

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL XII & XIII  
PENGURUTAN DATA**



**Disusun Oleh :  
Liya Khoirunnisa / 2311102124  
IF-11-05**

**Dosen Pengampu :  
Arif Amrulloh, S.Kom., M.Kom**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

## I. DASAR TEORI

Algoritma sorting adalah algoritma untuk meletakkan kumpulan elemen data ke dalam urutan tertentu, berdasarkan satu atau beberapa kunci ke dalam tiap tiap elemen. Berdasarkan data terurutnya, algoritma sorting dibagi menjadi dua jenis, yaitu pengurutan dari nilai terkecil hingga terbesar (ascending) dan pengurutan dari nilai terbesar hingga terkecil (descending).

Ada 2 macam metode sorting, yaitu insertion sort dan selection sort:

- a. Insertion sort adalah cara pengurutan data dengan mengecek satu persatu mulai dari yang kedua sampai dengan yang terakhir. Apabila ditemukan data yang lebih kecil dari data sebelumnya, maka data tersebut disisipkan pada posisi yang sesuai. Algoritma ini cukup efisien untuk mengurutkan sebuah list yang hampir terurut.

Contoh pseudocode proses mencari elemen data maksimum (ascending):

```
Procedure InsertionSort1 (input/output L : LarikInt, input n:integer)
L[1] ≤ L[2] ≤ ... ≤ L[n]

DEKLARASI
i,j,y : integer

ALGORITMA
for i ← 2 to n do
  y ← L[i]
  j ← i-1
  while (j ≥ 1) and (not ketemu) do
    if y < L[j] then
      L[j+1] ← L[j]
      j ← j-1
    else
      ketemu ← true
    endif
  endwhile
  {j<1 or ketemu}
  L[j+1] ← y
endfor
```

- b. Selection sort adalah cara pengurutan dengan membandingkan elemen sekarang dengan elemen yang berikutnya sampai terakhir. Jika ditemukan elemen paling kecil, kemudian ditukar dengan elemen sekarang.

Contoh pseudocode proses mencari elemen data maksimum (ascending):

```

Procedure SelectionSort1 (input/output L : LarikInt, input n:integer)
L[1] ≤ L[2] ≤ ... ≤ L[n]

DEKLARASI
i,j,imaks,maks,temp : integer

ALGORITMA
for i ← n downto 2 do
    imaks ← 1
    maks ← L[1]

    for j ← 2 to i do
        if L[j] > L[maks] then
            imaks ← j
            maks ← L[j]
        endif
    endfor

    temp ← L[i]
    L[i] ← L[maks]
    L[imaks] ← temp
endfor

```

## II. GUIDED

1. Hercules, preman terkenal seantero ibukota, memiliki kerabat di banyak daerah. Tentunya Hercules sangat suka mengunjungi semua kerabatnya itu. Diberikan masukan nomor rumah dari semua kerabatnya di suatu daerah, buatlah program rumahkerabat yang akan menyusun nomor-nomor rumah kerabatnya secara terurut membesar menggunakan algoritma selection sort.

Masukan dimulai dengan sebuah Integer  $n$  ( $0 < n < 1000$ ), banyaknya daerah kerabat Hercules tinggal. Isi  $n$  baris berikutnya selalu dimulai dengan sebuah Integer  $m$  ( $0 < m < 1000000$ ) yang menyatakan banyaknya rumah kerabat di daerah tersebut, diikuti dengan rangkaian bilangan bulat positif, nomor rumah para kerabat.

Keluaran terdiri dari  $n$  baris, yaitu rangkaian rumah kerabatnya terurut membesar di masing- masing daerah.

