

LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL 12 - 13
PENGURUTAN DATA



Disusun Oleh :
Muhammad Djoko Susilo / 2311102212

Dosen Pengampu :
Abednego Dwi Septiadi, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

I. DASAR TEORI

Algoritma Selection Sort

Metode ini mencari nilai ekstrim pada sekumpulan data, kemudian meletakkan pada posisi yang seharusnya. Data pada metode ini akan diurut membesar (ascending) dan pada dengan indeks kecil ada di “kiri” dan indeks besar ada di “kanan”. Algoritma dalam metode ini sebagai berikut:

1. Cari nilai terkecil dalam rentang data tersisa
2. Pindah data dari yang terkiri ke rentang data tersisa tersebut
3. Ulangi proses sampai hanya satu data saja

Algoritma Insertion Sort

Menyisipkan suatu nilai pada posisi yang seharusnya, berbeda dengan selection sort pada metode ini tidak harus di cari nilai ekstrimnya terlebih dahulu, tetapi hanya mencari nilai tertentu dan akan ditaruh kepada tempatnya menggunakan *sequential search*.

II. GUIDED

1. Soal Studi Case

Buatlah sebuah program yang dapat digunakan untuk mengurutkan data yang sudah dimasukan oleh User dan akan diurutkan dari yang pertama dengan konsep adalah pengurutan nomor rumah.

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 + 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()
    }
}

```

Screenshot Output

```

PS D:\code\2311102212_Djoko_Modul-12-13> go run "d:\code\2311102212_Djoko_Modul-12-13\guided1.go"
Masukkan jumlah daerah kerabat (n): 2

Masukkan jumlah nomor rumah kerabat untuk daerah 1: 3
Masukkan 3 nomor rumah kerabat: 4 3 5
Nomor rumah terurut untuk daerah 1: 3 4 5

Masukkan jumlah nomor rumah kerabat untuk daerah 2: 5
Masukkan 5 nomor rumah kerabat: 3
4 5 3 7
Nomor rumah terurut untuk daerah 2: 3 3 4 5 7
PS D:\code\2311102212_Djoko_Modul-12-13>

```

Deskripsi Program

Program ini adalah sebuah program yang dapat mengurutkan data menggunakan selection sort, dimulai dari sistem yang meminta pengguna untuk memasukkan jumlah daerah (akan masuk kedalam variabel **n**) yang ingin dimasukan lalu sistem akan mengulang seberapa banyak daerahnya didalam setiap pengulangan tersebut akan dilakukan pengulangan kembali untuk memasukkan jumlah kerabat yang ada di daerah tersebut yang akan dimasukan ke dalam ke variabel **m** setelah itu maka akan langsung di urutkan menggunakan fungsi **selectionSort** yang memberikan nilai **arr** dan **m** dan akan mengembalikan nilai urutnya kembali ke fungsi main/utama.

2. Soal Studi Case

Buatlah sebuah program yang digunakan untuk membaca data integer seperti contoh yang diberikan di bawah ini. Kemudian, urutkan data tersebut menggunakan metode *insertion sort*, dan periksa apakah data yang telah diurutkan memiliki jarak yang sama terhadap data sebelumnya.

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)

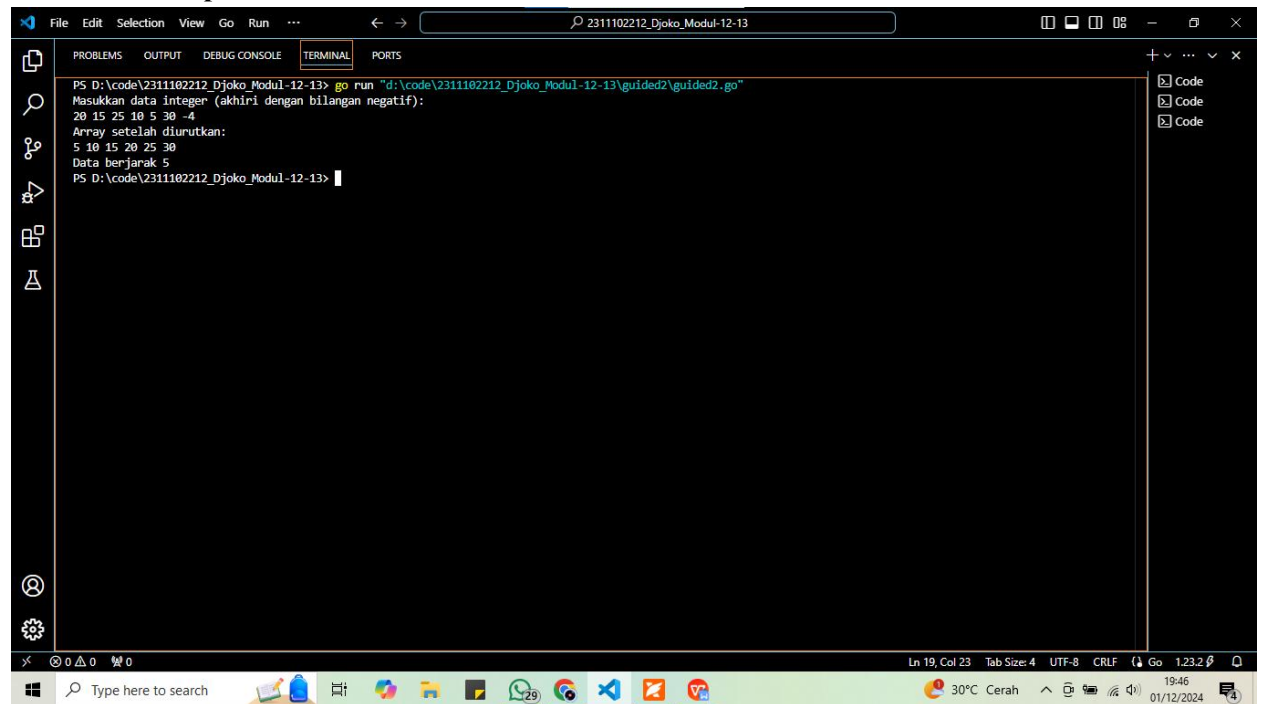
// Periksa apakah selisih elemen tetap
isConstant, difference :=
isConstantDifference(arr, n)

// 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")
}
}

