

**LAPORAN PRAKTIKUM  
ALGORITMA DAN PEMROGRAMAN 2**

**MODUL 12  
PENGURUTAN DATA**



**Disusun Oleh :**

**Afra Lintang Maharani (2311102258)**

**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

Pengurutan data adalah proses penyusunan elemen-elemen yang ada dalam sebuah himpunan, daftar, atau deret bilangan. Tujuannya adalah untuk mempermudah pencarian data, analisis data, dan pengolahan data. Pengurutan data memiliki berbagai macam metode, di antaranya

- Insertion Sort:

Insertion sort adalah metode pengurutan data dengan algoritma yang mengurutkan elemen-elemen dengan cara menyisipkan setiap elemen ke posisi yang tepat pada daftar yang sudah terurut. Proses ini dilakukan dengan membandingkan elemen yang akan disisipkan dengan elemen-elemen sebelumnya, lalu menempatkannya di posisi yang sesuai.

- Selection Sort:

Selection sort adalah metode pengurutan data yang mengurutkannya dengan cara memilih elemen terkecil pada daftar yang belum terurut, kemudian menukarnya dengan elemen pertama pada daftar tersebut. Proses penukaran ini akan berlanjut sampai daftar yang ada menjadi daftar yang terurut.

Selain kedua metode di atas, masih ada beberapa metode pengurutan data lainnya, seperti

- Bubble Sort
- Quick Sort
- Merge Sort

Pengurutan data sering digunakan dalam berbagai implementasi, seperti dalam pengolahan data dan algoritma pencarian. Dengan mengurutkan data, proses pencarian dan analisis data menjadi lebih efisien dan mudah dilakukan. Metode pengurutan yang dipilih akan bergantung pada kebutuhan, ukuran data, dan karakteristik data yang akan diurutkan.

## II. GUIDED

### 1. Soal Studi Case

Buatlah sebuah program beserta fungsi yang digunakan untuk menghitung nilai faktorial dan permutasi.

**Masukan** terdiri dari dua buah bilangan positif a dan b.

**Keluaran** berupa sebuah bilangan bulat yang menyatakan nilai a permutasi b apabila  $a \geq b$  atau b permutasi a untuk kemungkinan yang lain.

#### 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 1
        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()
    }
}

```

## Output

```

PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\modul pengu
data\Gui1.go"
Masukkan jumlah daerah kerabat(n) : 2

Masukkan jumlah nomor rumah kerabat untuk daerah 1 : 2
masukkan 2 nomor rumah kerabat : 43 34
nomor rumah terurut untuk daerah 1 : 3443

Masukkan jumlah nomor rumah kerabat untuk daerah 2 : 5
masukkan 5 nomor rumah kerabat : 1 3 56 7 98
nomor rumah terurut untuk daerah 2 : 1375698
PS D:\Dokument\Semester 2\.vscode>

```

## Deskripsi

Program ini menggunakan algoritma Selection Sort untuk mengurutkan nomor-nomor rumah kerabat pada beberapa daerah. Pertama-tama, program akan meminta pengguna untuk menginputkan jumlah daerah kerabat (n) yang akan diproses.

Untuk setiap daerah, program akan meminta pengguna untuk menginputkan jumlah nomor rumah kerabat (m), lalu meminta pengguna untuk menginputkan m buah nomor rumah tersebut.

Setelah semua input diberikan, program akan memanggil fungsi selectionsort untuk mengurutkan setiap array nomor rumah, dari yang terkecil ke yang terbesar. Algoritma Selection Sort bekerja dengan mencari elemen terkecil dalam array, lalu menukarkannya dengan elemen di posisi awal, kemudian melakukan hal yang sama untuk elemen berikutnya.

Proses pengurutan ini akan dilakukan untuk setiap daerah yang telah diinputkan sebelumnya. Dengan demikian, program akan mengurutkan nomor-nomor rumah untuk masing-masing daerah secara terpisah.

Akhirnya, program akan menampilkan output berupa array nomor rumah yang telah terurut untuk setiap daerah. Hal ini dapat membantu pengguna dengan mudah melihat dan memahami urutan nomor-nomor rumah kerabat mereka di berbagai daerah.

## 2. Soal Studi Case

### 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