### Source code

```
/* Program Selection Sort */
package main

import (
    "fmt"
)

// Fungsi untuk mengurutkan array menggunakan selection sort
func selectionSort(arr []int, n int) {
    for i := 0; i < n-1; i++ {
        idxMin := i
        for j := i + 1; j < n; j++ {
            // Cari elemen terkecil
            if arr[j] < arr[idxMin] {
                idxMin = j
            }
        }
        // Tukar elemen terkecil dengan elemen di posisi i
        arr[i], arr[idxMin] = arr[idxMin], arr[i]
    }
}

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

```

        for daerah := 1; daerah <= n; daerah++ {
            var m int
            fmt.Printf("\nMasukkan jumlah nomor rumah kerabat
untuk daerah %d: ", daerah)
            fmt.Scan(&m)

            arr := make([]int, m)
            fmt.Printf("Masukkan %d nomor rumah kerabat : ", m)
            for i := 0; i < m; i++ {
                fmt.Scan(&arr[i])
            }
            selectionSort(arr, m)

            fmt.Printf("Nomor rumah terurut untuk daerah %d : ",
daerah)
            for _, num := range arr {
                fmt.Printf("%d ", num)
            }
            fmt.Println()
        }
    }
}

```

## Screenshoot Output



```

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [ ] [ ] ... ^ X
PS D:\Prak Alpro 2\laprakModul12> go run "d:\Prak Alpro 2\laprakModul12\guided1.go"
Masukkan jumlah daerah kerabat (n): 3

Masukkan jumlah nomor rumah kerabat untuk daerah 1: 5
Masukkan 5 nomor rumah kerabat : 2 1 7 9 13
Nomor rumah terurut untuk daerah 1 : 1 2 7 9 13

Masukkan jumlah nomor rumah kerabat untuk daerah 2: 6
Masukkan 6 nomor rumah kerabat : 189 15 27 39 75 133
Nomor rumah terurut untuk daerah 2 : 15 27 39 75 133 189

Masukkan jumlah nomor rumah kerabat untuk daerah 3: 3
Masukkan 3 nomor rumah kerabat : 4 9 1
Nomor rumah terurut untuk daerah 3 : 1 4 9
PS D:\Prak Alpro 2\laprakModul12> 

```

## Deskripsi Program

Program di atas dibuat untuk mengurutkan nomor rumah kerabat berdasarkan daerah menggunakan algoritma selection sort. Program dimulai dengan meminta input jumlah daerah kerbat. Lalu setiap daerah diminta untuk menginputkan jumlah nomor rumah yang ada di daerah tersebut, lalu pengguna diminta input nomor rumah satu persatu. Setiap nomor rumah pada masing-masing daerah diurutkan menggunakan

selection sort. Setelah semua nomor urut, program akan menampilkan nomor rumah yang sudah urut.

**\*Note:** Pada terminal terdapat 2 problem dikarenakan package dan func main digunakan juga di file lain

2. Buatlah sebuah program yang digunakan untuk membaca data Integer seperti contoh yang diberikan di bawah ini, kemudian diurutkan (menggunakan metoda insertion sort), dan memeriksa apakah data yang terurut berjarak sama terhadap data sebelumnya.

Masukan terdiri dari sekumpulan bilangan bulat yang diakhiri oleh bilangan negatif. Hanya bilangan non negatif saja yang disimpan ke dalam array.

Keluaran terdiri dari dua baris. Baris pertama adalah Isi dari array setelah dilakukan pengurutan, sedangkan baris kedua adalah status jarak setiap bilangan yang ada di dalam array. "Data berjarak x" atau "data berjarak tidak tetap".

### Source code

```
package main

import (
    "fmt"
)

// Fungsi untuk mengurutkan array menggunakan Insertion Sort
func insertionSort(arr []int, n int) {
    for i := 1; i < n; i++ {
        key := arr[i]
        j := i - 1

        // Geser elemen yang lebih besar dari key ke kanan
        for j >= 0 && arr[j] > key {
            arr[j+1] = arr[j]
            j--
        }
        arr[j+1] = key
    }
}

// Fungsi untuk memeriksa apakah selisih elemen array tetap
func isConstantDifference(arr []int, n int) (bool, int) {
```

```

    if n < 2 {
        return true, 0
    }

    difference := arr[1] - arr[0]
    for i := 1; i < n-1; i++ {
        if arr[i+1]-arr[i] != difference {
            return false, 0
        }
    }
    return true, difference
}

func main() {
    var arr []int
    var num int

    // Input data hingga bilangan negatif ditemukan
    fmt.Println("Masukkan data integer (akhiri dengan
bilangan negatif):")
    for {
        fmt.Scan(&num)
        if num < 0 {
            break
        }
        arr = append(arr, num)
    }

    n := len(arr)

