

**LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 12
PENGURUTAN DATA**



Disusun Oleh :

Loisa Vanica Saragih / 2311102280

S1 IF11-05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

Pengurutan data adalah proses penyusunan elemen – elemen yang ada dalam sebuah himpunan, daftar, atau deret bilangan yang bertujuan untuk mempermudah dalam pencarian data, analisis data, dan pengolahan datanya. Adapun pengurutan data memiliki berbagai metode seperti,

- Insertion Sort,
Insertion sort adalah metode pengurutan data dengan algoritma yang mengurutkan elemen elemen yang akan diurutkan dengan cara menyisipkan nya ke posisi yang tepat pada daftar terurut.
- Selection Sort,
Selection sort merupakan metode pengurutan data yang mengurutkannya dengan cara memilih elemen terkecil pada daftar yang belum terurut untuk kemudian menukarnya dengan elemen pertama yang ada pada daftar tersebut. Proses penukaran ini akan berlanjut sampai daftar yang ada menjadi daftar yang terurut.

Selain kedua metode diatas, pengurutan data masih memiliki metode lain, seperti bubble sort, quick sort, merge sort, dll.

Dalam impelementasinya sendiri, pengurutan data ini sering digunakan contohnya seperti dalam pengolahan data dan algoritma pencarian.

II. GUIDED

1. Hercules, preman terkenal seantero ibukota, memiliki kerabat dibanyak 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, di ikuti dengan rangkaian bilangan bulat positif, nomor rumah para kerabat.

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

Sourcecode

```
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; 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)
```

```

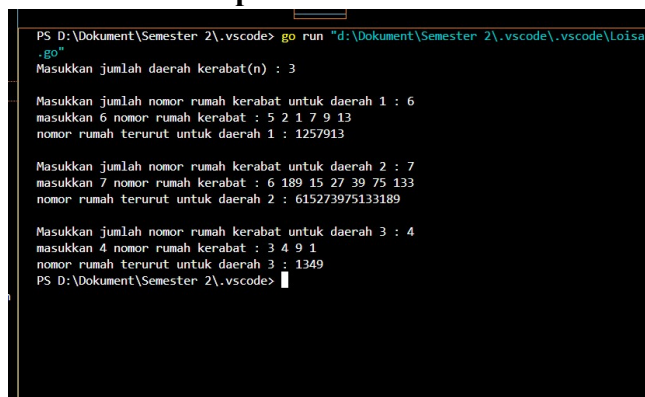
//proses tiap daerah
for daerah := 1; daerah <= n; daerah++ {
    var m int
    fmt.Printf("\nMasukkan jumlah nomor rumah kerabat
untuk daerah %d : ", daerah)
    fmt.Scan(&m)

    // 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



```

PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Loisa
.go"
Masukkan jumlah daerah kerabat(n) : 3

Masukkan jumlah nomor rumah kerabat untuk daerah 1 : 6
masukkan 6 nomor rumah kerabat : 5 2 1 7 0 13
nomor rumah terurut untuk daerah 1 : 1257913

Masukkan jumlah nomor rumah kerabat untuk daerah 2 : 7
masukkan 7 nomor rumah kerabat : 6 189 15 27 39 75 133
nomor rumah terurut untuk daerah 2 : 615273975133189

Masukkan jumlah nomor rumah kerabat untuk daerah 3 : 4
masukkan 4 nomor rumah kerabat : 3 4 9 1
nomor rumah terurut untuk daerah 3 : 1349
PS D:\Dokument\Semester 2\.vscode>

```

Deskripsi Program

Program diatas adalah implementasi penggunaan selection sort yang akan digunakan untuk mengurutkan nomor rumah kerabat pada beberapa daerah. Program akan meminta pengguna untuk menginputkan banyaknya jumlah daerah kerabat (n), dan untuk tiap

tiap daerah, program akan meminta pengguna untuk menginputkan jumlah nomor rumah kerabat (m), kemudian pengguna akan diminta untuk menginputkan m buah nomor rumah. Setelah pengguna menginputkan semua yang diminta oleh program, selanjutnya program akan memanggil fungsi selectionsort yang akan mengurutkan array yang ada dengan urutan dari yang terkecil ke yang terbesar. Setelah array terurut, program akan melakukan hal yang sama ke setiap daerah yang sebelumnya telah pengguna inputkan. Terakhir, program akan mengeluarkan output berupa array yang terurut.

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".

Sourcecode