```

nomor rumah terurut untuk daerah 2 : 1375698
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\modul pengurutan
data\Gui2.go"
Masukkan data integer (akhiri dengan bilangan negatif):
1 2 3 56 87 -9
Array setelah diurutkan:
1 2 3 56 87
Data berjarak tidak tetap
PS D:\Dokument\Semester 2\.vscode>

```

## Deskripsi Program

Program ini mengimplementasikan algoritma Insertion Sort untuk mengurutkan sebuah array integer yang diinputkan oleh pengguna. Pertama-tama, pengguna diminta memasukkan bilangan-bilangan bulat satu per satu hingga memasukkan bilangan negatif untuk mengakhiri input.

Setelah pengguna selesai memasukkan bilangan, program akan mengurutkan array tersebut menggunakan fungsi `insertionSort()`. Algoritma Insertion Sort bekerja dengan membandingkan setiap elemen dalam array dengan elemen-elemen sebelumnya, lalu menyisipkan elemen tersebut ke posisi yang tepat sehingga array menjadi terurut.

Setelah proses pengurutan selesai, program akan memanggil fungsi `isConstantDifference()` untuk memeriksa apakah selisih antara setiap elemen array adalah sama atau tidak. Jika selisihnya konstan, fungsi ini akan mengembalikan nilai `true`, sebaliknya `false`.

Akhirnya, program akan menampilkan output berupa array yang telah terurut, serta informasi apakah selisih antara elemen-elemen array tersebut memiliki jarak yang tetap atau tidak.

### 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 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

```
//Afra Lintang 2311102258

package main

import (
    "fmt"
    "sort"
)

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)

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

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

        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\Semester3.go\m
data\unguil.go"
Masukkan jumlah daerah kerabat(n) : 3

Masukkan jumlah nomor rumah kerabat untuk daerah 1 : 4
masukkan 4 nomor rumah kerabat : 2 4 6 7
nomor rumah ganjil terurut untuk daerah 1 : 7
nomor rumah genap terurut untuk daerah 1 : 6 4 2

Masukkan jumlah nomor rumah kerabat untuk daerah 2 : 2
masukkan 2 nomor rumah kerabat : 98 7
nomor rumah ganjil terurut untuk daerah 2 : 7
nomor rumah genap terurut untuk daerah 2 : 98

Masukkan jumlah nomor rumah kerabat untuk daerah 3 : 5
masukkan 5 nomor rumah kerabat : 1 45 67 98 113
nomor rumah ganjil terurut untuk daerah 3 : 1 45 67 113
nomor rumah genap terurut untuk daerah 3 : 98
PS D:\Dokument\Semester 2\.vscode>

```

## Deskripsi Program

Program ini dirancang untuk memisahkan dan mengurutkan nomor-nomor rumah kerabat berdasarkan daerah tempat tinggal mereka. Pertama, pengguna akan diminta memasukkan jumlah daerah (n) yang akan diproses. Kemudian, untuk setiap daerah, pengguna akan diminta memasukkan jumlah nomor rumah (m) beserta nomor-nomor rumah tersebut.

Setelah semua input diberikan, program akan mengurutkan nomor-nomor rumah menggunakan fungsi `sort.Ints()`. Selanjutnya, program akan memisahkan nomor-nomor rumah menjadi dua slice yang berbeda - slice `oddNums` untuk nomor ganjil dan `evenNums` untuk nomor genap.

Program akan menampilkan nomor-nomor rumah ganjil yang telah terurut, dari yang terkecil hingga terbesar. Kemudian, program juga akan menampilkan nomor-nomor rumah genap yang telah terurut, dari yang terkecil hingga terbesar. Proses ini akan diulang untuk setiap daerah yang diinputkan sebelumnya. Dengan demikian, pengguna dapat dengan mudah melihat nomor-nomor rumah kerabat mereka yang telah dipisahkan dan diurutkan berdasarkan daerah tempat tinggal.

## 2. Soal Studi Case

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 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.

### Sourcecode

```
//Afra Lintang 2311102258

package main

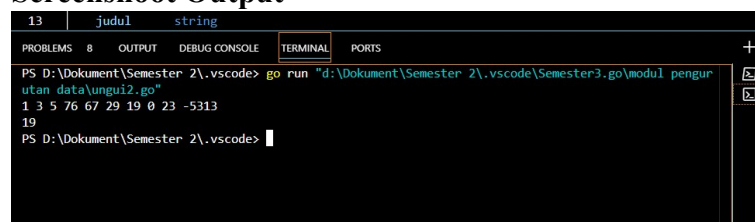
import (
    "fmt"
    "sort"
)

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

    for {
        fmt.Scan(&num)
        if num == -5313 {
            break
        }
        if num == 0 {

            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



```
13 | judul | string
PROBLEMS 8 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\modul_pengur
utan_data\ungui2.go"
1 3 5 76 67 29 19 0 23 -5313
19
PS D:\Dokument\Semester 2\.vscode>
```

## Deskripsi Program

Program diatas dirancang untuk menghitung median dari serangkaian bilangan bulat yang diinputkan pengguna. Pengguna akan diminta memasukkan bilangan-bilangan tersebut satu per satu, dan program akan menyimpannya dalam sebuah slice data.

Ketika pengguna memasukkan angka 0, program akan mulai menghitung median dari bilangan-bilangan yang telah dimasukkan sebelumnya. Untuk menghitung median, program terlebih dahulu akan mengurutkan bilangan-bilangan tersebut menggunakan fungsi `sort.Int()`.

Jika panjang slice data yang ada berjumlah genap, maka median akan dihitung sebagai rata-rata dua bilangan di tengah. Kemudian, program akan mencetak nilai median tersebut.

