

Nama : Andika Indra Prastawa

Nim : 2311102033

Kelas : IF – 11 -06

### Soal nomer 1

#### Sourcecode

```
package main

import "fmt"

//mengecek apakah elemen sudah ada dalam slice
func exist_2311102033(T []int, val int) bool {
    for _, v := range T {
        if v == val {
            return true
        }
    }
    return false
}

// Mengisi slice dengan bilangan unik hingga duplikat
ditemukan
func inputSet() []int {
    var T []int
    var val int
    for {
        fmt.Scan(&val)
        if exist_2311102033(T, val) { // berhenti
            jika bilangan sudah ada
                break
        }
    }
}
```

```

        T = append(T, val)

    }

    return T
}

// Mencari irisan dari dua slice
func findIntersection(T1, T2 []int) []int {
    var T3 []int
    for _, v := range T1 {
        if exist_2311102033(T2, v) &&
!exist_2311102033(T3, v) { // Menambahkan ke irisan
jika belum ada
            T3 = append(T3, v)
        }
    }
    return T3
}

// Ini mencetak slice secara horizontal
func printSet(T []int) {
    for i, v := range T {
        if i > 0 {
            fmt.Print(" ")
        }
        fmt.Print(v)
    }
    fmt.Println()
}

func main() {
    fmt.Println("Masukkan himpunan pertama (berhenti
jika duplikat):")
    s1 := inputSet()

    fmt.Println("Masukkan himpunan kedua (berhenti
jika duplikat):")

```

```

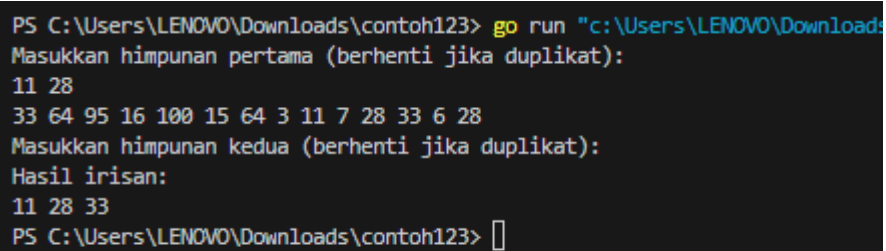
s2 := inputSet()

// Cari irisan kedua himpunan
s3 := findIntersection(s1, s2)

// Cetak hasil irisan
fmt.Println("Hasil irisan:")
printSet(s3)
}

```

### Screenshoot Output



```

PS C:\Users\LENOVO\Downloads\contoh123> go run "c:\Users\LENOVO\Downloads\contoh123.go"
Masukkan himpunan pertama (berhenti jika duplikat):
11 28
33 64 95 16 100 15 64 3 11 7 28 33 6 28
Masukkan himpunan kedua (berhenti jika duplikat):
Hasil irisan:
11 28 33
PS C:\Users\LENOVO\Downloads\contoh123> 

```

### Soal no 2

#### Sourcecode

```

package main

import "fmt"

// Maksimum jumlah data Mahasiswa
const nMax = 51

```

```

// Menyimpan data Mahasiswa
type Mahasiswa_2311102033 struct {
    NIM    string
    nama   string
    nilai  int
}

// Untuk menyimpan data Mahasiswa
type arrayMahasiswa_2311102033 [nMax]Mahasiswa_2311102033

// Menerima input data Mahasiswa
func inputMahasiswa_2311102033(T *arrayMahasiswa_2311102033, 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 mahasiswa: ")
        fmt.Scan(&T[i].NIM)
        fmt.Print("Masukkan nama mahasiswa: ")
        fmt.Scan(&T[i].nama)
        fmt.Print("Masukkan nilai mahasiswa: ")
        fmt.Scan(&T[i].nilai)
    }
}

//nilai pertama Mahasiswa berdasarkan dengan NIM
func carinilaipertama_2311102033(T arrayMahasiswa_2311102033, 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
}

```

```

}

// Mencari nilai terbesar mahasiswa berdasarkan dengan NIM
func cariNilaiTerbesar(T arrayMahasiswa_2311102033, N int, nim
string) int {
    maxNilai := -1
    found := false
    for i := 0; i < N; i++ {
        if T[i].NIM == nim {
            found = true
            if T[i].nilai > maxNilai {
                maxNilai = T[i].nilai
            }
        }
    }
    if found {
        return maxNilai
    }
    return -1 // Jika tidak ditemukan
}

func main() {
    var dataMahasiswa_2311102033 arrayMahasiswa_2311102033
    var jumlahData int
    var nim string

    // Input data Mahasiswa
    inputMahasiswa_2311102033(&dataMahasiswa_2311102033,
&jumlahData)

    // Cari nilai pertama berdasarkan NIM
    fmt.Print("Masukkan NIM untuk mencari nilai mahasiswa
pertama: ")

    fmt.Scan(&nim)