```

Screenshot Output



The screenshot shows the terminal output of a Go program. The program prompts the user to enter integers, and the user enters 20, 15, 25, 10, 5, 30, and -4. The program then displays the sorted array: 5 10 15 20 25 30. It also displays the constant difference: Data berjarak 5. The terminal window is titled '2311102212_Djoko_Modul-12-13' and shows the command 'go run "d:\code\2311102212_Djoko_Modul-12-13\guided2\guided2.go"' being executed.

```
PS D:\code\2311102212_Djoko_Modul-12-13> go run "d:\code\2311102212_Djoko_Modul-12-13\guided2\guided2.go"
Masukkan data integer (akhiri dengan bilangan negatif):
20 15 25 10 5 30 -4
Array setelah diurutkan:
5 10 15 20 25 30
Data berjarak 5
PS D:\code\2311102212_Djoko_Modul-12-13>
```

Deskripsi Program

Program ini adalah sebuah program yang dapat mengurutkan data menggunakan insertion sort. Sistem dimulai dengan meminta pengguna untuk memasukkan serangkaian data integer satu per satu, yang akan terus diterima hingga pengguna memasukkan bilangan negatif untuk menghentikan input. Data yang telah dimasukkan akan diurutkan menggunakan fungsi **insertionSort**, yang menerima array **arr** dan jumlah elemennya **n**. Setelah data diurutkan, program memeriksa apakah selisih antara elemen-elemen dalam array yang terurut memiliki nilai yang konstan menggunakan fungsi **isConstantDifference**. Jika selisihnya konstan, program akan menampilkan nilai selisih tersebut; jika tidak, akan menampilkan pesan bahwa data tidak berjarak tetap.

III. UNGUIDED

1. Soal Studi Case

Buatkan program yang dapat menampilkan sebuah data yang sudah terurut dari nomor rumah yang ganjil terlebih dahulu terurut memebesar dan nomor rumah genap terurut mengecil.