```
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

```

PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Loisa Vanica\Guided2
.go"
Masukkan data integer (akhiri dengan bilangan negatif):
7 23 11 0 5 19 2 29 3 13 17 0 -5313
Array setelah diurutkan:
0 0 2 3 5 7 11 13 17 19 23 29
Data berjarak tidak tetap
PS D:\Dokument\Semester 2\.vscode>

```

Deskripsi Program

Program diatas adalah program untuk implementasi insertion sort yang digunakan untuk mengurutkan dan memeriksa sebuah array apakah array tersebut memiliki selisih jarak yang tetap atau tidak. Pertama-tama, program akan meminta pengguna untuk menginputkan bilangan bilangan (integer) sampai ditemukannya bilangan negatif untuk mengakhiri inputan. Selanjutnya dengan menggunakan fungsi insertionsort program akan mengurutkan array yang sebelumnya telah di input oleh pengguna.

Setelah array terurut, program akan melanjutkan dengan memanggil fungsi isConstantDifference() yang berguna untuk memeriksa selisih diantara elemen yang ada dalam array, selisihnya tetap atau tidak. Jika selisih antar elemen array tetap maka program akan mengambil nilai true, namun jika sebaliknya maka program akan mengambil nilai false. Pada akhirnya, program akan mengeluarkan output berupa array yang sudah terurut dan pernyataan apakah selisih antar elemen array itu memiliki jarak yang tetap atau tidak.

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 ganjil lebih dulu terurut membesar dan kemudian menampilkan nomor rumah dengan nomor genap terurut mengecil.

Format **masukan** masih persis 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..

Sourcecode

```
package main

import (
    "fmt"
    "sort"
)

//Loisa Vanica_2311102280
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])
        }

        sort.Ints(arr)

        // Pisahkan nomor rumah ganjil dan genap
```



```

var oddNums, evenNums []int
for _, num := range arr {
    if num%2 == 0 {
        evenNums = append(evenNums, num)
    } else {
        oddNums = append(oddNums, num)
    }
}

// Tampilkan nomor rumah ganjil terurut membesar
fmt.Printf("nomor rumah ganjil terurut untuk daerah
%d : ", daerah)
for _, num := range oddNums {
    fmt.Printf("%d ", num)
}
fmt.Println()

// Tampilkan nomor rumah genap terurut mengecil
fmt.Printf("nomor rumah genap terurut untuk daerah
%d : ", daerah)
sort.Sort(sort.Reverse(sort.IntSlice(evenNums)))
for _, num := range evenNums {
    fmt.Printf("%d ", num)
}
fmt.Println()
}
}

```

Screenshoot Output

```

PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Loisa Vanica\Unguide
d1.go"
Masukkan jumlah daerah kerabat(n) : 3

Masukkan jumlah nomor rumah kerabat untuk daerah 1 : 6
masukkan 6 nomor rumah kerabat : 5 2 1 7 9 13
nomor rumah ganjil terurut untuk daerah 1 : 1 5 7 9 13
nomor rumah genap terurut untuk daerah 1 : 2

Masukkan jumlah nomor rumah kerabat untuk daerah 2 : 7
masukkan 7 nomor rumah kerabat : 6 189 15 27 39 75 133
nomor rumah ganjil terurut untuk daerah 2 : 15 27 39 75 133 189
nomor rumah genap terurut untuk daerah 2 : 6

Masukkan jumlah nomor rumah kerabat untuk daerah 3 : 4
masukkan 4 nomor rumah kerabat : 3 4 9 1
nomor rumah ganjil terurut untuk daerah 3 : 1 3 9
nomor rumah genap terurut untuk daerah 3 : 4
PS D:\Dokument\Semester 2\.vscode>

```

Deskripsi Program

Program diatas adalah program yang berguna untuk memisahkan dan mengurutkan nomor rumah kerabat berdasarkan daerah mereka tinggal. Pertama-tama, pengguna akan diminta untuk menginputkan jumlah daerah kerabat (n) yang kemudian akan di proses oleh program, setelahnya pengguna akan diminta untuk menginputkan jumlah nomor rumah kerabat (m), juga nomor nomor rumah kerabatnya. Setelah semua inputan yang diperlukan lengkap, program akan mengurutkan nomor rumah menggunakan fungsi `sort.Ints()`. Lalu program akan lanjut memisahkan nomor ganjil dan genap dalam dua slice yang berbeda yaitu `slice oddNums` dan `evenNums`, kemudian program akan menampilkan nomor rumah ganjil yang telah terurut dari yang terkecil dan terbesar, begitupun dengan nomor rumah yang genap. Program akan mengulangi hal yang sama pada setiap daerah. Lalu, output akan ditampilkan.

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 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 O.

