LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

JOBSHEET 10 PERTEMUAN 13 DOULE LINKED LIST

Dosen Pengampu : Triana Fatmawati, S.T., M.T.

Muhammad Afiq Firdaus

2341760189 / 21

SIB 1 E

PRODI D-IV SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

Praktikum 1

Buat paket baru dengan nama doublelinkedlists
Buat class di dalam paket tersebut dengan nama Node
Serta isikan code sesuai jobsheet

C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Prakt 13\doublelinkedlists\Node.java

```java
public class Node{
    int data;
    Node prev, next;

    Node(Node prev, int data, Node next){
     this.prev=prev;
     this.data=data;
     this.next=next;
    }
}
```

Buatlah sebuah class baru bernama DoubleLinkedLists pada package yang sama dengan node
Pada class DoubleLinkedLists tersebut, deklarasikan atribut sesuai dengan diagram class di atas.

doublelinkedlists > DoubleLinkedLists.java > DoubleLinkedLists >

```java
package doublelinkedlists;
public class DoubleLinkedLists {
    Node head;
    int size;
```

Selajuntnya, buat konstruktor pada class DoubleLinkedLists

```java
    public DoubleLinkedLists(){
        head = null;
        size = 0;
    }
```

Buat method isEmpty(). Method ini digunakan untuk memastikan kondisi linked list kosong

```java
    public boolean isEmpty(){
        return head == null;
    }
```

Kemudian, buat method addFirst(). Method ini akan menjalankan penambahan data di bagian depan linked list.

```java
public void addFirst(int item){
    if (isEmpty()){
        head = new Node(prev:null, item, next:null);
    }else{
        Node newNode = new Node(prev:null, item, head);
        head.prev = newNode;
        head = newNode;
    }
    size++;
}
```

Selain itu pembuatan method addLast() akan menambahkan data pada bagian belakang linked list

```java
public void addLast(int item){
    if (isEmpty()){
        addFirst(item);
    }else{
        Node current = head;
        while (current.next != null){
            current = current.next;
        }
        Node newNode = new Node (current, item, next:null);
        current.next = newNode;
        size++;
    }
}
```

Untuk menambahkan data pada posisi yang telah ditentukan dengan indeks, dapat dibuat dengan method add(int item, int index)

```java
public void add(int item, int index) throws Exception{
    if(isEmpty()){
        addFirst(item);
    }else if (index < 0 || index > size){
        throw new Exception(message:"Nilai indekx di luar batas");
    }else{
        Node current = head;
        int i = 0;
        while (i < index){
            current = current.next;
            i++;
        }
        if (current.prev == null){
            Node newNode = new Node(current.prev, item, current);
            newNode.prev = current.prev;
            newNode.next = current;
            current.prev.next = newNode;
            current.prev = newNode;
        }else{
            Node newNode = new Node(prev:null, item, current);
            newNode.prev = current.prev;
            newNode.next = current;
            current.prev.next = newNode;
            current.prev = newNode;
        }
    }
    size++;
}
```

.Jumlah data yang ada di dalam linked lists akan diperbarui secara otomatis,sehingga dapat dibuat method size() untuk mendapatkan nilai dari size.

```
69      public int size(){
70          return size;
71      }
```

Selanjutnya dibuat method clear() untuk menghapus semua isi linked lists, sehingga linked lists dalam kondisi kosong.

```
73      public void clear(){
74          head = null;
75          size = 0;
76      }
```

Untuk mencetak isi dari linked lists dibuat method print(). Method ini akan mencetak isi linked lists berapapun size-nya. Jika kosong akan dimunculkan suatu pemberitahuan bahwa linked lists dalam kondisi kosong.

```
78      public void print(){
79          if(!isEmpty()){
80              Node tmp = head;
81              while (tmp != null){
82                  System.out.println(tmp.data + "\t");
83                  tmp = tmp.next;
84              }
85              System.out.println(x:"\nberhasil diisi");
86          }else{
87              System.out.println(x:"Linked Lists Kosong");
88          }
89      }
90  }
```

