

Nama : Muhammad DJoko Susilo
NIM : 2311102212
Kelas : 11-06

Soal 1

Source code

```
package main

import "fmt"

type set [2022]int

// Fungsi yang dibuat untuk mengecek apakah input sudah ada atau
// belum di 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 memasukan elemen ke dalam array tanpa duplikasi
func inputSet(T *set, n *int) {
    *n = 0
    for {
        var val int
        fmt.Print("Masukkan angka : ")
        fmt.Scan(&val)
        if exist(*T, *n, val) {
            break
        }
        if !exist(*T, *n, val) {
            (*T)[*n] = val
            *n++
        }
    }
}

// Fungsi untuk mencari irisan antara dua himpunan
func findIntersection(T1, T2 set, n1, n2 int, T3 *set, h *int) {
    *h = 0
    for i := 0; i < n1; i++ {
        if exist(T2, n2, 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) {
    fmt.Print("Hasil Irisan: ")
    for i := 0; i < n; i++ {
        fmt.Print(T[i], " ")
    }
    fmt.Println()
}

// Fungsi utama
```

```

func main() {
    var s1_2311102212, s2_2311102212, s3_2311102212 set
    var n1, n2, n3 int

    fmt.Println("Masukkan elemen untuk himpunan pertama:")
    inputSet(&s1_2311102212, &n1)

    fmt.Println("Masukkan elemen untuk himpunan kedua:")
    inputSet(&s2_2311102212, &n2)

    findIntersection(s1_2311102212, s2_2311102212, n1, n2,
&s3_2311102212, &n3)
    printSet(s3_2311102212, n3)
}

```

Screenshot Program

```

PS D:\code\test> go run "d:\code\test\1\soal1.go"
Masukkan elemen untuk himpunan pertama:
Masukkan angka : 1
Masukkan angka : 1
Masukkan elemen untuk himpunan kedua:
Masukkan angka : 1
Masukkan angka : 1
Hasil Irisan: 1
PS D:\code\test>

```

Soal 2

Sourcecode

```

package main

import "fmt"

const nMax = 51

// Struct mahasiswa
type mahasiswa struct {
    NIM    string
    nama   string
    nilai  int
}

// Array mahasiswa
type arrayMahasiswa [nMax]mahasiswa

// Fungsi yang dibuat untuk memasukan data
func inputData(mhs *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("NIM: ")
            fmt.Scan(&mhs[i].NIM)
            fmt.Print("Nama: ")
            fmt.Scan(&mhs[i].nama)
            fmt.Print("Nilai: ")
            fmt.Scan(&mhs[i].nilai)
        }
    }

// Fungsi untuk mencari nilai pertama mahasiswa berdasarkan NIM
func cariNilaiPertama(mhs arrayMahasiswa, n int, cariNIM string) int
{
    for i := 0; i < n; i++ {
        if mhs[i].NIM == cariNIM {
            return mhs[i].nilai
        }
    }
    return -1 // Jika NIM tidak ditemukan
}

// Fungsi untuk mencari nilai terbesar mahasiswa berdasarkan NIM
func cariNilaiTerbesar(mhs arrayMahasiswa, n int, cariNIM string) int
{
    maxNilai := -1
    for i := 0; i < n; i++ {
        if mhs[i].NIM == cariNIM && mhs[i].nilai > maxNilai {
            maxNilai = mhs[i].nilai
        }
    }
    return maxNilai
}

// Fungsi untuk menampilkan hasil pencarian
func tampilkanHasil(nilai int, pesan string) {
    if nilai == -1 {
        fmt.Println("NIM tidak ditemukan.")
    } else {
        fmt.Printf("%s: %d\n", pesan, nilai)
    }
}

// Fungsi utama
func main() {
    var mhs_2311102212 arrayMahasiswa
    var n int

    // Input data mahasiswa
    inputData(&mhs_2311102212, &n)

    // Pencarian nilai berdasarkan NIM
    var cariNIM string
    fmt.Print("Masukkan NIM yang ingin dicari: ")
    fmt.Scan(&cariNIM)

    // Cari nilai pertama
    nilaiPertama := cariNilaiPertama(mhs_2311102212, n, cariNIM)
    tampilkanHasil(nilaiPertama, "Nilai pertama")

    // Cari nilai terbesar