    // Urutkan array menggunakan Insertion Sort
    insertionSort(arr, n)

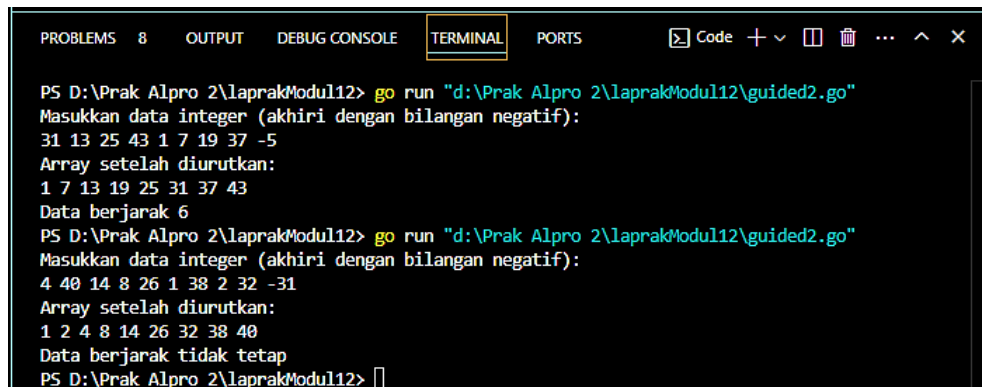
    // Periksa apakah selisih elemen tetap
    isConstant, difference := isConstantDifference(arr, n)

    // Tampilkan hasil pengurutan
    fmt.Println("Array setelah diurutkan:")
    for _, val := range arr {
        fmt.Printf("%d ", val)
    }
    fmt.Println()

    // Tampilkan status jarak
    if isConstant {
        fmt.Printf("Data berjarak %d\n", difference)
    } else {
        fmt.Println("Data berjarak tidak tetap")
    }
}

```

## Screenshoot Output



```
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + v [] [] ... ^ X

PS D:\Prak Alpro 2\laprakModul12> go run "d:\Prak Alpro 2\laprakModul12\guided2.go"
Masukkan data integer (akhiri dengan bilangan negatif):
31 13 25 43 1 7 19 37 -5
Array setelah diurutkan:
1 7 13 19 25 31 37 43
Data berjarak 6
PS D:\Prak Alpro 2\laprakModul12> go run "d:\Prak Alpro 2\laprakModul12\guided2.go"
Masukkan data integer (akhiri dengan bilangan negatif):
4 40 14 8 26 1 38 2 32 -31
Array setelah diurutkan:
1 2 4 8 14 26 32 38 40
Data berjarak tidak tetap
PS D:\Prak Alpro 2\laprakModul12> 
```

## Deskripsi Program

Program di atas dibuat untuk mengurutkan data yang dimasukkan pengguna dan memeriksa selisih dalam elemen array tersebut setelah pengurutan. Program dimulai dengan meminta inputan bilangan bulat yang diakhiri bilangan negatif untuk mengakhiri inputan. Data yang sudah dimasukkan akan diurutkan menggunakan algoritma insertion sort. Setelah diurutkan, program memeriksa apakah selisih antar elemen konstan atau tidak. Jika selisih antar elemen array tetap, maka program akan menampilkan selisihnya. Jika tidak, program akan memberitahu bahwa selisih antar elemen tidak tetap.

**\*Note:** Pada terminal terdapat 8 problem dikarenakan package dan func main digunakan juga di file lain



### III. UNGUIDED

1. Belakangan diketahui ternyata Hercules itu tidak berani menyeberang jalan, maka selalu diusahakan agar hanya menyeberang jalan sesedikit mungkin, hanya diujung Jalan. Karena nomor rumah sisi kiri jalan selalu ganjil dan sisi kanan jalan selalu genap, maka buatlah program kerabat dekat yang akan menampilkan nomor rumah mulai dari nomor yang ganjil lebih dulu terurut membesar dan kemudian menampilkan nomor rumah dengan nomor genap terurut mengecil.

Format Masukan masih persis sama seperti sebelumnya.

Keluaran terdiri dari n baris, yaitu rangkaian rumah kerabatnya terurut membesar untuk nomor ganjil, diikuti dengan terurut mengecil untuk nomor genap, di masing-masing daerah.

Keterangan: Terdapat 3 daerah dalam contoh masukan. Baris kedua berisi campuran bilangan ganjil dan genap. Baris berikutnya hanya berisi bilangan ganjil, dan baris terakhir hanya berisi bilangan genap.

Petunjuk:

Waktu pembacaan data, bilangan ganjil dan genap dipisahkan ke dalam dua array yang berbeda, untuk kemudian masing-masing diurutkan tersendiri.

Atau, tetap disimpan dalam satu array, diurutkan secara keseluruhan. Tetapi pada waktu pencetakan, mulai dengan mencetak semua nilai ganjil lebih dulu, kemudian setelah selesai cetaklah semua nilai genapnya.

#### Source code