Selanjutya dibuat class Main DoubleLinkedListsMain untuk mengeksekusi semua method yang ada pada class DoubleLinkedLists.

```
doublelinkedlists > DoubleLinkedListsMain.java > DoubleLinkedListsMain
1   package doublelinkedlists;
2   public class DoubleLinkedListsMain {
```

Pada main class pada langkah 16 di atas buatlah object dari class DoubleLinkedLists kemudian eksekusi

```
Run | Debug
public static void main(String[] args) throws Exception {
    DoubleLinkedLists dll = new DoubleLinkedLists();
    dll.print();
    System.out.println("Size : "+dll.size());
    System.out.println(x:"============================");
    dll.addFirst(item:3);
    dll.addLast(item:4);
    dll.addFirst(item:7);
    dll.print();
    System.out.println("Size : "+dll.size());
    System.out.println(x:"============================");
    dll.add(item:40, index:1);
    dll.print();
    System.out.println("Size : "+dll.size());
    System.out.println(x:"============================");
    dll.clear();
    dll.print();
    System.out.println("Size : "+dll.size());
}
}
```

Verifikasi Hasil Percobaan

```
Linked Lists Kosong
Size : 0
============================
7
3
4

berhasil diisi
Size : 3
============================
7
40
3
4

berhasil diisi
Size : 4
============================
Linked Lists Kosong
Size : 0
PS C:\Muhammad Afiq Firdaus\Semester 2\Algor
n 13>
```

Pertanyaan Percobaan

1. Jelaskan perbedaan antara single linked list dengan double linked lists!
2. Perhatikan class Node, di dalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?
3. Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan inisialisasi atribut head dan size seperti pada gambar berikut ini?

```
public DoubleLinkedLists() {
    head = null;
    size = 0;
}
```

4. Pada method addFirst(), kenapa dalam pembuatan object dari konstruktor class Node prev dianggap sama dengan null?

```
Node newNode = new Node(null, item, head);
```

5. Perhatikan pada method addFirst(). Apakah arti statement head.prev = newNode ?
6. Perhatikan isi method addLast(), apa arti dari pembuatan object Node dengan mengisikan parameter prev dengan current, dan next dengan null?

```
Node newNode = new Node(current, item, null);
```

7. Pada method add(), terdapat potongan kode program sebagai berikut:

```
if (current.prev == null) {
    Node newNode = new Node(null, item, current);
    current.prev = newNode;
    head = newNode;
```

jelaskan maksud dari bagian yang ditandai dengan kotak kuning.

Jawaban

1. Perbedaan utama antara single linked list dan double linked list adalah bahwa single linked list hanya memiliki referensi ke node berikutnya, sedangkan double linked list memiliki referensi ke node berikutnya dan sebelumnya. Hal ini membuat double linked list lebih fleksibel dan efisien dalam beberapa operasi, tetapi dengan biaya penggunaan memori dan kompleksitas yang lebih tinggi.
2. Atribut next dan prev dalam kelas Node berfungsi untuk menghubungkan node satu sama lain dalam dua arah dalam struktur double linked list. next menghubungkan node saat ini ke node berikutnya, sementara prev menghubungkan node saat ini ke node sebelumnya.
3. Inisialisasi atribut head ke null dan size ke **0** dalam konstruktor kelas DoubleLinkedLists memastikan bahwa objek dimulai dalam keadaan yang konsisten dan siap untuk digunakan. Ini membantu menjaga integritas struktur data dan mempermudah pelaksanaan operasi-operasi pada double linked list.
4. Mengatur prev ke null saat membuat node baru dalam metode addFirst adalah langkah penting untuk menunjukkan bahwa node baru tersebut tidak memiliki node sebelumnya, karena node baru ini akan menjadi node pertama dalam double linked

list. Ini memastikan bahwa struktur data tetap konsisten dan traversal dari head ke node lainnya berfungsi dengan benar.

