

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 12 & 13
PENGURUTAN DATA



Disusun Oleh:

Muhammad Djulianoor / 2311102253

Kelas

IF-11-06

Dosen Pengampu:

Abednego Dwi Septiadi

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Dasar Teori

Sorting

Sorting ialah proses pengurutan data baik dari nilai tertinggi maupun dari nilai terendah. Ada banyak algoritma populer untuk mengurutkan data, seperti *bubble sort*, *selection sort*, *quick sort*, *insertion sort*, *merge sort*.

Bubble Sort

Bubble sort adalah metode pengurutan algoritma dengan cara melakukan penukaran data secara terus-menerus hingga bisa dipastikan dalam suatu iterasi tertentu tidak ada lagi perubahan. Algoritma ini menggunakan perbandingan dalam operasi antar elemennya.

Selection Sort

Teknik ini adalah proses pengurutan yang paling sederhana dibanding teknik lainnya. *Selection sort* bekerja dengan cara mencari elemen terkecil dalam *array*, selanjutnya memindahkan elemen tersebut pada posisi pertama *array*.

Insertion Sort

Insertion sort merupakan algoritma yang melakukan proses pengurutan dengan cara menyisipkan elemen di antara elemen lain yang nilainya lebih kecil dan lebih besar daripada elemen yang disisipkan.

Merge Sort

Cara kerja teknik ini adalah dengan memecah kelompok data menjadi kelompok yang lebih kecil, selanjutnya melakukan pengurutan dalam kelompok-kelompok tersebut, dan menggabungkannya dengan kelompok data yang telah diurutkan.

Quick Sort

Algoritma ini menggunakan konsep pembagian kelompok data menjadi kelompok-kelompok yang lebih kecil, namun dengan pendekatan yang berbeda. Perbedaannya terletak pada proses pengurutan yang cepat

II. GUIDED

1. Soal Studi Case

Source code

```
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)

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

        // Membaca nomor rumah untuk daerah ini
        arr := make([]int, m)
        fmt.Printf("Masukkan %d nomor rumah kerabat: ", m)
        for i := 0; i < m; i++ {
```

```

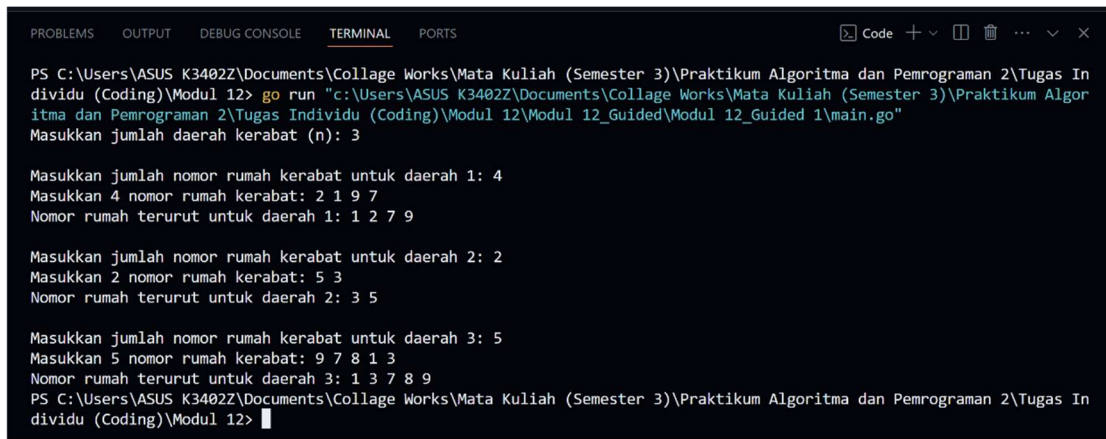
        fmt.Scan(&arr[i])
    }

    // Urutkan array dari terkecil ke terbesar
    selectionSort(arr, m)

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

```

Screenshoot Output



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12\Modul 12_Guided\Modul 12_Guided 1\main.go"
Masukkan jumlah daerah kerabat (n): 3

Masukkan jumlah nomor rumah kerabat untuk daerah 1: 4
Masukkan 4 nomor rumah kerabat: 2 1 9 7
Nomor rumah terurut untuk daerah 1: 1 2 7 9

Masukkan jumlah nomor rumah kerabat untuk daerah 2: 2
Masukkan 2 nomor rumah kerabat: 5 3
Nomor rumah terurut untuk daerah 2: 3 5

