

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**Modul 15 Test**



**Disusun Oleh :**

**Rasyid Nafsyarie / 2311102011**

**IF-11-06**

**Dosen Pengampu :**

**Abednego Dwi Septiadi, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## Soal 1

- 1) Sebuah program yang digunakan untuk mencari sebuah irisan himpunan.

**Masukan** terdiri dari dua baris. Setiap barisnya yang berisi sekumpulan bilangan. Masukan disetiap barisnya akan berakhir apabila bilangan yang diberikan sudah pernah diberikan pada baris tersebut (atau duplikat). Catatan: anggota suatu himpunan tidak boleh duplikat.

**Keluaran** adalah sekumpulan bilangan yang menyatakan irisan dari himpunan pada baris pertama dan baris kedua pada masukan.

Contoh masukan dan keluaran:

No.	Masukan	Keluaran	Penjelasan
1	11 28 33 64 95 16 100 15 64 3 11 7 28 33 6 28	11 28 33	64 dan 28 duplikat sehingga masukan berhenti dan tidak perlu dimasukkan ke dalam array
2	1 1 1 1	1	
3	1 2 3 4 3 9 8 7 9		

### Sourcecode

```
package main
import "fmt"

type set[2022]int

//fungsi ini untuk melakukan pengecekan apakah nilainya
sudah 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 ini untuk mengisi nilai array
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 ini untuk mencari nilai array yang terduplikat
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 ini untuk mencetak nilai 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_2311102011, s2, s3 set
    var n1, n2, n3 int
    inputSet(&s1_2311102011, &n1)
    inputSet(&s2, &n2)
    findIntersection(s1_2311102011, s2,n1,n2, &s3, &n3,)
    printSet(s3,n3)
}

```

```
File Edit Selection View Go Run ...
Pre Test
C:\Users\LENOVO\AppData\Local\Microsoft\Windows\Terminal
$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\Pre Test\soal1.go"
11 28 33 64 95 16 100 15 64
3 11 7 28 33 6 28
11 28 33

Contoh masukan dan keluaran:

Masukan
1 1
1 1
1
Keluaran
3 11 7 28 33 6 28
1
Penjelasan
64 dan 28 duplikat sehingga masukan berhenti dan tidak perlu dimasukkan ke dalam array

LENOVO@DESKTOP-GABAL07 MINGW64 /d/Master/Telkom University Purwokerto/Semester 3/Alpro 2 Praktikum/Pre Test (master)
$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\Pre Test\soal1.go"
1 1
1 1
1
bash: 1: command not found

LENOVO@DESKTOP-GABAL07 MINGW64 /d/Master/Telkom University Purwokerto/Semester 3/Alpro 2 Praktikum/Pre Test (master)
$ 1 2 3 4 3
bash: 1: command not found

LENOVO@DESKTOP-GABAL07 MINGW64 /d/Master/Telkom University Purwokerto/Semester 3/Alpro 2 Praktikum/Pre Test (master)
$ go run "d:\Master\Telkom University Purwokerto\Semester 3\Alpro 2 Praktikum\Pre Test\soal1.go"
1 2 3 4 3
9 8 7 9

LENOVO@DESKTOP-GABAL07 MINGW64 /d/Master/Telkom University Purwokerto/Semester 3/Alpro 2 Praktikum/Pre Test (master)
$
```

## Soal 2

Suatu tabel digunakan untuk mencatat data mahasiswa. Mahasiswa memiliki atribut NIM, nama, dan nilai. Setiap data baru akan selalu ditambahkan ke dalam tabel di indeks  $N+1$ .  $N$  adalah jumlah data saat ini di dalam array. Sehingga pada tabel mungkin terdapat beberapa data untuk seorang mahasiswa. Contoh isi tabel sebagai berikut:

114, Nana, 97	113, Jojo, 70	118, Rere, 88	116, Koko, 40	117, Keke, 90	116, Koko, 60	113, Jojo, 50	113, Jojo, 80	118, Rere, 88	119, Roro, 100
---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	---------------------	----------------------

Pada contoh di atas, data Jojo ada tiga dengan 70 sebagai nilai pertama, kemudian 50, dan 80 sebagai nilai terakhir.