5. Pernyataan head.prev = newNode menghubungkan node baru yang ditambahkan di awal daftar dengan node yang sudah ada sebelumnya. Ini memastikan bahwa node yang sebelumnya menjadi head sekarang memiliki referensi ke node baru sebagai node sebelumnya (prev). Ini merupakan langkah penting untuk memelihara integritas struktur double linked list saat menambahkan node baru di awal daftar.
6. Pernyataan Node newNode = new Node(current, item, null); dalam metode addLast() digunakan untuk membuat node baru yang akan ditambahkan di akhir double linked list. Parameter prev diisi dengan referensi ke node terakhir saat ini (current), dan next diisi dengan null untuk menunjukkan bahwa node baru adalah node terakhir dalam daftar.
7. Potongan kode dalam blok if pada metode add() digunakan untuk menyisipkan node baru ke dalam double linked list di posisi awal (indeks 0) dari daftar. Node baru dibuat dan diatur dengan benar, kemudian referensi prev dari node pertama diubah untuk menunjuk ke node baru, dan head diperbarui untuk menunjuk ke node baru (karena sekarang node baru adalah node pertama dalam daftar setelah penyisipan).

Kegiatan Praktikum 2

Buatlah method removeFirst() di dalam class DoubleLinkedLists.

```
198     public void removeFirst() throws Exception {
199         if (isEmpty()) {
200             throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus");
201         } else if (size == 1) {
202             removeLast();
203         } else {
204             head = head.next;
205             head.prev = null;
206             size--;
207         }
208     }
```

Tambahkan method removeLast() di dalam class DoubleLinkedLists.

```
182     public void removeLast() throws Exception {
183         if (isEmpty()) {
184             throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus!");
185         } else if (head.next == null) {
186             head = null;
187             size--;
188             return;
189         }
190         Node current = head;
191         while (current.next.next != null) {
192             current = current.next;
193         }
194         current.next = null;
195         size--;
196     }
```

Tambahkan pula method remove(int index) pada class DoubleLinkedLists dan amati hasilnya.

```
210     public void remove(int index) throws Exception {
211         if (isEmpty() || index >= size) {
212             throw new Exception(message:"Nilai indeks di luar batas");
213         } else if (index == 0) {
214             removeFirst();
215         } else {
216             Node current = head;
217             int i = 0;
218             while (i < index) {
219                 current = current.next;
220                 i++;
221             }
222             if (current.next == null) {
223                 current.prev.next = null;
224             } else if (current.prev == null) {
225                 current = current.next;
226                 current.prev = null;
227                 head = current;
228             } else {
229                 current.prev.next = current.next;
230                 current.next.prev = current.prev;
231             }
232             size--;
233         }
234     }
235 }
```

Untuk mengeksekusi method yang baru saja dibuat, tambahkan potongan kode program berikut pada main class

```
25  // PRAKTIKUM 2
26  package doublelinkedlists;
27  public class DoubleLinkedListsMain {
        Run | Debug
28      public static void main(String[] args) throws Exception {
29          DoubleLinkedLists dll = new DoubleLinkedLists();
30          dll.addLast(item:50);
31          dll.addLast(item:40);
32          dll.addLast(item:10);
33          dll.addLast(item:20);
34          dll.print();
35          System.out.println("Size : "+dll.size());
36          System.out.println(x:"============================");
37          dll.removeFirst();
38          dll.print();
39          System.out.println("Size : "+dll.size());
40          System.out.println(x:"============================");
41          dll.removeLast();
42          dll.print();
43          System.out.println("Size : "+dll.size());
44          System.out.println(x:"============================");
45          dll.remove(index:1);
46          dll.print();
47          System.out.println("Size : "+dll.size());
48      }
49  }
```

Verifikasi Hasil Percobaan

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 13>  c:; cd '
n 13'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp'
eb74886\redhat.java\jdt_ws\pertemuan 13_5d738fd\bin' 'doublelinkedlists.DoubleLinkedListsMain'
50
40
10
20

berhasil diisi
Size : 4
============================
40
10
20

berhasil diisi
Size : 3
============================
40
10

berhasil diisi
Size : 2
============================
40

berhasil diisi
Size : 1
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 13>
```

Pertanyaan Percobaan

1. Apakah maksud statement berikut pada method removeFirst()?
   head = head.next; head.prev = null;
2. Bagaimana cara mendeteksi posisi data ada pada bagian akhir pada method removeLast()?
3. Jelaskan alasan potongan kode program di bawah ini tidak cocok untuk perintah remove!