Sourcecode

```
package main

import "fmt"

// Fungsi untuk mengurutkan array menggunakan Selection Sort
func selectionSort(arr []int, ascending bool) {
    n := len(arr)
    i := 1
    for i <= n-1 {
        idxMin := i - 1
        j := i
        for j < n {
            // Kondisi untuk ascending atau descending
            if (ascending && arr[idxMin] < arr[j]) ||
                (!ascending && arr[idxMin] > arr[j]) {
                idxMin = j
            }
            j = j + 1
        }
        // Tukar elemen
        t := arr[idxMin]
        arr[idxMin] = arr[i-1]
        arr[i-1] = t
        i = i + 1
    }
}

// Fungsi untuk memisahkan nomor ganjil dan genap,
// serta mengurutkan secara manual
func processRumahManual(arr []int) []int {
    var ganjil, genap []int

    // Memisahkan angka ganjil dan genap
    for _, num := range arr {
        if num%2 == 0 {
            genap = append(genap, num)
        } else {
            ganjil = append(ganjil, num)
        }
    }

    // Mengurutkan ganjil secara menurun (descending)
    // dengan selection sort
}
```



```

        selectionSort(ganjil, false)

        // Mengurutkan genap secara menaik (ascending)
        dengan selection sort
        selectionSort(genap, true)

        // Menggabungkan ganjil dan genap
        return append(ganjil, genap...)
    }

func main() {
    var banyakDaerah2311102212 int
    fmt.Print("Masukan Berapa Daerah: ")
    fmt.Scan(&banyakDaerah2311102212)

    // Menggunakan slice dua dimensi untuk menyimpan
    data rumah
    rumah := make([][]int, banyakDaerah2311102212)

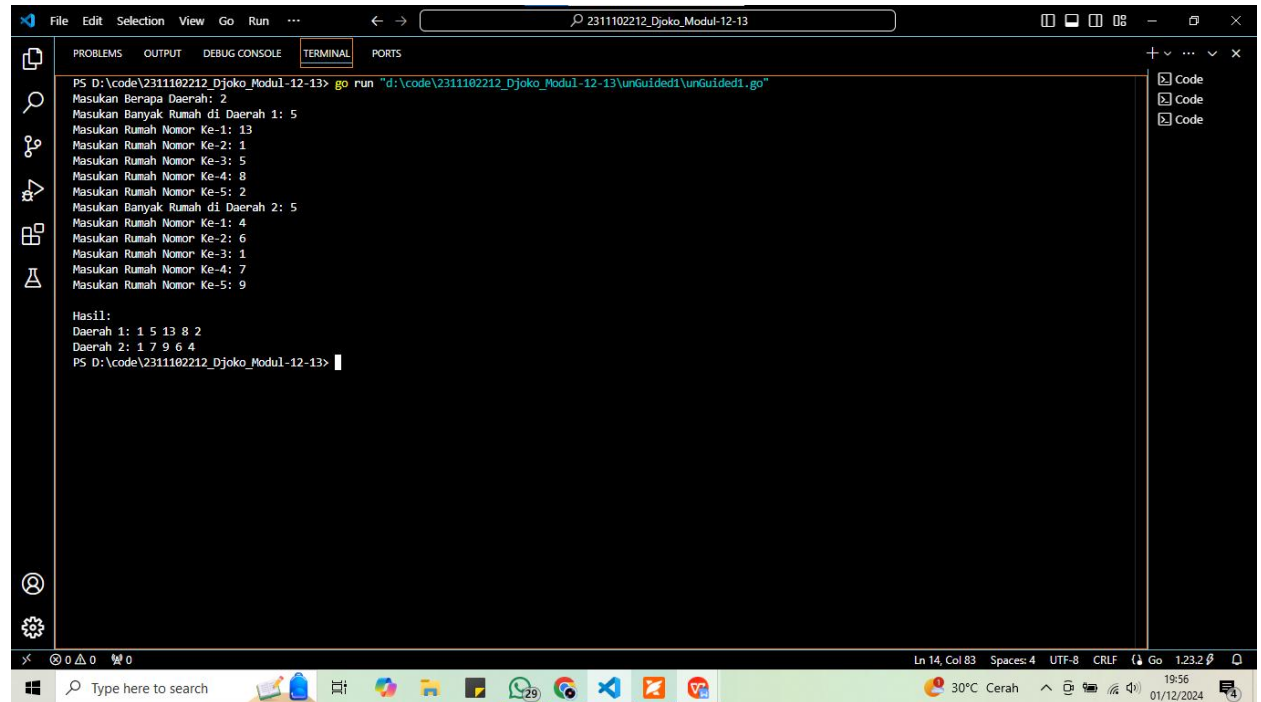
    for i := 0; i < banyakDaerah2311102212; i++ {
        var banyakRumah int
        fmt.Printf("Masukan Banyak Rumah di Daerah %d: ", i+1)
        fmt.Scan(&banyakRumah)

        // Membuat slice untuk menyimpan nomor rumah di
        daerah ini
        rumah[i] = make([]int, banyakRumah)
        for j := 0; j < banyakRumah; j++ {
            fmt.Printf("Masukan Rumah Nomor Ke-%d: ",
j+1)
            fmt.Scan(&rumah[i][j])
        }
    }

    // Memproses dan menampilkan hasil
    fmt.Println("\nHasil:")
    for i, daerah := range rumah {
        hasil := processRumahManual(daerah)
        fmt.Printf("Daerah %d: ", i+1)
        for _, num := range hasil {
            fmt.Printf("%d ", num)
        }
        fmt.Println()
    }
}

