

MODUL 15
ALGORITMA PEMOGRAMAN
Ujian praktikum



Disusun Oleh :

Didik Setiawan/2311102030

IF-11-06

Dosen Pengampu :

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

I. Studi kasus

- 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		

```
package main
```

```
import (  
    "fmt"  
    "strconv"  
    "strings"  
)
```

```
// fungsi untuk menginputkan dan memeriksa inputan
```

```
func inputSet() map[int]struct{} {  
    var input string  
    fmt.Scanln(&input)  
    set := make(map[int]struct{})  
    for _, val := range strings.Fields(input) {  
        if num, err := strconv.Atoi(val); err == nil {  
            set[num] = struct{}{}  
        }  
    }  
    return set  
}
```

```
// mencari inputan
```

```

func findIntersection(set2311102030, set2 map[int]struct{}) []int {
    var intersection []int
    for num := range set2311102030 {
        if _, exists := set2[num]; exists {
            intersection = append(intersection, num)
        }
    }
    return intersection
}

func main() {
    //pengguna meninputkan
    fmt.Println("Masukkan himpunan pertama:")
    set2311102030 := inputSet()

    fmt.Println("Masukkan himpunan kedua:")
    set2 := inputSet()
    //mencetak
    fmt.Println("Irisan himpunan:", findIntersection(set2311102030,
set2))
}

```

Output

```

PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> go run "D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15\soal1.go"4
Masukkan himpunan pertama:
1 1
Masukkan himpunan kedua:
Irisan himpunan: []
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15>

```

II. Studi kasus

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
```

```
package main

import (
    "fmt"
    "strconv"
    "strings"
)

const nMax = 51

// Definisi struct mahasiswa
type mahasiswa struct {
    NIM string
    nama_2311102030 string
    nilai int
}

// Array untuk menyimpan data mahasiswa
type arrayMahasiswa [nMax]mahasiswa

// Fungsi untuk membaca data mahasiswa dari input
func inputMahasiswa(n int) arrayMahasiswa {
    var arr arrayMahasiswa

    for i := 0; i < n; i++ {
```

```

        fmt.Printf("Masukkan data mahasiswa ke-%d (NIM Nama Nilai):", i+1)

        data := strings.Fields()

        if len(data) < 3 {
            fmt.Println("Input tidak valid. Masukkan NIM, nama, dan nilai.")
            i--
            continue
        }

        nilai, err := strconv.Atoi(data[2])
        if err != nil {
            fmt.Println("Nilai harus berupa angka.")
            i--
            continue
        }

        arr[i] = mahasiswa{
            NIM: data[0],
            nama_2311102030: data[1],
            nilai: nilai,
        }
    }
    return arr
}

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

// Fungsi untuk mencari nilai tertinggi
func nilaiTertinggi(arr arrayMahasiswa, n int) mahasiswa {
    max := arr[0]
    for i := 1; i < n; i++ {
        if arr[i].nilai > max.nilai {
            max = arr[i]
        }
    }
}

```

```

    return max
}

func main() {
    var n int
    fmt.Print("Masukkan jumlah mahasiswa: ")
    fmt.Scan(&n)

    if n > nMax {
        fmt.Printf("Jumlah mahasiswa tidak boleh lebih dari %d.\n",
nMax)
        return
    }

    // Input data mahasiswa
    mahasiswaArray := inputMahasiswa(n)

    // Mencari nilai berdasarkan NIM
    var nim string
    fmt.Print("Masukkan NIM untuk mencari nilai: ")
    fmt.Scan(&nim)
    nilai := cariNilai(mahasiswaArray, n, nim)
    if nilai != -1 {
        fmt.Printf("Nilai mahasiswa dengan NIM %s adalah %d.\n", nim,
nilai)
    } else {
        fmt.Printf("Mahasiswa dengan NIM %s tidak ditemukan.\n", nim)
    }

    // Mencari mahasiswa dengan nilai tertinggi
    mahasiswaMax := nilaiTertinggi(mahasiswaArray, n)
    fmt.Printf("Mahasiswa dengan nilai tertinggi adalah %s (%s) dengan
nilai %d.\n",
        mahasiswaMax.nama_2311102030, mahasiswaMax.NIM,
mahasiswaMax.nilai)
}

```

Output

```

PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> go run "D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15\soal2.go"4
# command-line-arguments
.\soal2.go:28:11: not enough arguments in call to strings.Fields
      have ()
      want (string)
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> █

```

III. Studi kasus

- 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 (rill) 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%.

Terdapat program berikut sesuai dengan spesifikasi dari eduprogram yang diberikan

```
package main

import "fmt"

// Definisi struct propinsi dengan atribut nama_propinsi, nim, kelas,
// jurusan, dan jumlah_penduduk_2311102030
type propinsi struct {
    nama_propinsi, jurusan    string
    jumlah_penduduk_2311102030 float64
}

// Definisi tipe data array propinsi dengan kapasitas maksimal 2023
type arrMhs [2023]propinsi

