LAPORAN PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 2

MODUL XII & XIII

PENGURUTAN DATA

Disusun Oleh :

Naya Putwi Setiasih / 2311102155

S1 11 IF - 05

Dosen Pengampu :

Arif Amrulloh, S.Kom., M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. DASAR TEORI

- **Ide Algoritma Selection Sort**
  Pengurutan ini untuk mencari nilai ekstrim pada sekumpulan data, meletakkan pada posisi yang seharusnya.
  - Cari nilai terkecil di dalam rentangdata tersisa.
  - Pindahkan / tukar tempat dengan data yang berada pada posisi paling kiri pada rentang data tersisa tersebut.
  - Ulangi proses ini sampai tersisa hanya satu data saja.

  Algoritma ini dikenal Selection Sort, melibatkan dua proses yaitu pencarian indeks nilai ekstrim dan proses pertukaran dua nilai atau swap.

	Notasi Algoritma	Notasi dalam bahasa Go
1	i ← 1	i = 1
2	while i <= n-1 do	for i <= n-1 {
3	idx_min ← i – 1	idx_min = i – 1
4	j ← i	j = i
5	while j < n do	for j < n {
6	if a[idx_min] > a[j] then	if a[idx_min] > a[j] {
7	idx_min ← j	idx_min = j
8	endif	}
9	j ← j + 1	j = j + 1
10	endwhile	}
11	t ← a[idx_min]	t = a[idx_min]
12	a[idx_min] ← a[i-1]	a[idx_min] = a[i-1]
13	a[i-1] ← t	a[i-1] = t
14	i ← i + 1	i = i + 1
15	endwhile	}

- **Algortima Selection Sort**
  Algoritma selection sort untuk mengurutkan array bertipe data bilangan bulat secara membesar atau ascending.
- **Ide Algoritma Insertion Sort**
  Pengurutan insertion untuk menyimpan suatu nilai pada posisi yang seharusnya. Berbeda dengan pengurutan seleksi, yang mana pada pengurutan ini tidak dilakukan pencarian nilai ekstrim terlebih dahulu, cukup memilih suatu nilai tertentu kemudian mencari posisinya secara sequential search. Data akan diurut mengecil (descending), dan data dengan indeks kecil ada di kiri dan indeks besar ada di kanan.
  - Untuk suatu data yang belum terurut dan sejumlah data yang sudah diurutkan. Geser data yang sudah terurut ke sebelah kanan, sehingga ada ruang kosong untuk memasukkan data yang belum terurut ke dalam data yang sudah terurut dan tetap menjaga keterurutan.
  - Ulangi proses tersebut untuk setiap data yang belum terurut terhadap rangkaian data yang sudah terurut.
- **Algoritma Insertion Sort**
  Algoritma insertion sort untuk mengurutkan array bertipe data bilangan bulat secara mengecil atau descending.

II. GUIDED

1.

Soal Studi Case

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.

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)

    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

```
Masukkan jumlah daerah kerabat (n): 3

Masukkan jumlah nomor rumah kerabat untuk daerah 1: 3
Masukkan 3 nomor rumah kerabat : 2 7 9
Nomor rumah terurut untuk daerah 1 : 2 7 9

Masukkan jumlah nomor rumah kerabat untuk daerah 2: 3
Masukkan 3 nomor rumah kerabat : 5 2 1
Nomor rumah terurut untuk daerah 2 : 1 2 5

Masukkan jumlah nomor rumah kerabat untuk daerah 3: 3
Masukkan 3 nomor rumah kerabat : 4 5 7 1
Nomor rumah terurut untuk daerah 3 : 4 5 7
PS D:\Alpro 2>
```

Deskripsi Program

Program ini ditulis dalam bahasa Go dan berfungsi untuk mengurutkan nomor rumah kerabat berdasarkan input pengguna. Pengguna diminta untuk memasukkan jumlah daerah kerabat (“n”), dan untuk setiap daerah, mereka harus memberikan jumlah nomor rumah kerabat (“m”) serta nomor – nomor tersebut satu per satu. Setelah semua data dimasukkan, program akan menggunakan algoritma Selection Sort untuk mengurutkan nomor rumah secara ascending. Proses sorting melibatkan dua loop: loop luar yang iterasi sebanyak “n-1” kali dan loop dalam yang mencari indeks elemen terkecil dari sisa elemen dalam array. etelah menemukan elemen

terkecil, program akan menukarnya dengan elemen di posisi saat ini. Hasil akhir adalah urutan nomor rumah yang terurut untuk setiap daerah, yang kemudian dicetak ke layar, memberikan pengguna informasi yang jelas mengenai urutan nomor rumah kerabat mereka.