```

```

        nilaiPertama :=
carinilaipertama_2311102033(dataMahasiswa_2311102033, 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)

        }

// Cari nilai terbesar berdasarkan NIM

fmt.Print("Masukkan NIM untuk mencari nilai terbesar: ")

fmt.Scan(&nim)

        nilaiTerbesar := cariNilaiTerbesar(dataMahasiswa_2311102033,
jumlahData, nim)

        if nilaiTerbesar != -1 {

            fmt.Printf("Nilai terbesar untuk NIM %s adalah %d\n",
nim, nilaiTerbesar)

        } else {

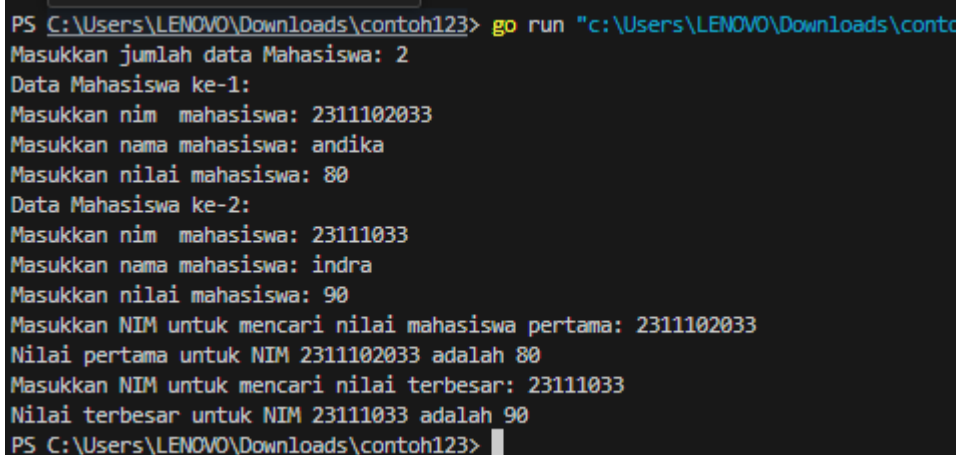
            fmt.Printf("Data dengan NIM %s tidak ditemukan\n", nim)

        }

    }
}

```

### Screenshoot Output



```

PS C:\Users\LENOVO\Downloads\contoh123> go run "c:\Users\LENOVO\Downloads\cont
Masukkan jumlah data Mahasiswa: 2
Data Mahasiswa ke-1:
Masukkan nim mahasiswa: 2311102033
Masukkan nama mahasiswa: andika
Masukkan nilai mahasiswa: 80
Data Mahasiswa ke-2:
Masukkan nim mahasiswa: 23111033
Masukkan nama mahasiswa: indra
Masukkan nilai mahasiswa: 90
Masukkan NIM untuk mencari nilai mahasiswa pertama: 2311102033
Nilai pertama untuk NIM 2311102033 adalah 80
Masukkan NIM untuk mencari nilai terbesar: 23111033
Nilai terbesar untuk NIM 23111033 adalah 90
PS C:\Users\LENOVO\Downloads\contoh123>

```

### Soal nomer 3

#### Sourcecode

```
package main

import (
    "fmt"
    "strings"
)

const nProv = 34

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

// InputData mengisi data nama provinsi,
// populasi, dan angka pertumbuhan

func InputData_2311102033(namaProv *NamaProv,
    pop *PopProv, tumbuh *TumbuhProv) {
    for i := 0; i < nProv; i++ {
        fmt.Printf("Masukkan nama provinsi ke-
%d: ", i+1)

        fmt.Scanln(&(*namaProv)[i])

        fmt.Printf("Masukkan populasi provinsi
%s: ", (*namaProv)[i])

        fmt.Scanln(&(*pop)[i])

        fmt.Printf("Masukkan angka pertumbuhan
provinsi %s: ", (*namaProv)[i])

        fmt.Scanln(&(*tumbuh)[i])
    }
}
```

```

// ProvinsiTercepat mengembalikan indeks
provinsi dengan pertumbuhan tercepat

func ProvinsiTercepat(tumbuh TumbuhProv) int {
    maxIndex := 0
    for i := 1; i < nProv; i++ {
        if tumbuh[i] > tumbuh[maxIndex] {
            maxIndex = i
        }
    }
    return maxIndex
}

// Prediksi menampilkan nama provinsi dengan
prediksi jumlah penduduk tahun depan jika
pertumbuhan > 2%

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

// IndeksProvinsi mencari indeks provinsi
berdasarkan nama

func IndeksProvinsi(namaProv NamaProv, nama
string) int {
    for i := 0; i < nProv; i++ {
        if strings.EqualFold(namaProv[i], nama)
{
            return i
        }
    }
}

