NIM: 003
Masukkan Nama: davis
Masukkan 3 nilai mahasiswa:
Nilai 1: 0
Nilai 2: 100
Nilai 3: 90

Masukkan NIM untuk mencari nilai pertama: 002
Nilai pertama mahasiswa dengan NIM 002 adalah 90

Masukkan NIM untuk mencari nilai terbesar: 003
Nilai terbesar mahasiswa dengan NIM 003 adalah 100

Data Mahasiswa:
NIM: 001, Nama: alex, Nilai: 100, 90, 70
NIM: 002, Nama: ryan, Nilai: 90, 60, 50
NIM: 003, Nama: davis, Nilai: 0, 100, 90
PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> 

```

3. SOURCE CODE

```

package main

import (
    "fmt"
    "strings"
)

// Maksimum jumlah provinsi
const nProv = 34

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

```

```

// Fungsi untuk menginput data provinsi
func InputData(nama *NamaProv, pop *PopProv, tumbuh *TumbuhProv, jumlah *int)
{
    fmt.Printf("Berapa jumlah provinsi yang ingin diinput? (max %d): ", nProv)
    fmt.Scanln(jumlah)
    if *jumlah > nProv || *jumlah <= 0 {
        fmt.Println("Jumlah provinsi tidak valid. Gunakan nilai antara 1
hingga", nProv)
        return
    }

    for i := 0; i < *jumlah; 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[i])
    }
}

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

// Fungsi untuk mencari indeks provinsi berdasarkan nama
func IndeksProvinsi(nama NamaProv, target string, jumlah int) int {
    for i := 0; i < jumlah; i++ {
        if strings.EqualFold(nama[i], target) {
            return i
        }
    }
    return -1
}

// Prosedur untuk menampilkan prediksi populasi provinsi
func Prediksi(nama NamaProv, pop PopProv, tumbuh TumbuhProv, jumlah int) {
    fmt.Println("Prediksi provinsi dengan pertumbuhan > 2%:")
}

```

```

    for i := 0; i < jumlah; 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
        jumlah  int
        cari     string
    )

    // Input data
    InputData(&nama, &pop, &tumbuh, &jumlah)
    if jumlah <= 0 {
        fmt.Println("Tidak ada data yang diinput. Program selesai.")
        return
    }

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

    // Cari provinsi berdasarkan nama
    fmt.Println("Masukkan nama provinsi yang ingin dicari:")
    fmt.Scanln(&cari)
    indeks := IndeksProvinsi(nama, cari, jumlah)
    if indeks != -1 {
        fmt.Printf("Provinsi %s ditemukan pada indeks %d\n", cari, indeks)
    } else {
        fmt.Printf("Provinsi %s tidak ditemukan\n", cari)
    }

    // Tampilkan prediksi populasi
    Prediksi(nama, pop, tumbuh, jumlah)
}

```

SCREENSHOT OUTPUT

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> go run "c:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro\test3.go"
Berapa jumlah provinsi yang ingin diinput? (max 34): 3
Masukkan nama provinsi ke-1: jawa
Masukkan populasi provinsi jawa: 1000
Masukkan angka pertumbuhan penduduk provinsi jawa: 10
Masukkan nama provinsi ke-2: papua
Masukkan populasi provinsi papua: 100
Masukkan angka pertumbuhan penduduk provinsi papua: 1
Masukkan nama provinsi ke-3: bali
Masukkan populasi provinsi bali: 5000
Masukkan angka pertumbuhan penduduk provinsi bali: 50
Provinsi dengan pertumbuhan tercepat: bali
Masukkan nama provinsi yang ingin dicari:
bali
Provinsi bali ditemukan pada indeks 2
Prediksi provinsi dengan pertumbuhan > 2%:
jawa: Populasi tahun depan = 11000
papua: Populasi tahun depan = 200
bali: Populasi tahun depan = 255000
PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> █

```

4. SOURCE CODE

```

package main

import (
    "fmt"
)

const NMAX = 1000000

type arrInt [NMAX]int

// Fungsi untuk melakukan selection sort
func sorting(T *arrInt, n int) {
    /* I.S. T terdefinisi berisi sejumlah n bilangan bulat
       F.S. Array T terurut secara membesar berdasarkan algoritma selection
       sort */
    for i := 0; i < n-1; i++ {
        minIdx := i
        for j := i + 1; j < n; j++ {
            if T[j] < T[minIdx] {
                minIdx = j
            }
        }
        T[i], T[minIdx] = T[minIdx], T[i]
    }
}

```



```
// Fungsi untuk menghitung median dari array yang sudah terurut
func median(T arrInt, n int) float64 {
    /* I.S. T terdefinisi dan sudah terurut, n adalah jumlah elemen di T
       F.S. Mengembalikan nilai median dari array */
    if n%2 == 1 {

        return float64(T[n/2])
    } else {

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

func main() {
    var data arrInt
    var n int
    var x int

    fmt.Println("Masukkan bilangan (akhiri dengan -5313541):")
    for {
        fmt.Scan(&x)

        // Jika marker -5313541 ditemukan, hentikan program
        if x == -5313541 {
            break
        }

        // Jika data adalah 0, urutkan dan hitung median
        if x == 0 {
            // Urutkan data menggunakan selection sort
            sorting(&data, n)

            // Hitung median
            med := median(data, n)
            fmt.Printf("%.1f\n", med) // Cetak median
        } else {
            // Simpan bilangan positif ke dalam array
            data[n] = x
            n++
        }
    }
}
```

SCREENSHOT OUTPUT

```
PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS
PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> go run "c:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro\test4.go"
Masukkan bilangan (akhiri dengan -5313541):
7 23 11 0 5 19 2 29 3 13 17 0 -5313541
11.0
12.0
PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> z
```

5. SOURCE CODE

```
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 [NMAX]Partai

func main() {
    var t TabPartai
    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, 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
    }
}

```

SCREENSHOT OUTPUT

```

PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> go run "c:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro\test5.go"
1 3 2 3 2 1 1 2 3 4 2 1 1 2 3 1 4 3 2 1 -1
1(7) 2(6) 3(5) 4(2)
PS C:\olan\KULYEAH\LaPrak Alpro 2\Test Alpro> 

```