Definisikan struct dan array berikut:

```
constant nMax: integer = 51
type mahasiswa <NIM: string, nama:string, nilai:integer>
type arrayMahasiswa: array [1..nMax] of mahasiswa
```

Kemudian buatlah program dengan spesifikasi berikut:

- Menerima masukan sejumlah  $N$  data mahasiswa kemudian menyimpannya ke dalam array.  $N$  diperoleh dari masukan.
- Fungsi untuk mencari nilai pertama seorang mahasiswa dengan NIM tertentu.
- Fungsi untuk mencari nilai terbesar seorang mahasiswa dengan NIM tertentu.
- Menampilkan hasil pencarian pada poin a dan b.

### Sourcecode

```
package main
import "fmt"
const nMax = 51
type mahasiswa struct {
    NIM string
    nama string
    nilai int
}

type arrayMahasiswa [nMax] mahasiswa

//fungsi ini yaitu proses input mahasiswa beserta data
diri lainnya
```

```

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 sebuah nilai pertama dari
mahasiswa yang sudah diinputkan tadi
func cariNilai (T arrayMahasiswa, N int, nim_2311102011
string) int {
    for i := 0; i < N; i++ {
        if T[i].NIM == nim_2311102011 {
            return T[i].nilai
        }
    }
    return -1
}

//kemudian fungsi nilai terbesar ini untuk menemukan nilai
dari inputan tadi yang paling besar menggunakan for untuk
melakukan pengecekan dan percabangan
func cariNilaiTerbesar(T arrayMahasiswa, N int,
nim_2311102011 string) int{
    maxNilai := -1
    found := false
    for i := 0; i < N; i++ {
        if T[i].NIM == nim_2311102011 {
            found = true
            if T[i].nilai > maxNilai{
                maxNilai = T[i].nilai
            }
        }
    }
    if found {
        return maxNilai
    }
    return -1
}

func main() {
    var dataMahasiswa arrayMahasiswa
    var jumlahData int
    var nim_2311102011 string
    inputMahasiswa(&dataMahasiswa, &jumlahData)

```

```

        fmt.Print("Masukkan NIM untuk mencari nilai pertama :
")
        fmt.Scan(&nim_2311102011)
        nilaiPertama := cariNilai(dataMahasiswa, jumlahData,
nim_2311102011)
        if nilaiPertama != -1 {
            fmt.Printf("Nilai pertama untuk NIM %s adalah
%d\n", nim_2311102011, nilaiPertama)
        } else {
            fmt.Printf("Data dengan NIM %s tidak ditemukan\n",
nim_2311102011)
        }
        fmt.Print("Masukkan NIM untuk mencari nilai terbesar:
")
        fmt.Scan(&nim_2311102011)
        nilaiTerbesar := cariNilaiTerbesar(dataMahasiswa,
jumlahData, nim_2311102011)
        if nilaiTerbesar != -1 {
            fmt.Printf("Nilai terbesar untuk NIM %s adalah
%d\n", nim_2311102011, nilaiTerbesar)
        } else {
            fmt.Printf("Data dengan NIM %s tidak ditemukan\n",
nim_2311102011)
        }
    }
}