Masukkan jumlah nomor rumah kerabat untuk daerah 3: 5
Masukkan 5 nomor rumah kerabat: 9 7 8 1 3
Nomor rumah terurut untuk daerah 3: 1 3 7 8 9
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12>

```

Deskripsi Program

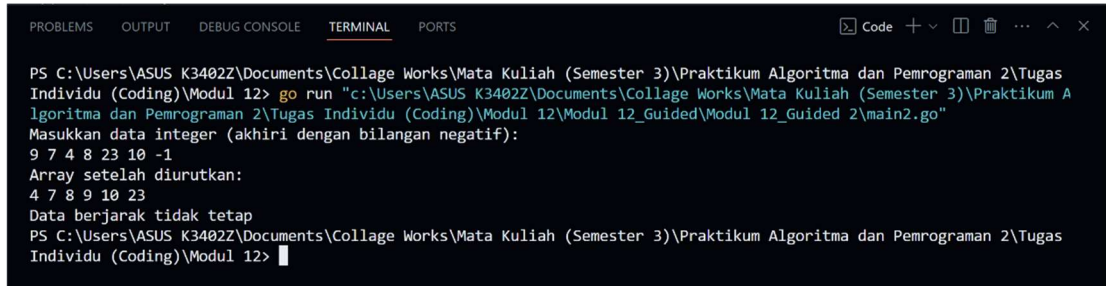
Program di atas merupakan implementasi dari *selection sort* yang bertujuan untuk mengurutkan data dari yang terkecil hingga yang terbesar. Pada program ini pengguna diminta untuk memasukkan nomor rumah kerabat, pengguna bisa saja memasukkan nomor rumah kerabat dengan angka yang tidak berurutan, tujuan dari program ini adalah untuk mengurutkan nomor rumah kerabat dari yang terkecil hingga yang terbesar.

2. Source Code

```
3. package main
4.
5. import (
6.     "fmt"
7. )
8.
9. // Fungsi untuk mengurutkan array menggunakan Insertion Sort
10. func insertionSort(arr []int, n int) {
11.     for i := 1; i < n; i++ {
12.         key := arr[i]
13.         j := i - 1
14.
15.         // Geser elemen yang lebih besar dari key ke kanan
16.         for j >= 0 && arr[j] > key {
17.             arr[j+1] = arr[j]
18.             j--
19.         }
20.         arr[j+1] = key
21.     }
22. }
23.
24. // Fungsi untuk memeriksa apakah selisih elemen array tetap
25. func isConstantDifference(arr []int, n int) (bool, int) {
26.     if n < 2 {
27.         return true, 0
28.     }
29.
30.     difference := arr[1] - arr[0]
31.     for i := 1; i < n-1; i++ {
32.         if arr[i+1]-arr[i] != difference {
33.             return false, 0
34.         }
35.     }
36.     return true, difference
37. }
38.
39. func main() {
40.     var arr []int
41.     var num int
42.
```

```
43. // Input data hingga bilangan negatif ditemukan
44. fmt.Println("Masukkan data integer (akhiri dengan bilangan
    negatif):")
45. for {
46.     fmt.Scan(&num)
47.     if num < 0 {
48.         break
49.     }
50.     arr = append(arr, num)
51. }
52.
53. n := len(arr)
54.
55. // Urutkan array menggunakan Insertion Sort
56. insertionSort(arr, n)
57.
58. // Periksa apakah selisih elemen tetap
59. isConstant, difference := isConstantDifference(arr, n)
60.
61. // Tampilkan hasil pengurutan
62. fmt.Println("Array setelah diurutkan:")
63. for _, val := range arr {
64.     fmt.Printf("%d ", val)
65. }
66. fmt.Println()
67.
68. // Tampilkan status jarak
69. if isConstant {
70.     fmt.Printf("Data berjarak %d\n", difference)
71. } else {
72.     fmt.Println("Data berjarak tidak tetap")
73. }
74. }
75.
```

Screenshoot Output