```
/* Liya Khoirunnisa - 2311102124 */
package main

import (
    "fmt"
)

// Fungsi untuk Selection Sort
func selectionSort_124(arr []int, ascending bool) {
    // Mendapatkan panjang array
    n := len(arr)
    for i := 0; i < n-1; i++ {
```

```

        // Temukan indeks elemen terkecil atau terbesar
        index := i
        for j := i + 1; j < n; j++ {
            // Jika ascending, cari elemen terkecil namun
            // jika descending, cari elemen terbesar
            if (ascending && arr[j] < arr[index]) ||
            (!ascending && arr[j] > arr[index]) {
                index = j
            }
        }
        // Tukar elemen
        arr[i], arr[index] = arr[index], arr[i]
    }
}

func main() {
    // Deklarasi variabel
    var jumlahDaerah, jumlahNomor int

    // Input jumlah daerah
    fmt.Print("Masukkan jumlah daerah: ")
    fmt.Scan(&jumlahDaerah)

    // Perulangan untuk setiap daerah
    for i := 0; i < jumlahDaerah; i++ {
        // Input jumlah nomor rumah di daerah
        fmt.Print("\nMasukkan jumlah nomor rumah di daerah
", i+1, ": ")
        fmt.Scan(&jumlahNomor)

        // Slice untuk menyimpan nomor rumah
        nomorRumah := make([]int, jumlahNomor)

        // Input nomor rumah
        fmt.Print("Masukkan nomor rumah          :
")
        for j := 0; j < jumlahNomor; j++ {
            fmt.Scan(&nomorRumah[j])
        }

        // Pisahkan nomor ganjil dan genap
        var nomorGanjil []int
        var nomorGenap []int

        // Perulangan untuk memisahkan nomor ganjil dan
        genap
        for _, nomor := range nomorRumah {
            if nomor%2 == 0 {
                nomorGenap = append(nomorGenap, nomor)
            } else {
                nomorGanjil = append(nomorGanjil, nomor)
            }
        }

        // Urutkan nomor ganjil dan genap menggunakan
        Selection Sort
    }
}

```

```

        selectionSort_124(nomorGanjil, true) // Urutkan
        ganjil secara ascending
        selectionSort_124(nomorGenap, false) // Urutkan
        genap secara descending

        // Tampilkan hasil
        fmt.Print("Nomor rumah di daerah ", i+1, "
: ")
        for _, nomor := range nomorGanjil {
            fmt.Print(nomor, " ")
        }
        for _, nomor := range nomorGenap {
            fmt.Print(nomor, " ")
        }
        fmt.Println() // Untuk pindah ke baris baru setelah
        semua nomor ditampilkan
    }
}

```

## Screenshoot Output

```

PROBLEMS 17 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [ ] [ ] ... ^ X
PS D:\Prak Alpro 2\laprakModul12> go run "d:\Prak Alpro 2\laprakModul12\unguided1.go"
Masukkan jumlah daerah: 3

Masukkan jumlah nomor rumah di daerah 1: 5
Masukkan nomor rumah          : 2 1 7 9 13
Nomor rumah di daerah 1        : 1 7 9 13 2