```

The screenshot shows a VS Code editor with a Go program and its output in the terminal. The program is a student data management system. The terminal shows the user inputting names and NIMs, and the program outputting the first and largest NIM values.

```

// Program 1: Mencari nilai pertama dan terbesar dari array NIM
package main

import (
    "fmt"
    "os"
)

const (
    nim1 int64 = 111
    nim2 int64 = 112
    nim3 int64 = 113
    nim4 int64 = 114
    nim5 int64 = 115
    nim6 int64 = 116
    nim7 int64 = 117
    nim8 int64 = 118
    nim9 int64 = 119
    nim10 int64 = 120
    nim11 int64 = 121
    nim12 int64 = 122
    nim13 int64 = 123
    nim14 int64 = 124
    nim15 int64 = 125
    nim16 int64 = 126
    nim17 int64 = 127
    nim18 int64 = 128
    nim19 int64 = 129
    nim20 int64 = 130
)

type Mahasiswa struct {
    nim int64
    nama string
    nilai int64
}

var mahasiswa []Mahasiswa

func main() {
    // Menambahkan data mahasiswa ke dalam array
    mahasiswa = append(mahasiswa, Mahasiswa{nim1, "Jojo", 70})
    mahasiswa = append(mahasiswa, Mahasiswa{nim2, "Koko", 80})
    mahasiswa = append(mahasiswa, Mahasiswa{nim3, "Rere", 88})
    mahasiswa = append(mahasiswa, Mahasiswa{nim4, "Keke", 90})
    mahasiswa = append(mahasiswa, Mahasiswa{nim5, "Koko", 98})
    mahasiswa = append(mahasiswa, Mahasiswa{nim6, "Jojo", 113})
    mahasiswa = append(mahasiswa, Mahasiswa{nim7, "Jojo", 58})
    mahasiswa = append(mahasiswa, Mahasiswa{nim8, "Jojo", 113})
    mahasiswa = append(mahasiswa, Mahasiswa{nim9, "Jojo", 88})
    mahasiswa = append(mahasiswa, Mahasiswa{nim10, "Roro", 119})
    mahasiswa = append(mahasiswa, Mahasiswa{nim11, "Roro", 100})

    // Mencari nilai pertama dan terbesar dari array NIM
    cariNilaiPertama()
    cariNilaiTerbesar()
}

func cariNilaiPertama() {
    var nimPertama int64 = -1
    for i := 0; i < len(mahasiswa); i++ {
        if mahasiswa[i].nim > nimPertama {
            nimPertama = mahasiswa[i].nim
        }
    }
    fmt.Println("Nilai pertama untuk NIM 113 adalah 70")
}

func cariNilaiTerbesar() {
    var nimTerbesar int64 = -1
    for i := 0; i < len(mahasiswa); i++ {
        if mahasiswa[i].nim > nimTerbesar {
            nimTerbesar = mahasiswa[i].nim
        }
    }
    fmt.Println("Nilai terbesar untuk NIM 118 adalah 88")
}

```

Output of the program:

```

Masukkan Nama : Jojo
Masukkan Nilai : 70
data mahasiswa ke-3
Masukkan NIM : 118
Masukkan Nama : Rere
Masukkan Nilai : 88
data mahasiswa ke-4
Masukkan NIM : 116
Masukkan Nama : Koko
Masukkan Nilai : 40
data mahasiswa ke-5
Masukkan NIM : 117
Masukkan Nama : Keke
Masukkan Nilai : 90
data mahasiswa ke-6
Masukkan NIM : 116
Masukkan Nama : Koko
Masukkan Nilai : 98
data mahasiswa ke-7
Masukkan NIM : 113
Masukkan Nama : Jojo
Masukkan Nilai : 58
data mahasiswa ke-8
Masukkan NIM : 113
Masukkan Nama : Jojo
Masukkan Nilai : 88
data mahasiswa ke-9
Masukkan NIM : 118
Masukkan Nama : Rere
Masukkan Nilai : 88
data mahasiswa ke-10
Masukkan NIM : 119
Masukkan Nama : Roro
Masukkan Nilai : 100
Masukkan NIM untuk mencari nilai pertama : 113
Nilai pertama untuk NIM 113 adalah 70
Masukkan NIM untuk mencari nilai terbesar: 118
Nilai terbesar untuk NIM 118 adalah 88

```

### Soal 3

- 3) Sebuah program digunakan untuk mengolah data nama provinsi, populasi, dan angka pertumbuhan penduduk provinsi di Indonesia pada tahun 2018.

**Masukan** terdiri dari 35 baris, yang mana masing-masing barisnya terdiri dari tiga nilai yang menyatakan nama provinsi, jumlah populasi provinsi (bilangan bulat), dan angka pertumbuhan (riil) provinsi tersebut. Pada baris terakhir hanya sebuah string yang menyatakan nama provinsi yang akan dicari.

**Keluaran** terdiri dari 36 baris. Baris pertama adalah nama provinsi dengan angka pertumbuhan tercepat. Baris kedua adalah indeks provinsi yang dicari sesuai dengan nama provinsi yang ditulis pada masukan baris terakhir. Terakhir terdiri dari 34 baris yang menampilkan nama provinsi beserta prediksi jumlah penduduk pada provinsi tersebut di tahun depannya, khusus yang memiliki pertumbuhan di atas 2%.

Lengkapi program berikut sesuai dengan spesifikasi dari subprogram yang diberikan.