```

```

        nilaiTerbesar := cariNilaiTerbesar(mhs_2311102212, n, cariNIM)
        tampilkanHasil(nilaiTerbesar, "Nilai terbesar")
    }
}

```

Screenshot Program

Soal 3

Sourcecode

```

package main

import (
    "fmt"
    "strings"
)

const nProv = 34

type Provinsi struct {
    nama      string
    populasi  int
    pertumbuhan float64
}

func inputData(provinsi *[nProv]Provinsi, cariProvinsi *string) {
    for i := 0; i < nProv; i++ {
        fmt.Printf("Masukkan data provinsi ke-%d:\n", i+1)
        fmt.Print("Nama: ")
        fmt.Scan(&provinsi[i].nama)
        fmt.Print("Populasi: ")
        fmt.Scan(&provinsi[i].populasi)
        fmt.Print("Pertumbuhan (%): ")
        fmt.Scan(&provinsi[i].pertumbuhan)
    }
    fmt.Print("Masukkan nama provinsi yang ingin dicari: ")
    fmt.Scan(cariProvinsi)
}

func provinsiTercepat(provinsi [nProv]Provinsi) int {
    idx := 0
    maxPertumbuhan := provinsi[0].pertumbuhan
}

```

```

        for i := 1; i < nProv; i++ {
            if provinsi[i].pertumbuhan > maxPertumbuhan {
                maxPertumbuhan = provinsi[i].pertumbuhan
                idx = i
            }
        }
        return idx
    }

func prediksiPopulasi(provinsi [nProv]Provinsi) {
    fmt.Println("Provinsi dengan pertumbuhan > 2% dan prediksi
populasi tahun depan:")
    for i := 0; i < nProv; i++ {
        if provinsi[i].pertumbuhan > 2 {
            prediksi := float64(provinsi[i].populasi) +
(float64(provinsi[i].populasi) * provinsi[i].pertumbuhan / 100)
            fmt.Printf("%s: %.0f\n", provinsi[i].nama,
prediksi)
        }
    }
}

func indexProvinsi(provinsi [nProv]Provinsi, cari string) int {
    for i := 0; i < nProv; i++ {
        if strings.EqualFold(provinsi[i].nama, cari) {
            return i
        }
    }
    return -1
}

func main() {
    var provinsi_2311102212 [nProv]Provinsi
    var cariProvinsi string

    inputData(&provinsi_2311102212, &cariProvinsi)

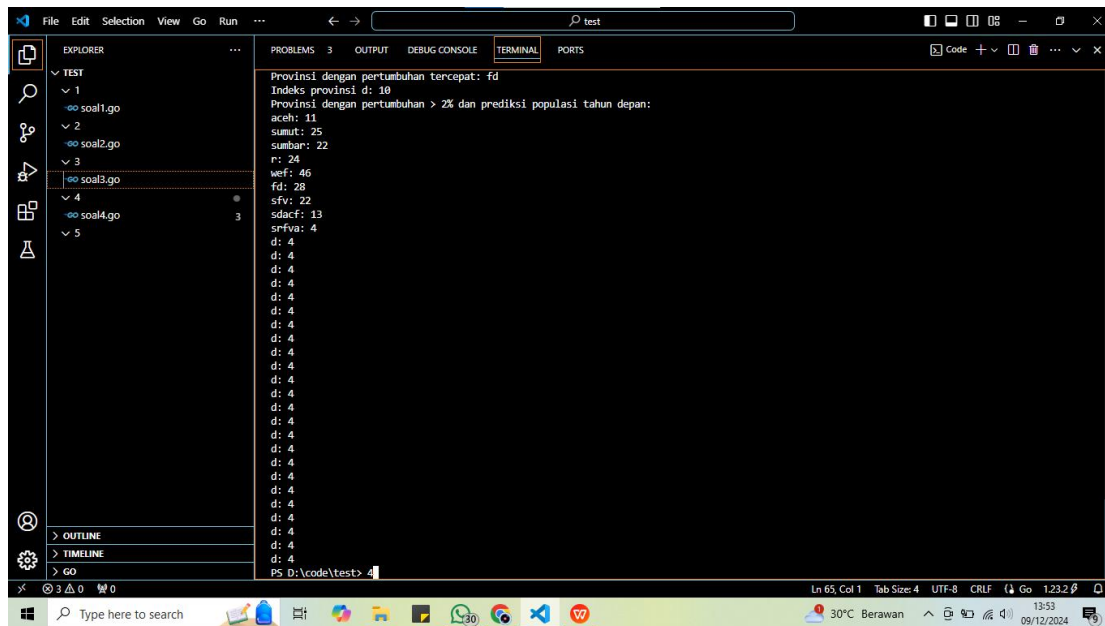
    idxTercepat := provinsiTercepat(provinsi_2311102212)
    fmt.Printf("Provinsi dengan pertumbuhan tercepat: %s\n",
provinsi_2311102212[idxTercepat].nama)

    idxCari := indexProvinsi(provinsi_2311102212, cariProvinsi)
    if idxCari != -1 {
        fmt.Printf("Indeks provinsi %s: %d\n", cariProvinsi,
idxCari)
    } else {
        fmt.Printf("Provinsi %s tidak ditemukan.\n",
cariProvinsi)
    }

    prediksiPopulasi(provinsi_2311102212)
}