Masukkan jumlah nomor rumah di daerah 2: 6
Masukkan nomor rumah          : 189 15 27 39 75 133
Nomor rumah di daerah 2        : 15 27 39 75 133 189

Masukkan jumlah nomor rumah di daerah 3: 3
Masukkan nomor rumah          : 4 9 1
Nomor rumah di daerah 3        : 1 9 4
PS D:\Prak Alpro 2\laprakModul12> 

```

**\*Note:** Untuk output daerah 1 dan 3 pada modul salah, yang benar adalah output saya. Kenapa salah? Karena contoh pada daerah pertama saya hanya menginputkan nomor “2 1 7 9 13” tanpa ada nomor “12 8”, namun pada modul outputnya terdapat nomor “12 8”. Pada daerah ketiga saya hanya menginputkan “4 9 1” tanpa ada nomor “8 2”, namun pada modul outputnya terdapat nomor “8 2”.

## Deskripsi Program

Program di atas dibuat untuk mengurutkan nomor rumah kerabat berdasarkan daerah menggunakan algoritma selection sort. Program

dimulai dengan meminta input jumlah daerah kerabat. Lalu setiap daerah diminta untuk menginputkan jumlah nomor rumah yang ada di daerah tersebut, lalu pengguna diminta input nomor rumah satu persatu. Setiap nomor rumah pada masing-masing daerah diurutkan menggunakan selection sort. Nomor ganjil diurutkan secara ascending (menaik), sedangkan nomor genap diurutkan secara descending (menurun). Setelah semua nomor urut, program akan menampilkan nomor rumah yang sudah urut.

**\*Note:** Pada terminal terdapat 17 problem dikarenakan package dan func main digunakan juga di file lain

2. Kompetisi pemrograman yang baru saja berlalu diikuti oleh 17 tim dari berbagai perguruan tinggi temama. 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 ke bawah."*

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-5313. Keluaran adalah median yang diminta, satu data per baris.

Keterangan:

Sampai bilangan 0 yang pertama, data terbaca adalah 7 23 11, setelah tersusun: 7 11 23. maka median saat itu adalah 11.

Sampai bilangan 0 yang kedua, data adalah 7 23 11 5 19 2 29 3 13. 17, setelah tersusun diperoleh. 2 3 5 7 11 13 17 19 23 29. Karena ada 10 data, genap, maka median adalah  $(11+13)/2=12$ .

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 metode insertion sort dan ambil mediannya.

### Source code

```
/* Liya Khoirunisa - 2311102124 */
package main

import (
    "fmt"
)

// Fungsi untuk menghitung median
func hitungMedian_124(data []int) int {
    n := len(data)
    if n%2 == 1 {
        // Jika jumlah data ganjil, ambil nilai tengah
        return data[n/2]
    } else {
        // Jika jumlah data genap, ambil rata-rata dua nilai
        // tengah dibulatkan ke bawah
        return (data[n/2-1] + data[n/2]) / 2
    }
}

// Fungsi untuk mengurutkan data menggunakan selection sort
func selectionSort_124(arr []int) {
    for i := 0; i < len(arr)-1; i++ {
        // Menentukan indeks elemen terkecil di sisa array
        minIdx := i
        for j := i + 1; j < len(arr); j++ {
            if arr[j] < arr[minIdx] {
                minIdx = j
            }
        }
        // Menukar elemen terkecil dengan elemen di posisi i
        arr[i], arr[minIdx] = arr[minIdx], arr[i]
    }
}
```

```

func main() {
    // Deklarasi variabel
    var data []int
    var num int

    // Meminta input rangkaian bilangan bulat
    fmt.Println("Masukkan rangkaian bilangan bulat (ketik -
5313 untuk mengakhiri): ")
    for {
        // Membaca input dari pengguna
        fmt.Scan(&num)

        // Jika menginputkan -5313, maka keluar dari loop
        if num == -5313 {
            break
        }

        // Jika input 0 maka data akan diproses
        if num == 0 {
            if len(data) > 0 {
                // Urutkan data menggunakan selection sort
                selectionSort_124(data)