```
Node tmp = head.next;

head.next=tmp.next;
tmp.next.prev=head;
```

4. Jelaskan fungsi kode program berikut ini pada fungsi remove!

```
current.prev.next = current.next;
current.next.prev = current.prev;
```

Jawaban :

1. Pernyataan head = head.next; head.prev = null; dalam metode removeFirst() digunakan untuk menghapus node pertama (head) dari double linked list dengan menggeser head ke node berikutnya dalam daftar dan kemudian mengatur referensi prev dari node baru yang menjadi head menjadi null. Hal ini memastikan bahwa node baru tersebut menjadi node pertama dalam daftar dengan prev yang benar diatur ke null, dan daftar tetap konsisten setelah penghapusan node pertama.
2. memeriksa apakah daftar kosong sebelum memulai iterasi. Selanjutnya, iterasi menggunakan loop while untuk mencari node terakhir dalam daftar. Setelah menemukan node terakhir, operasi current.prev.next = null; menghapus node terakhir

dengan menghapus referensi next-nya ke node sebelumnya. Terakhir, size-- digunakan untuk mengurangi ukuran daftar setelah penghapusan.

3. Potongan kode tersebut tidak benar-benar menghapus tmp dari daftar secara utuh. Meskipun node tmp (node kedua) tidak lagi memiliki referensi ke node ketiga, node tersebut masih memiliki referensi ke head (node pertama).
4. Potongan kode program current.prev.next = current.next; dan current.next.prev = current.prev; pada fungsi remove() digunakan untuk menghapus node current dari double linked list dengan memperbarui referensi prev dan next dari node tetangga sebelum dan sesudah current.

Kegiatan Praktikum 3

Buatlah method getFirst() di dalam class DoubleLinkedLists untuk mendapatkan data pada awal linked lists.

```
381     public int getFirst() throws Exception {
382         if (isEmpty()) {
383             throw new Exception(message:"Linked List kosong");
384         }
385         return head.data;
386     }
```

Selanjutnya, buatlah method getLast() untuk mendapat data pada akhir linked lists.

```
388     public int getLast() throws Exception {
389         if (isEmpty()) {
390             throw new Exception(message:"Linked List kosong");
391         }
392         Node tmp = head;
393         while (tmp.next != null) {
394             tmp = tmp.next;
395         }
396         return tmp.data;
397     }
```

Method get(int index) dibuat untuk mendapatkan data pada indeks tertentu

```
399     public int get(int index) throws Exception {
400         if (isEmpty() || index >= size) {
401             throw new Exception(message:"Nilai indeks di luar batas.");
402         }
403         Node tmp = head;
404         for (int i = 0; i < index; i++) {
405             tmp = tmp.next;
406         }
407         return tmp.data;
408     }
```

Pada main class tambahkan potongan program berikut dan amati hasilnya!