Masukan berbentuk rangkaian bilangan bulat. Masukan tidak akan berisi lebih dari 1000000 data, tidak termasuk bilangan O. Data O 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.

Sourcecode

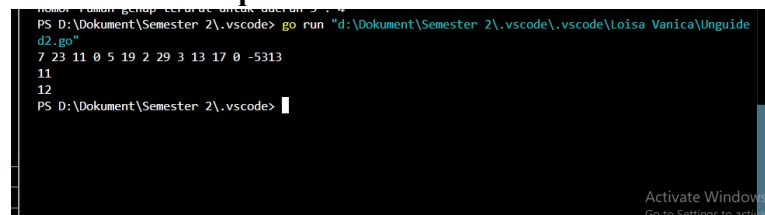
```
package main

import (
    "fmt"
    "sort"
)

//Loisa Vanica_2311102280
func main() {
    var num int
    var data []int

    for {
        fmt.Scan(&num)
        if num == -5313 {
            break
        }
        if num == 0 {
            // Cetak median
            n := len(data)
            sort.Ints(data)
            median := data[n/2]
            if n%2 == 0 {
                median = (data[n/2-1] + data[n/2]) / 2
            }
            fmt.Println(median)
        } else {
            data = append(data, num)
        }
    }
}
```

Screenshoot Output



```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Loisa Vanica\Unguide
d2.go"
7 23 11 0 5 19 2 29 3 13 17 0 -5313
11
12
PS D:\Dokument\Semester 2\.vscode>
```

Deskripsi Program

Program diatas digunakan untuk menghitung median dari sebuah deret bilangan, program ini akan meminta pengguna untuk menginput bilangan

bulat. Setiap pengguna menginputkan sebuah bilangan, bilangan tersebut akan di simpan dalam slice data. Jikalau pengguna menginputkan angka 0 maka, program akan mencari median dari deret bilangan yang telah di inputkan. Sebelum menghitung median, program akan mengurutkan data dalam slice data dengan fungsi `sort.Inst()`. Lalu program akan menghitung median nya. Jika, panjang slice data yang ada genap maka median akan dihitung sebagai rata-rata dua bilangan ditengah, setelahnya program akan mencetak mediannya. Lalu, jika pengguna memasukkan -5313 maka program akan berhenti dan keluar dari perulangan.

3. Sebuah program perpustakaan digunakan untuk mengelola data buku a 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 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.)
```

Sourcecode

```
package main

import (
    "fmt"
```

```

        "sort"
    )

    //Loisa vanica 2311102280
    const nMax = 7919

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

    type DaftarBuku struct {
        Pustaka []Buku
        nPustaka int
    }

    func DaftarkanBuku(pustaka *DaftarBuku, n int) {
        pustaka.nPustaka = n
        pustaka.Pustaka = make([]Buku, n)
        for i := 0; i < n; i++ {
            fmt.Printf("Masukkan data buku ke-%d\n", i+1)
            fmt.Print("ID: ")
            fmt.Scan(&pustaka.Pustaka[i].id)
            fmt.Print("Judul: ")
            fmt.Scan(&pustaka.Pustaka[i].judul)
            fmt.Print("Penulis: ")
            fmt.Scan(&pustaka.Pustaka[i].penulis)
            fmt.Print("Penerbit: ")
            fmt.Scan(&pustaka.Pustaka[i].penerbit)
            fmt.Print("Eksemplar: ")
            fmt.Scan(&pustaka.Pustaka[i].eksemplar)
            fmt.Print("Tahun: ")
            fmt.Scan(&pustaka.Pustaka[i].tahun)
            fmt.Print("Rating: ")
            fmt.Scan(&pustaka.Pustaka[i].rating)
        }
    }

    func CetakTerfavorit(pustaka DaftarBuku, n int) {
        sort.Slice(pustaka.Pustaka, func(i, j int) bool {

```

```

        return pustaka.Pustaka[i].rating >
pustaka.Pustaka[j].rating
    })
    for i := 0; i < n; i++ {
        fmt.Printf("Judul: %s, Penulis: %s, Penerbit: %s,
Tahun: %d\n",
            pustaka.Pustaka[i].judul,
pustaka.Pustaka[i].penulis,
            pustaka.Pustaka[i].penerbit,
pustaka.Pustaka[i].tahun)
    }
}

func UrutBuku(pustaka *DaftarBuku, n int) {
    sort.Slice(pustaka.Pustaka, func(i, j int) bool {
        return pustaka.Pustaka[i].rating >
pustaka.Pustaka[j].rating
    })
}