```

Screenshot Output



```
PS D:\code\2311102212_Djoko_Modul-12-13> go run "d:\code\2311102212_Djoko_Modul-12-13\unGuided1\unGuided1.go"
Masukan Berapa Daerah: 2
Masukan Banyak Rumah di Daerah 1: 5
Masukan Rumah Nomor Ke-1: 13
Masukan Rumah Nomor Ke-2: 1
Masukan Rumah Nomor Ke-3: 5
Masukan Rumah Nomor Ke-4: 8
Masukan Rumah Nomor Ke-5: 2
Masukan Banyak Rumah di Daerah 2: 5
Masukan Rumah Nomor Ke-1: 4
Masukan Rumah Nomor Ke-2: 6
Masukan Rumah Nomor Ke-3: 1
Masukan Rumah Nomor Ke-4: 7
Masukan Rumah Nomor Ke-5: 9

Hasil:
Daerah 1: 1 5 13 8 2
Daerah 2: 1 7 9 6 4
PS D:\code\2311102212_Djoko_Modul-12-13>
```

Deskripsi Program

Program ini adalah sebuah program yang dapat mengurutkan data menggunakan **selection sort**. Program dimulai dengan meminta pengguna untuk memasukkan jumlah daerah (disimpan dalam variabel `banyakDaerah2311102212`). Sistem akan mengulang sebanyak jumlah daerah yang dimasukkan, dan di dalam setiap pengulangan, pengguna diminta untuk memasukkan jumlah rumah dalam setiap daerah (disimpan dalam variabel `banyakRumah`) beserta nomor-nomor rumah tersebut.

Selanjutnya, nomor rumah akan diproses dalam fungsi `processRumahManual` yang memisahkan nomor **ganjil** dan **genap**, mengurutkan nomor ganjil secara **menaik (ascending)** dan nomor genap secara **menurun (descending)** menggunakan algoritma **selection sort**. Setelah diproses, hasil pengurutan nomor rumah akan ditampilkan per daerah

2. Soal Studi Case

Buatlah sebuah program yang digunakan untuk membaca data integer seperti contoh yang diberikan di bawah ini. Kemudian, urutkan data (menggunakan metode **selection sort**), dan hitung nilai mediannya dari data yang telah diurutkan.

Sourcecode

```
package main

import "fmt"

// Fungsi untuk mengurutkan data menggunakan selection sort
func selectionSort(data []int) {
    n := len(data)
    i := 1
    for i <= n-1 {
        idxMin := i - 1
        j := i
        for j < n {
            if data[idxMin] > data[j] {
                idxMin = j
            }
            j = j + 1
        }
        // Tukar elemen
        t := data[idxMin]
        data[idxMin] = data[i-1]
        data[i-1] = t
        i = i + 1
    }
}

// Fungsi untuk menghitung median
func calculateMedian(data []int) int {
    // Mengurutkan data dengan selection sort
    selectionSort(data)

    // Menentukan median
    n := len(data)
    if n%2 == 1 {
        return data[n/2] // Jika ganjil, median adalah
        elemen tengah
    } else {
        return (data[n/2-1] + data[n/2]) / 2 // Jika
        genap, median adalah rata-rata dua elemen tengah
    }
}

func main() {
    var input2311102212 int
    var data []int
    var i int
    i = 1
    for {
        fmt.Printf("Masukan data ke-%d = ", i)
```

```

        fmt.Scan(&input2311102212)

        // Jika input adalah -5313, hentikan program
        if input2311102212 == -5313 {
            break
        }

        // Jika input adalah 0, hitung dan cetak median
        if input2311102212 == 0 {
            if len(data) == 0 {
                fmt.Println("Tidak ada data")
            } else {
                median := calculateMedian(data)
                fmt.Println(median)
            }
        } else {
            // Masukkan angka ke dalam data (selain 0 dan
-5313)
            data = append(data, input2311102212)
        }
        i++
    }
}