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:\Alpro 2> go run "d:\Alpro 2\Modul 12\tempCodeRunnerFile.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
```

Deskripsi Program

Program ini ditulis dalam Bahasa Go dan bertujuan untuk mengurutkan serangkaian angka bulat yang dimasukkan oleh pengguna. Pengguna diminta untuk memasukkan data integer secara berurutan hingga mereka memasukkan bilangan negatif, yang menandakan akhir dari input. Setelah data dikumpulkan, program akan menggunakan fungsi Insertion Sort untuk mengurutkan array. Proses pengurutan dilakukan dengan cara membandingkan elemen saat ini dengan elemen sebelumnya, dan jika elemen saat ini lebih kecil, elemen yang lebih besar akan digeser ke kanan hingga posisi yang tepat ditemukan untuk elemen tersebut. Setelah array terurut, program kemudian memeriksa apakah selisih antara elemen-elemen dalam array tetap konstan menggunakan fungsi isConstantDifference. Fungsi ini menghitung selisih antara dua elemen berturut-turut dan memverifikasi konsistensinya di seluruh array. Hasil akhir akan menampilkan array yang telah diurutkan dan memberikan informasi apakah selisih antar elemen tetap atau tidak, serta nilai selisih jika konstan.

III. UNGUIDED

1. Belakangan diketahui ternyata Herculas itu tidak berani menyebrang jalan, maka selalu diusahakan agar hanya menyebrang jalan sesedikit mungkin, hanya diujung jalan. Karena nomor rumah sisi kiri jalan selalu ganjil dan sisi kanan jalan selalu genpa, 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.
   Masukan : masih persis sama seperti sebelumnya.
   Keluaran : terdiri dari n baris, yaitu rangkaian rumah kerabatnya terurut membesar untuk nommor ganjil, diikuti dengan terurut menegcil untuk nomor genap, di masing – masing daerah.
   Sourcecode

```
package main

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

// Fungsi untuk memisahkan bilangan ganjil dan genap
dari array
func pisahkanGanjilGenap(arr []int) ([]int, []int) {
    var ganjil, genap []int
    for _, num := range arr {
        if num%2 == 0 {
            genap = append(genap, num)
        } else {
            ganjil = append(ganjil, num)
        }
    }
    return ganjil, genap
}

func main() {
    scanner := bufio.NewScanner(os.Stdin)

    fmt.Println("Masukkan jumlah daerah:")
    scanner.Scan()
    jumlahDaerah, _ := strconv.Atoi(scanner.Text())

    for i := 1; i <= jumlahDaerah; i++ {
        fmt.Printf("Masukkan angka untuk daerah %d
(pisahkan dengan spasi):\n", i)
        scanner.Scan()
```

```
        input := scanner.Text()

        // Konversi input menjadi array integer
        strNumbers := strings.Fields(input)
        var numbers []int
        for _, strNum := range strNumbers {
            num, _ := strconv.Atoi(strNum)
            numbers = append(numbers, num)
        }

        // Pisahkan ganjil dan genap
        ganjil, genap := pisahkanGanjilGenap(numbers)

        // Urutkan ganjil secara menaik
        sort.Slice(ganjil, func(i, j int) bool {
            return ganjil[i] < ganjil[j]
        })

        // Urutkan genap secara menurun
        sort.Slice(genap, func(i, j int) bool {
            return genap[i] > genap[j]
        })