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

func CariBuku(pustaka DaftarBuku, n int, r int) {
    idx := sort.Search(n, func(i int) bool {
        return pustaka.Pustaka[i].rating <= r
    })
    if idx < n && pustaka.Pustaka[idx].rating == r {
        buku := pustaka.Pustaka[idx]
        fmt.Printf("Buku ditemukan: Judul: %s, Penulis: %s,
Penerbit: %s, Eksemplar: %d, Tahun: %d, Rating: %d\n",
            buku.judul, buku.penulis, buku.penerbit,
buku.eksemplar, buku.tahun, buku.rating)
    } else {
        fmt.Println("Tidak ada buku dengan rating seperti
itu.")
    }
}

func main() {
    var pustaka DaftarBuku

```

```

var n, r int

fmt.Print("Masukkan jumlah buku: ")
fmt.Scan(&n)

DaftarkanBuku(&pustaka, n)

fmt.Println("Daftar buku terfavorit:")
CetakTerfavorit(pustaka, n)

UrutBuku(&pustaka, n)

fmt.Println("Cetak 5 buku terbaru:")
Cetak5Terbaru(pustaka, n)

fmt.Print("Masukkan rating buku yang dicari: ")
fmt.Scan(&r)

CariBuku(pustaka, n, r)
}

```

Screenshoot Output

```

PS D:\Document\Semester 2\vscode> go run "d:\Document\Semester 2\vscode\loisa Vanica\ungul
daft.go
Masukkan jumlah buku: 3
Masukkan data buku ke-1
ID: 001
Judul: Hujan
Penulis: Terelipe
Penerbit: BumiFiksi
Eksemplar: 100
Tahun: 2020
Rating: 9
Masukkan data buku ke-2
ID: 002
Judul: Bumi
Penulis: Terelipe
Penerbit: BumiFiksi
Eksemplar: 200
Tahun: 2021
Rating: 9
Masukkan data buku ke-3
ID: 003
Judul: Bulan
Penulis: Terelipe
Penerbit: BumiFiksi
Eksemplar: 300
Tahun: 2022
Rating: 10
Daftar buku terfavorit:
Judul: Bulan, Penulis: Terelipe, Penerbit: BumiFiksi, Tahun: 2022
Judul: Hujan, Penulis: Terelipe, Penerbit: BumiFiksi, Tahun: 2020
Judul: Bumi, Penulis: Terelipe, Penerbit: BumiFiksi, Tahun: 2021
Cetak 5 buku terbaru:
Judul: Bulan, Rating: 10
Judul: Hujan, Rating: 9
Judul: Bumi, Rating: 9
Masukkan rating buku yang dicari: 9
Buku ditemukan: Judul: Hujan, Penulis: Terelipe, Penerbit: BumiFiksi, Eksemplar: 100, Tahun: 2020, Rating: 9
PS D:\Document\Semester 2\vscode> go run "d:\Document\Semester 2\vscode\Semester3.go modul 12 dan 13\daft.go

```

Deskripsi Program

Program diatas adalah program yang berguna untuk mengelola data buku yang ada disebuah perpustakaan. Program diatas memiliki beberapa fungsi yang berguna untuk mendaftarkan buku, mengurutkan buku berdasarkan rating, dsb.

Struktur data buku pada program berguna untuk menyimpan informasi dari setiap buku seperti ID, nama penulis, penerbit, eksemplar, tahun terbit, dan ratingnya. Kemudian, pengguna akan diminta memasukkan data buku yang ingin dimasukkan ke dalam koleksi, kemudian fungsi CetakTerfavorit dan UrutBuku akan berguna untuk mengurutkan buku berdasarkan rating secara menurun. Lalu, ada fungsi cetak5terbaru, fungsi ini akan mencetak 5 buku dengan rating tertinggi yang ada dalam program, jika buku yang diinputkan berjumlah kurang dari 5 maka program akan mencetak sesuai jumlah buku yang ada dalam koleksi. Terakhir, terdapat fungsi caribuku yang berguna untuk mencari buku sesuai dengan rating dan apa yang diinginkan oleh pengguna. Setelahnya, program akan mencetak output akhirnya.

DAFTAR PUSTAKA

- GeeksforGeeks. (n.d.). *Sorting Algorithms*. Retrieved from <https://www.geeksforgeeks.org/sorting-algorithms/>
- W3Schools. (n.d.). *Selection Sort Algorithm*. Retrieved from https://www.w3schools.com/dsa/dsa_algo_selectionsort.php
- W3Schools. (n.d.). *Insertion Sort Algorithm*. Retrieved from https://www.w3schools.com/dsa/dsa_algo_insertionsort.php