```

Screenshot Output

```

PS D:\code\2311102212_Djoko_Modul-12-13> go run "d:\code\2311102212_Djoko_Modul-12-13\unGuided2\unGuided2.go"
Masukan data ke-1 = 5
Masukan data ke-2 = 1
Masukan data ke-3 = 7
Masukan data ke-4 = 4
Masukan data ke-5 = 5
Masukan data ke-6 = 0
5
Masukan data ke-7 = 4
Masukan data ke-8 = 13
Masukan data ke-9 = 45
Masukan data ke-10 = 34
Masukan data ke-11 = 32
Masukan data ke-12 = 54
Masukan data ke-13 = 23
Masukan data ke-14 = 9
Masukan data ke-15 = 0
9
Masukan data ke-16 = -5313
PS D:\code\2311102212_Djoko_Modul-12-13>

```

Deskripsi Program

Program ini adalah sebuah program yang dapat mengurutkan data menggunakan **selection sort**. Program dimulai dengan meminta pengguna untuk memasukkan data integer satu per satu, yang akan disimpan dalam sebuah array. Jika pengguna memasukkan angka **0**, program akan mengurutkan data yang telah dimasukkan menggunakan fungsi `selectionSort` dan menghitung nilai median menggunakan fungsi `calculateMedian`. Jika pengguna memasukkan angka **-5313**, program akan berhenti. Program menggunakan fungsi `selectionSort` untuk mengurutkan array secara menaik, dan hasil median akan dikembalikan serta ditampilkan ke layar melalui fungsi utama `main`.

3. Soal Studi Case

Buatlah sebuah program yang digunakan untuk mengelola perpustakaan, dengan fitur sebagai berikut:

1. Input data buku yang mencakup ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, dan Rating.
2. Urutkan data buku berdasarkan rating secara menurun menggunakan metode `insertion sort`.
3. Tampilkan informasi buku dengan rating tertinggi.
4. Tampilkan 5 buku dengan rating tertinggi.
5. Cari buku berdasarkan rating yang diinputkan oleh pengguna, dengan menggunakan metode `binary search`.

Sourcecode

```
package main

import "fmt"

type Buku struct {
    ID, Judul, Penulis, Penerbit string
    Eksemplar, Tahun, Rating int
}

type Perpustakaan struct {
    Pustaka []Buku
    NPustaka int
}

// Fungsi untuk membaca data buku
func DaftarkanBuku(p *Perpustakaan, n int) {
    p.NPustaka = n
    p.Pustaka = make([]Buku, n)
    for i := 0; i < n; i++ {
        fmt.Printf("Masukkan data buku ke-%d (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):%n", i+1)
        fmt.Scan(&p.Pustaka[i].ID,
```

```

        &p.Pustaka[i].Judul, &p.Pustaka[i].Penulis,
        &p.Pustaka[i].Penerbit, &p.Pustaka[i].Eksemplar,
        &p.Pustaka[i].Tahun, &p.Pustaka[i].Rating)
    }
}

// Fungsi untuk mencetak buku favorit
func CetakTerfavorit(p Perpustakaan) {
    maxRating := -1
    var favorit Buku
    for _, buku := range p.Pustaka {
        if buku.Rating > maxRating {
            maxRating = buku.Rating
            favorit = buku
        }
    }
    fmt.Printf("Buku Terfavorit: %s - %s (%d),
Rating: %d\n", favorit.Judul, favorit.Penerbit,
favorit.Tahun, favorit.Rating)
}

// Fungsi pengurutan menggunakan metode insertion sort
func insertionSort(p *Perpustakaan) {
    for i := 1; i < p.NPustaka; i++ {
        temp := p.Pustaka[i]
        j := i
        // Geser elemen-elemen yang lebih kecil dari
temp ke kanan
        for j > 0 && temp.Rating > p.Pustaka[j-
1].Rating {
            p.Pustaka[j] = p.Pustaka[j-1]
            j = j - 1
        }
        p.Pustaka[j] = temp
    }
}

// Fungsi untuk mencetak 5 buku dengan rating tertinggi
func Cetak5Terbaru(p Perpustakaan) {
    fmt.Println("5 Buku dengan Rating Tertinggi:")
    for i := 0; i < p.NPustaka && i < 5; i++ {
        fmt.Printf("%s - %s (%d), Rating: %d\n",
p.Pustaka[i].Judul, p.Pustaka[i].Penerbit,
p.Pustaka[i].Tahun, p.Pustaka[i].Rating)
    }
}

// Fungsi untuk mencari buku berdasarkan rating
menggunakan pencarian biner
func CariBuku(p Perpustakaan, rating int) {