```java
// PRAKTIKUM 3
package doublelinkedlists;

public class DoubleLinkedListsMain {
    Run | Debug
    public static void main(String[] args) throws Exception {
        DoubleLinkedLists dll = new DoubleLinkedLists();
        dll.print();
        System.out.println("Size : " + dll.size());
        System.out.println(x:"============================");
        dll.addFirst(item:3);
        dll.addLast(item:4);
        dll.addFirst(item:7);
        dll.print();
        System.out.println("Size : " + dll.size());
        System.out.println(x:"============================");
        dll.add(item:40, index:1);
        dll.print();
        System.out.println("Size : " + dll.size());
        System.out.println(x:"============================");
        System.out.println("Data awal pada Linked Lists adalah : " + dll.getFirst());
        System.out.println("Data akhir pada Linked Lists adalah : " + dll.getLast());
        System.out.println("Data indeks ke-1 pada Linked Lists adalah : " + dll.get(index:1));
    }
}
```

Verifikasi Hasil Percobaan

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 13>  c:; cd 'c:\Mu
n 13'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\
eb74886\redhat.java\jdt_ws\pertemuan 13_5d738fd\bin' 'doublelinkedlists.DoubleLinkedListsMain'
Linked Lists Kosong
Size : 0
============================
7
3
4

berhasil diisi
Size : 3
============================
7
40
3
4

berhasil diisi
Size : 4
============================
Data awal pada Linked Lists adalah : 7
Data akhir pada Linked Lists adalah : 4
Data indeks ke-1 pada Linked Lists adalah : 40
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemuan 13>
```

Pertanyaan Percobaan

1. Jelaskan method size() pada class DoubleLinkedLists!
2. Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke1!
3. Jelaskan perbedaan karakteristik fungsi Add pada Double Linked Lists dan Single Linked Lists!
4. Jelaskan perbedaan logika dari kedua kode program di bawah ini!

```
public boolean isEmpty(){
    if(size ==0){
        return true;
    } else{
        return false;
    }
}
```

(a)

```
public boolean isEmpty(){
    return head == null;
}
```

(b)

Jawaban :

1. Metode size() pada class DoubleLinkedLists bertujuan untuk mengembalikan jumlah elemen (node) yang ada dalam double linked list. Ini adalah nilai dari atribut size yang di-maintain dan diperbarui setiap kali ada penambahan atau penghapusan elemen dalam daftar.
2. **Tentukan Konvensi Indeks:**
   membuat konvensi bahwa indeks pertama (1) mengacu pada elemen pertama dalam double linked list. Ini berbeda dengan indeks standar dalam bahasa pemrograman **seperti Java yang dimulai dari 0.**
   Modifikasi Metode Penambahan dan Penghapusan:
   perlu memodifikasi metode add(int item, int index) dan remove(int index) sehingga mereka menerima indeks yang dimulai dari 1 sebagai input, bukan indeks yang dimulai dari 0.
3. **Single Linked List:**
   - Lebih sederhana dan menggunakan memori lebih sedikit.
   - Cocok untuk aplikasi di mana operasi traversal hanya satu arah diperlukan.

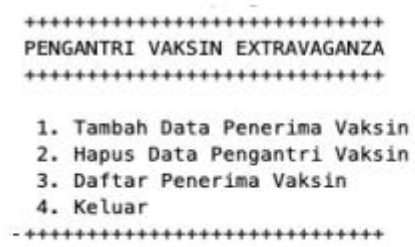
   Double Linked List:

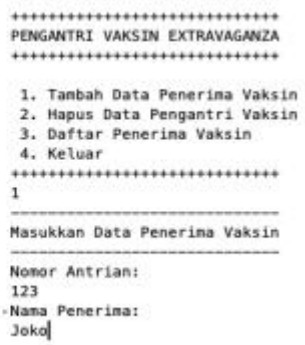
   - Lebih kompleks dan menggunakan lebih banyak memori.
   - Mendukung traversal dua arah yang lebih fleksibel dan efisien dalam beberapa operasi seperti penghapusan node tertentu.

4. Pemilihan metode tergantung pada konteks dan kebutuhan spesifik implementasi linked list Anda.
   - Jika Anda sudah memelihara ukuran list dengan variabel size dan memastikan size selalu akurat, metode (a) bisa digunakan.
   - Jika Anda tidak ingin menambah kompleksitas dengan variabel tambahan atau lebih suka logika yang lebih sederhana, metode (b) adalah pilihan yang baik.