                // Tampilkan data yang sudah tersusun
                fmt.Println("Data yang sudah tersusun:",
data)

                // Hitung median
                median := hitungMedian_124(data)
                fmt.Println("Median:", median)
            }
            // Tidak mengosongkan data untuk pembacaan
            berikutnya
        } else {
            data = append(data, num)
        }
    }
}

```

## Screenshoot Output

```

PS D:\Prak Alpro 2\laprakModul12> go run "d:\Prak Alpro 2\laprakModul12\uguided2.go"
Masukkan rangkaian bilangan bulat (ketik -5313 untuk mengakhiri):
7 23 11 0 5 19 2 29 3 13 17 0 -5313
Data yang sudah tersusun: [7 11 23]
Median: 11
Data yang sudah tersusun: [2 3 5 7 11 13 17 19 23 29]
Median: 12
PS D:\Prak Alpro 2\laprakModul12>

```

### Deskripsi Program

Program di atas dibuat untuk menghitung median dari rangkaian bilangan bulat. Program ini menggunakan algoritma selection sort untuk mengurutkan data. Program dimulai dengan meminta input rangkaian bilangan bulat sampai pengguna menginputkan -5313 untuk menghentikan input. Setiap angka inputan selain 0, akan dimasukkan ke data. Jika 0, program akan memproses data yang telah dimasukkan. Alur programnya yaitu data akan diurutkan dengan menggunakan selection sort. Setelah data terurut, program akan menghitung median. Jika data ganjil, mediannya adalah nilai yang ditengah data. Namun jika data genap, mediannya adalah rata rata dari dua nilai tengah data. Kemudian hasil median akan dicetak ke layar.

**\*Note:** Pada terminal terdapat 20 problem dikarenakan package dan func main digunakan juga di file lain

3. Sebuah program perpustakaan digunakan untuk mengelola data buku di dalam suatu perpustakaan. Misalnya terdefinisi struct dan array seperti berikut ini:

```
const nMax integer = 7919
type Buku = <
    id, judul, penulis, penerbit : string
    eksemplar, tahun, rating : integer >

type DaftarBuku array [1..nMax] of Buku
Pustaka : DaftarBuku
nPustaka: integer
```

Masukan terdiri dari beberapa baris, Baris pertama adalah bilangan bulat N yang menyatakan banyaknya data buku yang ada di dalam perpustakaan. N baris berikutnya, masing-masingnya adalah data buku sesuai dengan atribut atau field pada struct. Baris terakhir adalah bilangan bulat yang menyatakan rating buku yang akan dicari.

Keluaran terdiri dari beberapa baris. Baris pertama adalah data buku terfavorit, baris kedua adalah lima judul buku dengan rating tertinggi, selanjutnya baris terakhir adalah data buku yang dicari sesuai rating yang diberikan pada masukan baris terakhir.

Lengkapi subprogram-subprogram dibawah ini, sesuai dengan I.S. dan F.S yang diberikan.

```
procedure DaftarkanBuku(in/out pustaka : DaftarBuku, n : integer)
{I.S. sejumlah n data buku telah siap para piranti masukan
F.S. n berisi sebuah nilai, dan pustaka berisi sejumlah n data buku}

procedure Cetak Terfavorit(in pustaka : DaftarBuku, in n : integer)
{I.S. array pustaka berisi n buah data buku dan belum terurut
F.S. Tampilan data buku (judul, penulis, penerbit, tahun) terfavorit, yaitu memiliki
rating tertinggi}

procedure UrutBuku(in/out pustaka : DaftarBuku, n : integer)
{I.S. Array pustaka berisi n data buku
F.S. Array pustaka terurut menurun/mengecil terhadap rating.
Catatan: Gunakan metoda Insertion sort}

procedure Cetak5Terbaru( in pustaka : DaftarBuku, n integer)
{I.S. pustaka berisi n data buku yang sudah terurut menurut rating
F.S. Laporan 5 judul buku dengan rating tertinggi
Catatan: Isi pustaka mungkin saja kurang dari 5}

procedure CariBuku (in pustaka : DaftarBuku, n : integer, r : integer)
{I.S. pustaka berisi n data buku yang sudah terurut menurut rating
F.S. Laporan salah satu buku (judul, penulis, penerbit, tahun, eksemplar, rating) dengan
rating yang diberikan. Jika tidak ada buku dengan rating yang ditanyakan, cukup
tuliskan "Tidak ada buku dengan rating seperti itu". Catatan: Gunakan pencarian
biner/belah dua.}
```

### Source code