        // Cetak hasil
        fmt.Printf("Output untuk daerah %d:\n", i)
        for _, num := range ganjil {
            fmt.Printf("%d ", num)
        }
        for _, num := range genap {
            fmt.Printf("%d ", num)
        }
        fmt.Println()
    }
}
```

Screenshoot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 12\unguided 2.go"
Masukkan jumlah daerah:
3
Masukkan angka untuk daerah 1 (pisahkan dengan spasi):
5 2 1 7 9 13
Output untuk daerah 1:
1 5 7 9 13 2
Masukkan angka untuk daerah 2 (pisahkan dengan spasi):
6 9 2 6 8 1
Output untuk daerah 2:
1 9 8 6 6 2
Masukkan angka untuk daerah 3 (pisahkan dengan spasi):
5 3 9 8 2 5
Output untuk daerah 3:
3 5 5 9 8 2
```

Deskripsi Program

Program yang ditulis dalam Bahasa Go ini adalah aplikasi sederhana untuk memisahkan dan mengurutkan angka ganjil dan genap dari input pengguna. Program dimulai dengan meminta pengguna untuk memasukkan jumlah daerah yang akan diproses. Untuk setiap daerah, pengguna diminta untuk memasukkan serangkaian angka yang dipisahkan oleh spasi.

Setelah menerima input, program akan mengonversi string input menjadi array integer. Kemudian, fungsi “pisahkanGanjilGenap” digunakan untuk memisahkan angka-angka tersebut menjadi dua kategori: angka ganjil dan angka genap. Angka ganjil akan diurutkan dalam urutan menaik, sementara angka genap akan diurutkan dalam urutan menurun menggunakan fungsi “sort.Slice”.

Setelah proses pemisahan dan pengurutan selesai, program mencetak hasilnya untuk setiap daerah. Hasil yang ditampilkan adalah daftar angka ganjil diikuti oleh daftar angka genap, masing-masing sesuai dengan urutan yang telah ditentukan.

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 genop, 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."

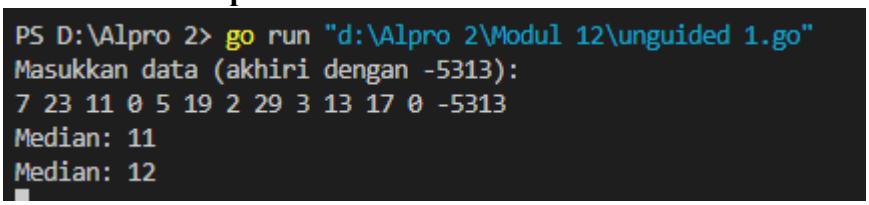
Buatlah program median yang mencetak nilai median terhadap seluruh data yang sudah terbaca, jika data yang dibaca saat itu adalah 0.
Masukan berbentuk rangkaian bilangan bulat. Masukan tidak akan berisi lebih dari 1000000 data, tidak termasuk bilangan 0. Data 0 merupakan tanda bahwa median harus dicetak, 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 bredua, 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 bukkan 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.

Sourcecode

```
package main

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

func findMedian(data []int) int {
    n := len(data)
    if n == 0 {
        return 0
    }
    if n%2 == 1 {
        return data[n/2]
    }
    return (data[n/2-1] + data[n/2]) / 2
}

func main() {
    var numbers []int
    scanner := bufio.NewScanner(os.Stdin)

    fmt.Println("Masukkan data (akhiri dengan -5313):")
    for scanner.Scan() {
        inputLine := scanner.Text()
```

```
        if inputLine == "" {
            break
        }
        // Memisahkan angka dengan spasi
        inputs := strings.Fields(inputLine)
        for _, s := range inputs {
            num, err := strconv.Atoi(s)
            if err != nil {
                fmt.Println("Input tidak valid:", s)
                return
            }
            if num == -5313 {
                return
            } else if num == 0 {
                // Ketika 0 ditemukan, hitung median
                sort.Ints(numbers)
                fmt.Println("Median:",
findMedian(numbers))
            } else {
                // Tambahkan angka ke dalam array
                numbers = append(numbers, num)
            }
        }
    }
}
```