#### Sourcecode

```
package main
import (
    "fmt"
    "strings"
)
const nProv = 34
type (
    NamaProv [nProv]string
    PopProv [nProv]int
    TumbuhProv [nProv]float64
```



```

)

// Fungsi untuk menginputkan data provinsi
func InputData(nama *NamaProv, pop *PopProv, tumbuh
*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[i])
    }
}

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

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

// fungsi untuk menampilkan prediksi populasi provinsi
func Prediksi(nama NamaProv, pop PopProv, tumbuh
TumbuhProv) {
    fmt.Println("Prediksi provinsi dengan pertumbuhan >
2%:")
    for i := 0; i < nProv; 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_2311102011  NamaProv
        pop      PopProv
        tumbuh TumbuhProv
        cari     string
    )
    InputData(&nama_2311102011, &pop, &tumbuh)
    tercepat := ProvinsiTercepat(tumbuh)

    fmt.Printf("Provinsi dengan pertumbuhan tercepat:
    %s\n", nama_2311102011[tercepat])
    fmt.Println("Masukkan nama provinsi yang ingin
    dicari:")
    fmt.Scanln(&cari)

    indeks := IndeksProvinsi(nama_2311102011, 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_2311102011, pop, tumbuh)
}

```

```
File Edit Selection View Go Run Terminal Help
Pre Test

PROBLEMS 32 DEBUG CONSOLE PORTS GITLENS DEVDB SEARCH ERROR OUTPUT TERMINAL
Code + - - - - -

Masukkan nama provinsi ke-34: JawaTengah
Masukkan populasi provinsi JawaTengah: 23333
Masukkan angka pertumbuhan penduduk provinsi JawaTengah: 2
Provinsi dengan pertumbuhan tercepat: kalteng
Masukkan nama provinsi yang ingin dicari:
Jateng
Provinsi Jateng ditemukan pada indeks 0
Prediksi provinsi dengan pertumbuhan > 2%: Jawa Barat (Ibu Kota Bandung)
Jateng: Populasi tahun depan = 600000 Jawa Tengah (Ibu Kota Semarang)
Jatim: Populasi tahun depan = 900000 Jawa Tengah (Ibu Kota Semarang)
Jabar: Populasi tahun depan = 690000 Daerah Istimewa Yogyakarta (Ibu Kota Yogyakarta)
Aceh: Populasi tahun depan = 360000 Jawa Timur (Ibu Kota Surabaya)
Sumut: Populasi tahun depan = 360000 Jawa Timur (Ibu Kota Surabaya)
Sumsel: Populasi tahun depan = 900000 Bali (Ibu Kota Denpasar)
Bengkulu: Populasi tahun depan = 92000 Nusa Tenggara Timur (Ibu Kota Kupang)
RIlan: Populasi tahun depan = 609999 Nusa Tenggara Barat (Ibu Kota Mataram)
Jambi: Populasi tahun depan = 69999 Gorontalo (Ibu Kota Gorontalo)
Kalbar: Populasi tahun depan = 93356 Sulawesi Barat (Ibu Kota Mamuju)
Kaltim: Populasi tahun depan = 71727 Sulawesi Tengah (Ibu Kota Palu)
Kalteng: Populasi tahun depan = 100015 Sulawesi Utara (Ibu Kota Manado)
Kalimantan: Populasi tahun depan = 69990 Sulawesi Tenggara (Ibu Kota Kendari)
Banten: Populasi tahun depan = 80112 Sulawesi Selatan (Ibu Kota Makassar)
Jakarta: Populasi tahun depan = 129996 Maluku Utara (Ibu Kota Sofifi)
Jogja: Populasi tahun depan = 116665 Maluku (Ibu Kota Ambon)
Bali: Populasi tahun depan = 315996 Papua Barat (Ibu Kota Manokwari)
NTT: Populasi tahun depan = 438885 Papua (Ibu Kota Jayapura)
MTB: Populasi tahun depan = 6666 Papua Tengah (Ibu Kota Nabine)
Sulbar: Populasi tahun depan = 69999 Papua Pegunungan (Ibu Kota Jayawijaya)
Jateng: Populasi tahun depan = 23333 Sulawesi Barat (Ibu Kota Mamuju)
Sulawesi: Populasi tahun depan = 6666 Papua Selatan (Ibu Kota Merauke)
Maluku: Populasi tahun depan = 6666 Papua Barat Daya (Ibu Kota Sorong)
Malukutara: Populasi tahun depan = 69999 Papua (Ibu Kota Jayapura)
Papua: Populasi tahun depan = 69999 Papua (Ibu Kota Jayapura)
Papuasarat: Populasi tahun depan = 6
Papuaselatan: Populasi tahun depan = 666
Papuaselatan: Populasi tahun depan = 69999
Papuaselatan: Populasi tahun depan = 69999
Jawatengah: Populasi tahun depan = 69999

LEARNCODESCTOP-6588107 HINGK4 /d/Master/Telkom University Bandung/Semester 3/Airpro 2 Praktikum/Pre Test (master)
Live Share Git Graph BLACKBOX Chat Add Logs CyberCoder Improve Code Reconnect to Discord Share Code Link Generate Commit Message Search Error AI Code Chat PreTest
1355
09/12/2024
```