```



```

    }

    return -1
}

func main() {
    var namaProv NamaProv
    var pop PopProv
    var tumbuh TumbuhProv

    // Input data
    InputData_2311102033(&namaProv, &pop,
&tumbuh)

    // Provinsi dengan pertumbuhan tercepat
    tercepatIndex := ProvinsiTercepat(tumbuh)

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

    // Indeks provinsi berdasarkan nama
    var cariNama string

    fmt.Print("Masukkan nama provinsi yang ingin
dicari: ")

    fmt.Scanln(&cariNama)

    indeks := IndeksProvinsi(namaProv, cariNama)

    if indeks != -1 {
        fmt.Printf("Indeks provinsi %s adalah
%d\n", cariNama, indeks)
    } else {
        fmt.Printf("Provinsi %s tidak
ditemukan\n", cariNama)
    }

    // Prediksi populasi tahun depan
    Prediksi(namaProv, pop, tumbuh)
}

```

## Screenshoot Output

```
PS C:\Users\LENOVO\Downloads\contoh123> go run "c:\Users\LENOVO\Downl
Masukkan nama provinsi ke-1: jawa Tengah
Masukkan populasi provinsi Jawa Tengah: 200
Masukkan angka pertumbuhan provinsi Jawa Tengah: 20
Masukkan nama provinsi ke-2: Jawa Barat
Masukkan populasi provinsi Jawa Barat: 400
Masukkan angka pertumbuhan provinsi Jawa Barat: 10
Masukkan nama provinsi ke-3: Jawa Timur
Masukkan populasi provinsi Jawa Timur: 500
Masukkan angka pertumbuhan provinsi Jawa Timur: 5
Masukkan nama provinsi ke-4: Banten
Masukkan populasi provinsi Banten: 200
Masukkan angka pertumbuhan provinsi Banten: 2
Masukkan nama provinsi ke-5: Kalimantan
Masukkan populasi provinsi Kalimantan: 100
```

#### Soal nomer 4

#### Sourcecode

```
package main

import (
    "fmt"
    "sort"
)

// Fungsi menghitung median
func median(Data_2311102033 []int) float64 {
    n := len(Data_2311102033)
    if n == 0 {
        return 0
    }
    // Jika jumlah Data ganjil, Maka median adalah
    // nilai tengah
    // Jika jumlah Data genap, Maka median adalah rata-
    // rata dari dua nilai tengah
    if n%2 == 1 {
        return float64(Data_2311102033[n/2])
    }
    return float64(Data_2311102033[(n/2)-
1]+Data_2311102033[n/2]) / 2
}

func main() {
    const sentinel = -5313541 //akhir input
    var Data_2311102033 []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, menghitung median
            sort.Ints(Data_2311102033) // Mengurutkan
            fmt.Printf("Median: %.2f\n",
median(Data_2311102033))
        } else {
            // Tambahkan bilangan ke dalam array
            Data_2311102033 = append(Data_2311102033,
input)
        }
    }
}

```

### Screenshot Output

```

PS C:\Users\LENOVO\Downloads\contoh123> go run "c:\Users\LENOVO\Downloads\contoh
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 C:\Users\LENOVO\Downloads\contoh123> go run "c:\Users\LENOVO\Downloads\contoh
Masukkan bilangan (akhiri dengan -5313541):
23 12 24 26 20 10 25 8 0 -5313541
Median: 21.50
PS C:\Users\LENOVO\Downloads\contoh123>

```

### Soal nomer 5

```
package main

import (
    "fmt"
    "sort"
)

// Struct untuk menyimpan data partai
type Partai_2311102033 struct {
    Nama    int
    Suara   int
}

// Fungsi utama
func main() {
    var (
        input    int
        partaiMap = make(map[int]int) // Map untuk
        menyimpan jumlah suara per partai
    )

    fmt.Println("Masukkan angka partai yang dipilih
    (akhiri dengan -1):")

    // Membaca input hingga -1
    for {
        fmt.Scan(&input)
        if input == -1 {
            break
        }
        partaiMap[input]++
    }

    // Menyimpan data ke dalam slice untuk diurutkan
```

```

var partaiList []Partai_2311102033

for nama, suara := range partaiMap {
    partaiList = append(partaiList,
Partai_2311102033{Nama: nama, Suara: suara})
}

// Mengurutkan slice berdasarkan jumlah suara
secara descending

sort.Slice(partaiList, func(i, j int) bool {
    if partaiList[i].Suara ==
partaiList[j].Suara {
        return partaiList[i].Nama <
partaiList[j].Nama // Jika suara sama, urutkan
berdasarkan nama partai
    }

    return partaiList[i].Suara >
partaiList[j].Suara
}))

// Menampilkan hasil

fmt.Println("Hasil perolehan suara:")

for _, p := range partaiList {
    fmt.Printf("%d(%d) ", p.Nama, p.Suara)
}

fmt.Println()
}

```

### Screenshoot Output

```

PS C:\Users\LENOVO\Downloads\contoh123> go run "c:\Users\LENOVO\Downlo
Masukkan angka partai yang dipilih (akhiri dengan -1):
8 8 8 8 8 8 8 8 8 8 8 8 8 8 -1
Hasil perolehan suara:
8(15)
PS C:\Users\LENOVO\Downloads\contoh123>

```