Jika pengguna memasukkan angka -5313, maka program akan berhenti dan keluar dari perulangan.

Jadi, program ini memberikan kemudahan bagi pengguna untuk menghitung median dari serangkaian bilangan bulat yang mereka masukkan. Dengan menggunakan fungsi-fungsi yang tersedia, program dapat mengurutkan dan menghitung median secara akurat.

## 3. Soal Studi Case

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/mengcil 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.)
```



TELKOM UNIVERSITY

informatics lab

## Sourcecode

```
//Afra Lintang 2311102258

package main

import (
    "fmt"
    "sort"
)

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 CetakFavorit(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 UrutanBuku(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:")
    CetakFavorit(pustaka, n)

    UrutanBuku(&pustaka, n)

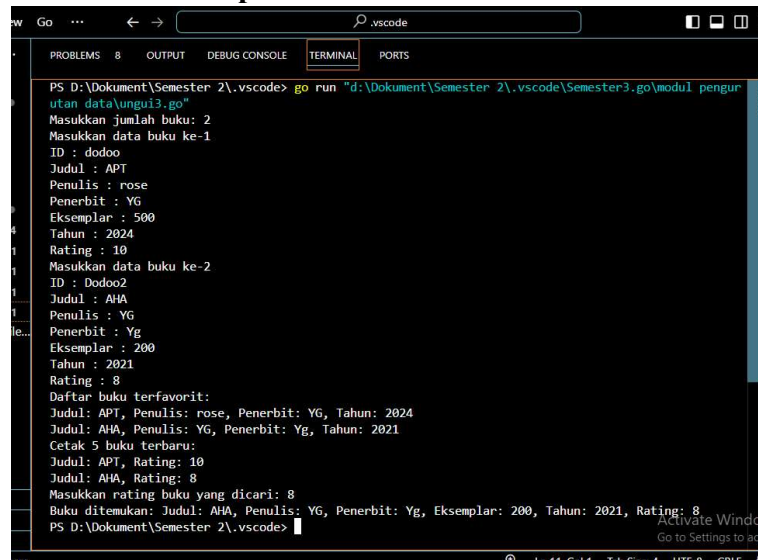
    fmt.Println("Cetak 5 buku terbaru:")
    Cetak5Terbaru(pustaka, n)

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

    CariBuku(pustaka, n, r)
}

```

## Screenshoot Output



```
PS D:\Dokument\Semester 2\.vscode> go run "d:\Dokument\Semester 2\.vscode\Semester3.go\modul pengur
utan data\ungui3.go"
Masukkan jumlah buku: 2
Masukkan data buku ke-1
ID : dodoo
Judul : APT
Penulis : rose
Penerbit : YG
Eksemplar : 500
Tahun : 2024
Rating : 10
Masukkan data buku ke-2
ID : Dodoo2
Judul : AHA
Penulis : YG
Penerbit : Yg
Eksemplar : 200
Tahun : 2021
Rating : 8
Daftar buku terfavorit:
Judul: APT, Penulis: rose, Penerbit: YG, Tahun: 2024
Judul: AHA, Penulis: YG, Penerbit: Yg, Tahun: 2021
Cetak 5 buku terbaru:
Judul: APT, Rating: 10
Judul: AHA, Rating: 8
Masukkan rating buku yang dicari: 8
Buku ditemukan: Judul: AHA, Penulis: YG, Penerbit: Yg, Eksemplar: 200, Tahun: 2021, Rating: 8
PS D:\Dokument\Semester 2\.vscode>
```

## Deskripsi Program

Program di atas berguna untuk mengelola data buku di perpustakaan. Struktur data buku menyimpan informasi penting untuk setiap buku, seperti ID, nama penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Ini memungkinkan program untuk melacak dan mengatur koleksi buku secara efektif.

Pengguna dapat memasukkan data buku baru yang akan ditambahkan ke dalam koleksi. Fungsi CetakFavorit dan UrutanBuku kemudian digunakan untuk mengurutkan buku berdasarkan rating secara menurun, memungkinkan pengguna dengan cepat melihat buku-buku terfavorit. Fungsi cetak5terbaru akan menampilkan 5 buku dengan rating tertinggi, atau semua buku jika jumlahnya kurang dari 5. Ini memberikan gambaran cepat tentang buku-buku terbaru dan populer di perpustakaan.

Terakhir, fungsi caribuku memungkinkan pengguna untuk mencari buku berdasarkan rating dan kriteria lain yang diinginkan. Ini mempermudah pencarian dan akses ke koleksi buku.

Secara keseluruhan, program ini tampaknya dirancang dengan baik untuk mengelola dan mengatur koleksi buku di perpustakaan dengan efisien. Fitur-fiturnya memungkinkan pengguna dengan mudah memasukkan, menelusuri, dan menemukan buku-buku yang mereka cari.



#### **IV. 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](https://www.w3schools.com/dsa/dsa_algo_selectionsort.php)