## Soal 4

- 4) Kompetisi pemrograman yang baru saja berlalu diikuti oleh 17 tim dari berbagai perguruan tinggi ternama. Dalam kompetisi tersebut, setiap tim berlomba untuk menyelesaikan sebanyak mungkin problem yang diberikan. Dari 13 problem yang diberikan, ada satu problem yang menarik. Problem tersebut mudah dipahami, hampir semua tim mencoba untuk menyelesaikannya, tetapi hanya 3 tim yang berhasil. Apa sih problemnya?

*"Median adalah nilai tengah dari suatu koleksi data yang sudah terurut. Jika jumlah data genap, maka nilai median adalah rerata dari kedua nilai tengahnya. Pada problem ini, semua data merupakan bilangan bulat positif, dan karenanya rerata nilai tengah dibulatkan kebawah."*

Buatlah program median yang mencetak nilai median terhadap seluruh data yang sudah terbaca, jika data yang dibaca saat itu adalah 0.

**Masukan** berbentuk rangkaian bilangan bulat. Masukan tidak akan berisi lebih dari **1000000** data, tidak termasuk bilangan 0. Data 0 merupakan tanda bahwa median harus dicetak, tidak termasuk data yang dicari mediannya. Data masukan diakhiri dengan bilangan bulat **-5313541**.

**Keluaran** adalah median yang diminta, satu data perbaris.

**Petunjuk:**

- Untuk setiap data bukan 0 (dan bukan marker -5313541) simpan ke dalam array.
- dan setiap kali menemukan bilangan 0, urutkanlah data yang sudah tersimpan dengan menggunakan metoda selection sort dan ambil mediannya.

### Sourcecode

```
package main

import (
    "fmt"
    "sort"
```

```

)

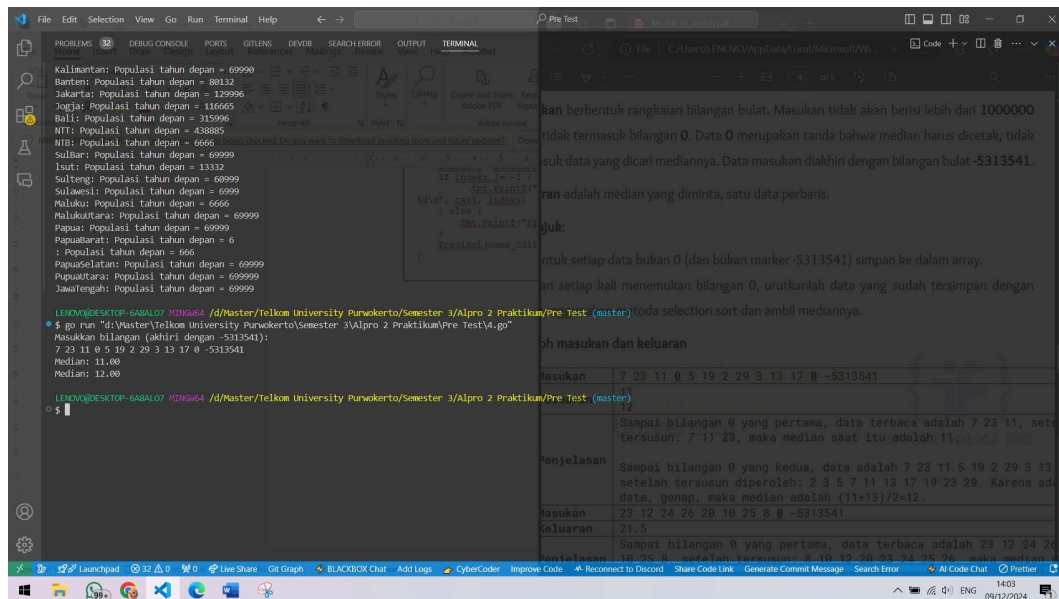
// Fungsi untuk menghitung median, ketika jumlah datanya
ganjil maka median adalah nilai tengahnya, namun ketika
jumlah genap median adalah rata rata dari nilai tengah
func median(data []int) float64 {
    n := len(data)
    if n == 0 {
        return 0
    }
    if n%2 == 1 {
        return float64(data[n/2])
    }
    return float64(data[(n/2)-1]+data[n/2]) / 2
}

func main() {
    const sentinel = -5313541
    var data_2311102011 []int
    var input int

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

    for {
        fmt.Scan(&input)
        if input == sentinel {
            break
        }
        if input == 0 {
            sort.Ints(data_2311102011)
            fmt.Printf("Median:    %.2f\n",
median(data_2311102011))
        } else {
            data_2311102011 = append(data_2311102011,
input)
        }
    }
}