```
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12\Modul 12_Guided\Modul 12_Guided 2\main2.go"
Masukkan data integer (akhiri dengan bilangan negatif):
9 7 4 8 23 10 -1
Array setelah diurutkan:
4 7 8 9 10 23
Data berjarak tidak tetap
PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12> █
```

Deskripsi Program

Program di atas mengimplementasikan *sorting*. *Sorting* yang dilakukan pada program tersebut ialah mengurutkan angka terkecil hingga angka terbesar. Algoritma yang digunakan adalah *insertion sort*. Pada program ini, angka yang dinilai yang kecil, maka akan dipindahkan ke sebelah kiri, angka yang besar akan pindah ke sebelah kanan.

III. UNGUIDED

1. Soal Studi Case

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 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.

No	Masukan	Keluaran
1	3 5 2 1 7 9 13 6 189 15 27 39 75 133 3 4 9 1	1 13 12 8 2 15 27 39 75 133 189 8 4 2

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 ganjil lebih dulu, kemudian setelah selesai cetaklah semua nilai genapnya.

Source code

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func selectionSort(arr_2311102253 []int, menaik bool) {
    n := len(arr_2311102253)
    for i := 0; i < n-1; i++ {
        extremeIdx := i
        for j := i + 1; j < n; j++ {
            if menaik {
                if arr_2311102253[j] < arr_2311102253[extremeIdx] {
                    extremeIdx = j
                }
            } else {
                if arr_2311102253[j] > arr_2311102253[extremeIdx] {
                    extremeIdx = j
                }
            }
        }
        arr_2311102253[i], arr_2311102253[extremeIdx] =
arr_2311102253[extremeIdx], arr_2311102253[i]
    }
}

func processRow(row string) string {
    x := strings.Fields(row)

    var odds, evens []int

    for _, numStr := range x {
        num, _ := strconv.Atoi(numStr)
        if num%2 == 0 {
            evens = append(evens, num)
        } else {
```

```

        odds = append(odds, num)
    }
}

selectionSort(odds, false)

selectionSort(evens, true)

var result strings.Builder
for _, odd := range odds {
    result.WriteString(fmt.Sprintf("%d ", odd))
}
for _, even := range evens {
    result.WriteString(fmt.Sprintf("%d ", even))
}

return strings.TrimSpace(result.String())
}

func main() {
    reader := bufio.NewReader(os.Stdin)

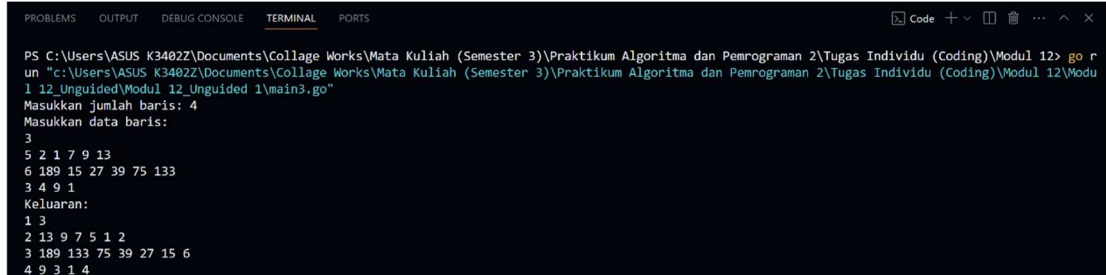
    fmt.Print("Masukkan jumlah baris: ")
    nStr, _ := reader.ReadString('\n')
    nStr = strings.TrimSpace(nStr)
    n, _ := strconv.Atoi(nStr)

    baris := make([]string, n)
    fmt.Println("Masukkan data baris: ")
    for i := 0; i < n; i++ {
        baris[i], _ = reader.ReadString('\n')
        baris[i] = strings.TrimSpace(baris[i])
    }

    fmt.Println("Keluaran:")
    for i, row := range baris {
        fmt.Printf("%d %s\n", i+1, processRow(row))
    }
}

```

Screenshoot Output



```
PS C:\Users\ASUS K34022\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12> go run "c:\Users\ASUS K34022\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12\Modul 12 Unguided\Modul 12 Unguided 1\main3.go"
Masukkan jumlah baris: 4
Masukkan data baris:
3
5 2 1 7 9 13
6 189 15 27 39 75 133
3 4 9 1
Keluaran:
1 3
2 13 9 7 5 1 2
3 189 133 75 39 27 15 6
4 9 3 1 4
```

Deskripsi Program

Program di atas merupakan implementasi dari *selection sort*. Masukan dari program ini adalah nomor rumah dari kerabat dekat. Program *selection sort* di atas merupakan program yang sederhana, karena *selection sort* akan memindahkan posisi angka berdasarkan permintaan. Pada kasus ini, program akan mengurutkan dari yang terkecil untuk angka yang ganjil. Sedangkan untuk angka yang genap akan diurutkan dari yang besar terlebih dahulu.