```

Screenshot Program



Soal 4

Sourcecode

```
package main

import "fmt"

// Fungsi untuk melakukan selection sort
func sorting(data []int) {
    n := len(data)
    for i := 0; i < n-1; i++ {
        // Temukan indeks elemen terkecil
        idxMin := i
        for j := i + 1; j < n; j++ {
            if data[j] < data[idxMin] {
                idxMin = j
            }
        }
        // Tukar elemen terkecil dengan elemen ke-i
        data[i], data[idxMin] = data[idxMin], data[i]
    }
}

// Fungsi untuk menghitung median
func median(data []int) float64 {
    // Mengurutkan array menggunakan selection sort
    sorting(data)

    // Menentukan median
    n := len(data)
    if n%2 == 1 {
        // Jika jumlah elemen ganjil, median adalah elemen tengah
        return float64(data[n/2])
    } else {
        // Jika jumlah elemen genap, median adalah rata-rata dua
        // elemen tengah
        return float64(data[n/2-1]+data[n/2]) / 2.0
    }
}
```

```

func main() {
    var input int
    var data []int
    var i int
    i = 1

    for {
        fmt.Printf("Masukkan data ke-%d = ", i)
        fmt.Scan(&input)

        // Jika input adalah -5313, hentikan program
        if input == -5313 {
            break
        }

        // Jika input adalah 0, hitung dan cetak median
        if input == 0 {
            if len(data) == 0 {
                fmt.Println("Tidak ada data")
            } else {
                median := calculateMedian(data)
                fmt.Printf("Median saat ini: %.2f\n", median)
            }
        } else {
            // Masukkan angka ke dalam data (selain 0 dan -
            data = append(data, input)
            i++
        }
    }

    fmt.Println("Program dihentikan.")
}

```

Screenshot Program

```

d: 4
d: 4
d: 4
d: 4
d: 4
d: 4
PS D:\code\test> 4go run "d:\code\test\4\asbdf.go"
4go : The term '4go' is not recognized as the name of a cmdlet, function, script file, or operable program. Check the spelling of the name, or
if a path was included, verify that the path is correct and try again.
At line:1 char:1
+ 4go run "d:\code\test\4\asbdf.go"
+ ~~~
+ CategoryInfo          : ObjectNotFound: (4go:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS D:\code\test> go run "d:\code\test\4\tempCodeRunnerFile.go"
# command-line-arguments
4\tempCodeRunnerFile.go:58:15: undefined: calculateMedian
PS D:\code\test> go run "d:\code\test\4\soal4.go"
Masukkan data ke-1 = 23
Masukkan data ke-2 = 12
Masukkan data ke-3 = 54
Masukkan data ke-4 = 23
Masukkan data ke-5 = 0
Median saat ini: 23.00
Masukkan data ke-6 = 324
Masukkan data ke-7 = 23
Masukkan data ke-8 = 1234
Masukkan data ke-9 = 3
Masukkan data ke-10 = 23
Masukkan data ke-11 = 23
Masukkan data ke-12 = 0
Median saat ini: 23.00
Masukkan data ke-13 = -5313
Program dihentikan.
PS D:\code\test>

```

Soal 5

Sourcecode

Screenshot Program