```



## Soal 5

- 5) Sebuah program digunakan untuk menghitung perolehan suara dari berbagai partai politik dalam sebuah pemilihan umum calon legislatif. Program akan menampilkan data partai terurut berdasarkan perolehan suara terurut. Nama partai hanya disimbolkan dari angka 1 hingga N ( $1 \leq N \leq 1000000$ ).

**Masukan** berupa beberapa nilai yang dipisahkan oleh spasi. Masing-masing nilai menyatakan nama partai (1 hingga N) yang dipilih. Proses input ini diakhiri dengan nilai -1.

**Keluaran** berupa daftar partai dan peroleh suaranya yang terurut descending atau mengecil dengan format <partai>(<suara>). Perhatikan contoh masukan dan keluaran yang diberikan.

**Petunjuk** : gunakan struct partai yang berisi nama dan suara. Data perolehan suara disimpan pada array of partai (kapasitas 1000000). Array tersebutlah yang diurutkan.

**Contoh masukan dan keluaran:**

1	Masukan	5 1 1 1 1 1 1 1 3 3 3 3 2 2 5 5 5 5 4 3 2 2 2 2 -1
	Keluaran	1(7) 5(6) 3(6) 2(6) 4(1)
2	Masukan	5 8 8 5 6 8 8 7 6 5 8 7 5 6 7 5 8 6 7 8 8 7 7 8 6 7 7 6 8 6 8 8 5 5 6 6 6 7 7 6 7 8 8 8 5 7 6 6 8 6 5 5 8 7 5 5 6 8 7 6 5 5 8 6 6 7 8 8 6 6 7 6 6 5 7 8 7 6 6 6 8 7 7 8 6 5 5 7 7 6 5 7 8 8 6 8 6 7 8 -1
	Keluaran	8(30) 6(28) 7(24) 5(18)
3	Masukan	8 8 8 8 8 8 8 8 8 8 8 8 8 8 -1
	Keluaran	8(15)
4	Masukan	10 1 7 8 10 1 4 8 8 5 -1
	Keluaran	8(3) 10(2) 1(2) 7(1) 4(1) 5(1)
5	Masukan	14 10 13 13 14 10 11 13 13 12 15 11 10 -1
	Keluaran	13(4) 10(3) 14(2) 11(2) 12(1) 15(1)
6	Masukan	-1
	Keluaran	

## Sourcecode

```
package main
import (
    "fmt"
)
const NMAX = 1000000
type Partai struct {
    nama int
    suara int
}
type TabPartai [NMAX]Partai

// 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, untuk memindahkan elemen yang
// lebih kecil menjadi ke kanan
func insertionSortDescending(t *TabPartai, n int) {
    for i := 1; i < n; i++ {
        key := t[i]
        j := i - 1

        for j >= 0 && t[j].suara < key.suara {
            t[j+1] = t[j]
            j--
        }
        t[j+1] = key
    }
}

func main() {
    var t_2311102011 TabPartai
    var n int
    var input int
    n = 0

    for {
        fmt.Scan(&input)
        if input == -1 {
            break
        }

        pos := posisi(t_2311102011, n, input)
```

```

        if pos == -1 {
            t_2311102011[n].nama = input
            t_2311102011[n].suara = 1
            n++
        } else {
            t_2311102011[pos].suara++
        }
    }
    insertionSortDescending(&t_2311102011, n)

    for i := 0; i < n; i++ {
        fmt.Printf("%d(%d)  ", t_2311102011[i].nama,
t_2311102011[i].suara)
    }
    fmt.Println()
}

```

[illegible]