2. Soal Studi Case

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 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.

No	Masukan	Keluaran
1	7 23 11 0 5 19 2 29 3 13 17 0 -5313	11 12

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

```
package main

import (
    "bufio"
    "fmt"
    "os"
    "strconv"
    "strings"
)

func selectionSort(arr_2311102253 []int) {
    n := len(arr_2311102253)
    for i := 0; i < n-1; i++ {
        minIdx := i
        for j := i + 1; j < n; j++ {
            if arr_2311102253[j] < arr_2311102253[minIdx] {
                minIdx = j
            }
        }
        arr_2311102253[i], arr_2311102253[minIdx] =
arr_2311102253[minIdx], arr_2311102253[i]
    }
}

func calculateMedian(arr_2311102253 []int) int {
    n := len(arr_2311102253)
    if n%2 == 1 {
        return arr_2311102253[n/2]
    }
    return (arr_2311102253[n/2-1] + arr_2311102253[n/2]) / 2
}

func main() {
    reader := bufio.NewReader(os.Stdin)

    fmt.Println("Masukkan data (akhiri dengan -5313):")
    data := []int{}
    for {
        input, _ := reader.ReadString('\n')
        input = strings.TrimSpace(input)
```

```

values := strings.Fields(input)

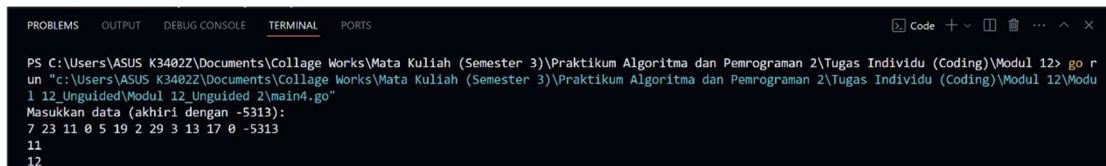
for _, value := range values {
    num, _ := strconv.Atoi(value)

    if num == -5313 {
        return
    }

    if num == 0 {
        selectionSort(data)
        median := calculateMedian(data)
        fmt.Println(median)
    } else {
        data = append(data, num)
    }
}
}
}

```

Screenshoot Output



```

PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12\Modul 12_Unguided\Modul 12_Unguided 2\main4.go"
Masukkan data (akhiri dengan -5313):
7 23 11 0 5 19 2 29 3 13 17 0 -5313
11
12

```

Deskripsi Program

Program di atas adalah program untuk mencari median atau nilai tengah dalam *array*. $n := \text{len}(\text{arr})$

```

if n%2 == 1 {
    return arr[n/2]
}

return (arr[n/2-1] + arr[n/2]) / 2

```

Syntax di atas merupakan algoritma dari pencarian median atau nilai tengah. Pada program ini, jika n dibagi 2 dan memiliki sisa 1, maka akan dikembalikan nilai dari $arr[n/2]$. Jika pada percabangan tersebut tidak terpenuhi, maka akan langsung pada $return (arr[n/2-1] + arr[n/2]) / 2$.

3. Soal Studi Case

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 di bawah 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 CetakTerfavorit(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

```
package main

import (
    "fmt"
)

const nMax = 7919

type Buku struct {
    id_2311102253, judul, penulis, penerbit string
    eksemplar, tahun, rating                int
}

type DaftarBuku [nMax]Buku

var pustaka DaftarBuku
var nPustaka int

func DaftarkanBuku(pustaka *DaftarBuku, n int) {
    nPustaka = n
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan data buku ke-%d (id_2311102253, judul, penulis, penerbit, eksemplar, tahun, rating):\n", i+1)
        fmt.Scan(&pustaka[i].id_2311102253, &pustaka[i].judul, &pustaka[i].penulis, &pustaka[i].penerbit, &pustaka[i].eksemplar, &pustaka[i].tahun, &pustaka[i].rating)
    }
}

func CetakTerfavorit(pustaka DaftarBuku, n int) {
    maxRating := -1
    var favorit Buku
    for i := 0; i < n; i++ {
        if pustaka[i].rating > maxRating {
            maxRating = pustaka[i].rating
            favorit = pustaka[i]
        }
    }
}
```

```

    fmt.Printf("Buku Terfavorit: %s oleh %s, diterbitkan oleh %s pada
tahun %d\n", favorit.judul, favorit.penulis, favorit.penerbit,
favorit.tahun)
}

func UrutBuku(pustaka *DaftarBuku, n int) {
    for i := 1; i < n; i++ {
        key := pustaka[i]
        j := i - 1
        for j >= 0 && pustaka[j].rating < key.rating {
            pustaka[j+1] = pustaka[j]
            j = j - 1
        }
        pustaka[j+1] = key
    }
}

func Cetak5Terbaru(pustaka DaftarBuku, n int) {
    fmt.Println("5 Buku dengan rating tertinggi:")
    for i := 0; i < 5 && i < n; i++ {
        fmt.Printf("%d. %s (%d)\n", i+1, pustaka[i].judul,
pustaka[i].rating)
    }
}

func CariBuku(pustaka DaftarBuku, n, rating int) {
    low, high := 0, n-1
    for low <= high {
        mid := (low + high) / 2
        if pustaka[mid].rating == rating {
            fmt.Printf("Ditemukan buku dengan rating %d: %s oleh %s,
diterbitkan oleh %s pada tahun %d, Eksemplar: %d\n",
rating, pustaka[mid].judul, pustaka[mid].penulis,
pustaka[mid].penerbit, pustaka[mid].tahun, pustaka[mid].eksemplar)
            return
        } else if pustaka[mid].rating < rating {
            high = mid - 1
        } else {
            low = mid + 1
        }
    }
}