```
/* Liya Khoirunisa - 2311102124 */
package main
```



```

import (
    "fmt"
)

// Nilai maksimum
const nMax int = 7919

// Struct buku
type Buku struct {
    id        string
    judul     string
    penulis   string
    penerbit  string
    eksemplar int
    tahun     int
    rating    int
}

// Slice dari struct buku
type DaftarBuku []Buku

// Fungsi untuk mendaftarkan buku baru ke pustaka
func daftarkanBuku_124(pustaka *DaftarBuku, n int) {
    for i := 0; i < n; i++ {
        var buku Buku

        // Deklarasi buku
        fmt.Printf("Masukkan data buku ke-%d:\n", i+1)
        fmt.Print("ID buku          : ")
        fmt.Scan(&buku.id)
        fmt.Print("Judul buku         : ")
        fmt.Scan(&buku.judul)
        fmt.Print("Penulis buku        : ")
        fmt.Scan(&buku.penulis)
        fmt.Print("Penerbit buku       : ")
        fmt.Scan(&buku.penerbit)
        fmt.Print("Eksemplar buku      : ")
        fmt.Scan(&buku.eksemplar)
        fmt.Print("Tahun terbit buku: ")
        fmt.Scan(&buku.tahun)
        fmt.Print("Rating buku         : ")
        fmt.Scan(&buku.rating)
        *pustaka = append(*pustaka, buku)
    }
}

// Fungsi untuk mencetak buku dengan rating tertinggi
func cetakTerfavorit_124(pustaka DaftarBuku) {
    if len(pustaka) == 0 {
        fmt.Println("Tidak ada data buku.")
        return
    }
    maxRating := pustaka[0]

    // Mencari buku dengan rating tertinggi
    for _, buku := range pustaka {

```

```

        if buku.rating > maxRating.rating {
            maxRating = buku
        }
    }

    // Menampilkan buku terfavorit
    fmt.Println("\n///// Buku Terfavorit /////")
    fmt.Printf("Judul buku      : %s\n", maxRating.judul)
    fmt.Printf("Penulis buku       : %s\n", maxRating.penulis)
    fmt.Printf("Penerbit buku        : %s\n",
maxRating.penerbit)
    fmt.Printf("Tahun terbit buku: %d\n", maxRating.tahun)
}

// Fungsi untuk mengurutkan pustaka berdasarkan rating
func urutBuku_124(pustaka *DaftarBuku) {
    for i := 1; i < len(*pustaka); i++ {
        key := (*pustaka)[i]
        j := i - 1
        for j >= 0 && (*pustaka)[j].rating < key.rating {
            (*pustaka)[j+1] = (*pustaka)[j]
            j--
        }
        (*pustaka)[j+1] = key
    }
}

// Fungsi untuk mencetak 5 buku dengan rating tertinggi
func cetak5Terbaru_124(pustaka DaftarBuku) {
    fmt.Println("\n///// 5 Buku dengan Rating Tertinggi
/////")
    for i := 0; i < len(pustaka) && i < 5; i++ {
        fmt.Printf("%d. Judul: %s, Rating: %d\n", i+1,
pustaka[i].judul, pustaka[i].rating)
    }
}

// Fungsi untuk mencari buku berdasarkan rating
func cariBuku_124(pustaka DaftarBuku, r int) {
    low, high := 0, len(pustaka)-1
    for low <= high {
        mid := (low + high) / 2
        if pustaka[mid].rating == r {
            buku := pustaka[mid]
            fmt.Printf("\nBuku ditemukan:\n")
            fmt.Printf("Judul buku      : %s\n",
buku.judul)
            fmt.Printf("Penulis buku       : %s\n",
buku.penulis)
            fmt.Printf("Penerbit buku        : %s\n",
buku.penerbit)
            fmt.Printf("Tahun terbit buku: %d\n",
buku.tahun)
            fmt.Printf("Eksemplar buku      : %d\n",
buku.eksemplar)
            fmt.Printf("Rating buku         : %d\n",

```