Screenshoot Output



Deskripsi Program

Program yang ditulis dalam Bahasa Go ini adalah aplikasi sederhana untuk menerima dan memproses input angka dari pengguna, serta menghitung median dari angka-angka yang dimasukkan. Pengguna diminta untuk memasukkan data numerik secara berurutan, dengan setiap angka dipisahkan oleh spasi. Proses input berlanjut hingga pengguna memasukkan angka khusus, yaitu -5313, yang menandakan akhir dari input. Selama proses input, jika pengguna memasukkan angka 0, program akan menghitung dan menampilkan median dari semua angka yang telah dimasukkan hingga saat itu. Untuk menghitung median, program pertama-tama mengurutkan daftar angka menggunakan fungsi “sort.Ints” kemudian menentukan nilai median berdasarkan jumlah elemen dalam daftar—apakah genap atau ganjil. Jika input tidak valid, program akan memberikan peringatan dan berhenti

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 berisı 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"
)

// Fungsi rekursif untuk mencetak faktor
func cetakFaktor(n int, i int) {
    if i > n {
        return // Kondisi dasar: jika i lebih besar dari
n, hentikan rekursi
    }

    // Cek apakah i adalah faktor dari n
    if n%i == 0 {
        fmt.Print(i, " ") // Cetak faktor
    }

    // Panggil fungsi untuk angka berikutnya
    cetakFaktor(n, i+1)
}

func main() {
    var n int
    fmt.Print("Masukkan bilangan bulat positif N: ")
    fmt.Scan(&n)

    fmt.Printf("Faktor dari %d adalah: ", n)
    cetakFaktor(n, 1) // Memulai pencetakan dari 1
    fmt.Println() // Pindah ke baris baru setelah
selesai
}
```

Screenshoot Output

```
PS D:\Alpro 2> go run "d:\Alpro 2\Modul 12\unguided 3.go"
Masukkan jumlah buku: 3
Masukkan data buku ke-1 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
1 HarryPotter J.K.Rowling Bloomsbury 50 1997 9
Masukkan data buku ke-2 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
2 LOTR Tolkien HarperCollins 30 1954 8
Masukkan data buku ke-3 (ID, Judul, Penulis, Penerbit, Eksemplar, Tahun, Rating):
3 AtomicHabits JamesClear Penguin 20 2018 10

Mengurutkan buku berdasarkan rating...
Buku Terfavorit:
ID: 3, Judul: AtomicHabits, Penulis: JamesClear, Penerbit: Penguin, Tahun: 2018, Rating: 10
5 Buku dengan Rating Tertinggi:
ID: 3, Judul: AtomicHabits, Penulis: JamesClear, Penerbit: Penguin, Tahun: 2018, Rating: 10
ID: 1, Judul: HarryPotter, Penulis: J.K.Rowling, Penerbit: Bloomsbury, Tahun: 1997, Rating: 9
ID: 2, Judul: LOTR, Penulis: Tolkien, Penerbit: HarperCollins, Tahun: 1954, Rating: 8

Masukkan rating buku yang ingin dicari: 8
Buku Ditemukan:
ID: 2, Judul: LOTR, Penulis: Tolkien, Penerbit: HarperCollins, Tahun: 1954, Rating: 8
```

Deskripsi Program

Program di atas adalah aplikasi yang ditulis dalam bahasa pemrograman Go, yang bertujuan untuk mengelola dan menampilkan informasi mengenai buku – buku yang terdaftar. Program ini memungkinkan pengguna untuk memasukkan data buku, termasuk ID, judul, penulis, penerbit, jumlah eksemplar, tahun terbit, dan rating. Setelah data dimasukkan, program akan mengurutkan buku berdasarkan rating menggunakan algoritma Insertion Sort. Selanjutnya, program menampilkan buku dengan rating tertinggi sebagai “buku favorit” dan juga menampilkan lima buku dengan rating tertinggi. Selain itu, pengguna dapat mencari buku berdasarkan rating tertentu menggunakan metode pencarian biner. Jika buku dengan rating yang dicari ditemukan, informasi lengkapnya akan ditampilkan; jika tidak, program akan memberi tahu bahwa tidak ada buku dengan rating tersebut.

DAFTAR PUSTAKA

Asisten Praktikum, “Modul 12 & 13 Pengurutan Data", Learning Management System, 2024.