// Fungsi untuk mencari jumlah_penduduk_2311102030 tertinggi
// dalam array propinsi
func jumlah_penduduk_2311102030(T arrMhs, n int) float64 {
    var tertinggi float64 = T[0].jumlah_penduduk_2311102030
    var j int = 1
    for j < n {
        if tertinggi < T[j].jumlah_penduduk_2311102030 {
```

```

        tertinggi = T[j].jumlah_penduduk_2311102030
    }
    j = j + 1
}
return tertinggi
}

// Fungsi main untuk mengisi data propinsi dan mencari
jumlah_penduduk_2311102030 tertinggi
func main() {
    var n int
    var dataMhs arrMhs

    // Meminta input jumlah propinsi
    fmt.Print("Masukkan jumlah propinsi (maks 2023): ")
    fmt.Scan(&n)

    // Validasi jumlah propinsi yang dimasukkan
    if n < 1 || n > 2023 {
        fmt.Println("Jumlah propinsi harus antara 1 dan 2023.")
        return
    }

    // Mengisi data propinsi
    for i := 0; i < n; i++ {
        fmt.Printf("\nMasukkan data propinsi ke-%d\n", i+1)
        fmt.Print("Nama_propinsi: ")
        fmt.Scan(&dataMhs[i].nama_propinsi)
        fmt.Print("jumlah_penduduk_2311102030: ")
        fmt.Scan(&dataMhs[i].jumlah_penduduk_2311102030)
    }

    // Mencari dan menampilkan jumlah_penduduk_2311102030
    tertinggi
    tertinggi := jumlah_penduduk_2311102030(dataMhs, n)
    fmt.Printf("\njumlah_penduduk_2311102030 tertinggi dari %d
propinsi adalah: %.2f\n", n,
    tertinggi)
}

```

Output


```

Masukkan data propinsi ke-1
Nama_propinsi: jumlah_penduduk: exit status 0xc000013a
PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> go run "D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15\soal3.go"
Masukkan jumlah propinsi (maks 2023): 2

Masukkan data propinsi ke-1
Nama_propinsi: jateng
jumlah_penduduk: 5000

Masukkan data propinsi ke-2
Nama_propinsi:
jabar
jumlah_penduduk: 3000

jumlah penduduk tertinggi dari 2 propinsi adalah: 5000.00
PS D:\kulia\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> 

```

IV. Studi kasus

- 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?

Halaman 94 | Modul Praktikum Algoritma dan Pemrograman 2

"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.

```

package main

import (
    "fmt"
    "sort"
)

func main() {
    var numbers []int
    for {

```

```

var input int
fmt.Scan(&input)
if input == -5313541 {
    break
}
if input != 0 {
    numbers = append(numbers, input)
}
}

// mengurutkan nama
sort.Ints(numbers)

// menghitung median
median := calculateMedian_2311102030(numbers)

// cetak
fmt.Println("Median:", median)
}

func calculateMedian_2311102030(data []int) float64 {
    length := len(data)
    if length == 0 {
        return 0
    }

    if length%2 == 1 {

        return float64(data[length/2])
    } else {

        mid1 := data[length/2-1]
        mid2 := data[length/2]
        return float64(mid1+mid2) / 2.0
    }
}

```

Output

```

PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> go run "D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15\soal4.go"
7 23 11 0 5 19 2 29 3 13 17 0 -5313541
Median: 12
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15>

```

V. Studi kasus

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.

```
package main

import (
    "fmt"
    "sort"
)

// struct untuk partai_2311102030
type Partai_2311102030 struct {
    Nama int
    Suara int
}

func main() {
    const NMAX = 1000000
    suara := make([]int, NMAX+1) // tipe tab partai_2311102030

    for {
        var input int
        fmt.Scan(&input)
        if input == -1 {
            break
        }
        if input >= 1 && input <= NMAX {
            suara[input]++
        }
    }

    // mengumpulkan partai dan suara
    var partai_2311102030 []Partai_2311102030
```

```

for i := 1; i <= NMAX; i++ {
    if suara[i] > 0 {
        partai_2311102030 = append(partai_2311102030,
Partai_2311102030{Nama: i, Suara: suara[i]})
    }
}

// mengurutkan partai berdasarkan partai dan suara secara menurun
sort.Slice(partai_2311102030, func(i, j int) bool {
    if partai_2311102030[i].Suara == partai_2311102030[j].Suara {
        return partai_2311102030[i].Nama <
partai_2311102030[j].Nama
    }
    return partai_2311102030[i].Suara > partai_2311102030[j].Suara
})

// menampilkan hasil
for _, p := range partai_2311102030 {
    fmt.Printf("%d(%d) ", p.Nama, p.Suara)
}
}

```

```

PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> go run "D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15\soal5.go"4
5 5 5 1 2 3 1 2 4 -1
5(4) 1(2) 2(2) 3(1) 4(1)
PS D:\kuliah\semester3\praktikum alpro\2311102030_Didik Setiawan modul 15> █

```