```

```

        left, right := 0, p.NPustaka-1
        for left <= right {
            mid := (left + right) / 2
            if p.Pustaka[mid].Rating == rating {
                fmt.Printf("Ditemukan buku: %s - %s
(%d), Rating: %d\n", p.Pustaka[mid].Judul,
p.Pustaka[mid].Penerbit, p.Pustaka[mid].Tahun,
p.Pustaka[mid].Rating)
                return
            } else if p.Pustaka[mid].Rating < rating {
                right = mid - 1
            } else {
                left = mid + 1
            }
        }
        fmt.Println("Tidak ada buku dengan rating seperti
itu.")
    }

func main() {
    var perpustakaan2311102212 Perpustakaan
    var n, rating int

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

    // Input data buku
    DaftarkanBuku(&perpustakaan2311102212, n)

    // Urutkan buku berdasarkan rating
    insertionSort(&perpustakaan2311102212)

    // Cetak buku favorit
    CetakTerfavorit(perpustakaan2311102212)

    // Cetak 5 buku terbaru
    Cetak5Terbaru(perpustakaan2311102212)

    // Cari buku berdasarkan rating
    fmt.Print("Masukkan rating yang dicari: ")
    fmt.Scan(&rating)
    CariBuku(perpustakaan2311102212, rating)
}

```

Screenshot Program

```
PS D:\code\2311102212_Djoko_Modul-12-13> go run "d:\code\2311102212_Djoko_Modul-12-13\unGuided3\unGuided3.go"
Masukkan jumlah buku: 5
Masukkan data buku ke-1 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
1 IkanDikolam Joko RY2Film 23 2023 4
Masukkan data buku ke-2 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
2 UmurSiKakek Ujang BNDBOOK 14 2024 3
Masukkan data buku ke-3 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
3 KucingTerbang Cian BGRBuku 2 2024 3
Masukkan data buku ke-4 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
4 ObengJatuh Sou ChinFilm 17 2024 4
Masukkan data buku ke-5 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
5 SiApung Olong JouBook 22 2024 3
Buku Terfavorit: IkanDikolam - RY2Film (2023), Rating: 4
5 Buku dengan Rating Tertinggi:
IkanDikolam - RY2Film (2023), Rating: 4
ObengJatuh - ChinFilm (2024), Rating: 4
UmurSiKakek - BNDBOOK (2024), Rating: 3
KucingTerbang - BGRBuku (2024), Rating: 3
SiApung - JouBook (2024), Rating: 3
Masukkan rating yang dicari: 3
Ditemukan buku: UmurSiKakek - BNDBOOK (2024), Rating: 3
PS D:\code\2311102212_Djoko_Modul-12-13>
```

Deskripsi Program

Program ini adalah sebuah sistem manajemen **perpustakaan** yang memungkinkan pengguna untuk:

1. **Mendaftarkan buku** dengan memasukkan detail seperti ID, Judul, Penulis, Penerbit, Jumlah Eksemplar, Tahun Terbit, dan Rating.
2. **Mengurutkan buku berdasarkan rating** secara menurun menggunakan **metode insertion sort**.
3. **Menampilkan buku dengan rating tertinggi (favorit)**.
4. **Menampilkan 5 buku dengan rating tertinggi**.
5. **Mencari buku berdasarkan rating** menggunakan metode **binary search**.