```

buku.rating)
    return
    } else if pustaka[mid].rating < r {
        high = mid - 1
    } else {
        low = mid + 1
    }
}
fmt.Println("\nTidak ada buku dengan rating seperti
itu.")
}

func main() {
    // Deklarasi variabel
    var pustaka DaftarBuku
    var n, rating int

    // Meminta input jumlah buku
    fmt.Print("Masukkan jumlah buku yang ingin didaftarkan:
")
    fmt.Scan(&n)

    // Panggil fungsi untuk mendaftarkan buku
    daftarkanBuku_124(&pustaka, n)

    // Panggil fungsi untuk mengurutkan buku
    urutBuku_124(&pustaka)

    // Panggil fungsi untuk menampilkan buku dengan rating
    tertinggi
    cetakTerfavorit_124(pustaka)

    // Panggil fungsi untuk menampilkan 5 buku dengan rating
    tertinggi
    cetak5Terbaru_124(pustaka)

    // Meminta input rating yang ingin dicari
    fmt.Print("\nMasukkan rating yang ingin dicari: ")
    fmt.Scan(&rating)

    // Panggil fungsi untuk mencari buku berdasarkan rating
    cariBuku_124(pustaka, rating)
}

```

## Screenshoot Output

```
PROBLEMS 23 OUTPUT DEBUG CONSOLE TERMINAL PORTS Code + - [] [] ... ^ >

PS D:\Prak Alpro 2\laprakModul12> go run "d:\Prak Alpro 2\laprakModul12\unguided3.go"
Masukkan jumlah buku yang ingin didaftarkan: 2
Masukkan data buku ke-1:
ID buku      : BLIYA001
Judul buku   : Kerinduan
Penulis buku : Khoirun
Penerbit buku : Gramedia
Eksemplar buku : 231
Tahun terbit buku: 2021
Rating buku  : 4
Masukkan data buku ke-2:
ID buku      : BLIYA002
Judul buku   : Cinta
Penulis buku : Nisa
Penerbit buku : Gramedia
Eksemplar buku : 110
Tahun terbit buku: 2024
Rating buku  : 5

///// Buku Terfavorit /////
Judul buku   : Cinta
Penulis buku : Nisa
Penerbit buku : Gramedia
Tahun terbit buku: 2024

Masukkan rating yang ingin dicari: 5

Buku ditemukan:
Judul buku   : Cinta
Penulis buku : Nisa
Penerbit buku : Gramedia
Tahun terbit buku: 2024
Eksemplar buku : 110
Rating buku  : 5
PS D:\Prak Alpro 2\laprakModul12> 
```

## Deskripsi Program

Program di atas dibuat untuk mendaftarkan, mengurutkan, mencari, dan menampilkan informasi buku. Program menggunakan algoritma insertion sort untuk mengurutkan buku berdasarkan rating dan menggunakan algoritma pencarian biner untuk menemukan buku berdasarkan rating. Cara kerja program dimulai dengan meminta input jumlah buku yang ingin didaftarkan. Lalu pengguna diminta untuk memasukkan data seperti id, judul, penulis, penerbit, tahun terbit, eksemplar, dan rating buku. Data tersebut disimpan dalam slice. Buku yang terdaftar akan diurutkan berdasarkan rating menggunakan insertion sort. Kemudian program mencetak buku terfavorit dengan rating tertinggi. Selanjutnya pengguna

dapat mencari buku berdasarkan rating. Dalam mencari buku menggunakan algoritma binary search.

**\*Note:** Pada terminal terdapat 12 problem dikarenakan package dan func main digunakan juga di file lain

## **DAFTAR PUSTAKA**

- [1] Retnoningsih, E. (2018). Algoritma pengurutan data (sorting) dengan metode insertion sort dan selection sort. *Information Management For Educators And Professionals: Journal of Information Management*, 3(1), 95-106.