```

```

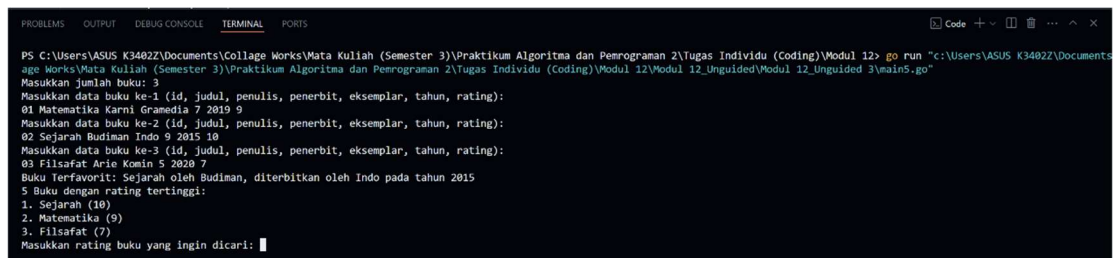
        fmt.Println("Tidak ada buku dengan rating seperti itu.")
    }

func main() {
    var n int
    fmt.Print("Masukkan jumlah buku: ")
    fmt.Scan(&n)
    DaftarkanBuku(&pustaka, n)
    UrutBuku(&pustaka, n)
    CetakTerfavorit(pustaka, n)
    Cetak5Terbaru(pustaka, n)

    var rating int
    fmt.Print("Masukkan rating buku yang ingin dicari: ")
    fmt.Scan(&rating)
    CariBuku(pustaka, n, rating)
}

```

Screenshoot Output



```

PS C:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12> go run "c:\Users\ASUS K3402Z\Documents\Collage Works\Mata Kuliah (Semester 3)\Praktikum Algoritma dan Pemrograman 2\Tugas Individu (Coding)\Modul 12\Unguided\Modul 12_Unguided 3\main5.go"
Masukkan jumlah buku: 3
Masukkan data buku ke-1 (id, judul, penulis, penerbit, eksemplar, tahun, rating):
01 Matematika Karni Gramedia 7 2019 9
Masukkan data buku ke-2 (id, judul, penulis, penerbit, eksemplar, tahun, rating):
02 Sejarah Budiman Indo 9 2015 10
Masukkan data buku ke-3 (id, judul, penulis, penerbit, eksemplar, tahun, rating):
03 Filsafat Arie Komin 5 2020 7
Buku Terfavorit: Sejarah oleh Budiman, diterbitkan oleh Indo pada tahun 2015
5 Buku dengan rating tertinggi:
1. Sejarah (10)
2. Matematika (9)
3. Filsafat (7)
Masukkan rating buku yang ingin dicari: 

```

Deskripsi Program

Program di atas telah diprogram dengan algoritma *insertion sort*. Pada program ini, keluaran yang terdiri dari beberapa baris setelah *user* memasukkan *input* program. Program tersebut akan mengurutkan *rating* buku, kemudian *user* dapat mencari *rating* yang ingin dicari pada program. Program juga terdiri dari beberapa prosedur yang memiliki tujuan masing-masing.