Tugas Praktikum

Buat program antrian vaksinasi menggunakan queue berbasis double linked list sesuai ilustrasi dan menu di bawah ini! (counter jumlah antrian tersisa di menu cetak(3) dan data orang yang telah divaksinasi di menu Hapus Data(2) harus ada) Ilustrasi Program Menu Awal dan Penambahan Data





Cetak Data (Komponen di area merah harus ada)

Hapus Data (Komponen di area merah harus ada)

2. Buatlah program daftar film yang terdiri dari id, judul dan rating menggunakan double linked lists, bentuk program memiliki fitur pencarian melalui ID Film dan pengurutan Rating secara descending. Class Film wajib diimplementasikan dalam soal ini.

Contoh Ilustrasi Program

Menu Awal dan Penambahan Data

```
==================================
DATA FILM LAYAR LEBAR
==================================
 1. Tambah Data Awal
 2. Tambah Data Akhir
 3. Tambah Data Index Tertentu
 4. Hapus Data Pertama
 5. Hapus Data Terakhir
 6. Hapus Data Tertentu
 7. Cetak
 8. Cari ID Film
 9. Urut Data Rating Film-DESC
 10. Keluar
==================================
```

```
==================================
DATA FILM LAYAR LEBAR
==================================
 1. Tambah Data Awal
 2. Tambah Data Akhir
 3. Tambah Data Index Tertentu
 4. Hapus Data Pertama
 5. Hapus Data Terakhir
 6. Hapus Data Tertentu
 7. Cetak
 8. Cari ID Film
 9. Urut Data Rating Film-DESC
 10. Keluar
==================================
1
Masukkan Data Film Posisi Awal
ID Film:
1222
Judul Film:
Spider-Man: No Way Home
Rating Film:
8.7
```

```
==================================
DATA FILM LAYAR LEBAR
==================================
 1. Tambah Data Awal
 2. Tambah Data Akhir
 3. Tambah Data Index Tertentu
 4. Hapus Data Pertama
 5. Hapus Data Terakhir
 6. Hapus Data Tertentu
 7. Cetak
 8. Cari ID Film
 9. Urut Data Rating Film-DESC
 10. Keluar
==================================
2
Masukkan Data Posisi Akhir
ID Film:
1346
Judul Film:
Uncharted
Rating Film:
6.7
```

```
==================================
DATA FILM LAYAR LEBAR
==================================
 1. Tambah Data Awal
 2. Tambah Data Akhir
 3. Tambah Data Index Tertentu
 4. Hapus Data Pertama
 5. Hapus Data Terakhir
 6. Hapus Data Tertentu
 7. Cetak
 8. Cari ID Film
 9. Urut Data Rating Film-DESC
 10. Keluar
==================================
3
Masukkan Data Film
Urutan ke-
ID Film:
1234
Judul Film:
Death on the Nile
Rating Film:
6.6
Data Film ini akan masuk di urutan ke-
3
```

Cetak Data

```
==================================
DATA FILM LAYAR LEBAR
==================================
 1. Tambah Data Awal
 2. Tambah Data Akhir
 3. Tambah Data Index Tertentu
 4. Hapus Data Pertama
 5. Hapus Data Terakhir
 6. Hapus Data Tertentu
 7. Cetak
 8. Cari ID Film
 9. Urut Data Rating Film-DESC
 10. Keluar
==================================
7
Cetak Data
ID: 1222
 Judul Film: Spider-Man: No Way Home
 ipk: 8.7
ID: 1765
 Judul film: Skyfall
 ipk: 7.8
ID: 1567
 Judul Film: The Dark Knight Rises
 ipk: 8.4
ID: 1234
 Judul Film: Death on The Nile
 ipk: 6.6
ID: 1346
 Judul Film: Uncharted
 ipk: 6.7
```

Pencarian Data

```
==================================
DATA FILM LAYAR LEBAR
==================================
 1. Tambah Data Awal
 2. Tambah Data Akhir
 3. Tambah Data Index Tertentu
 4. Hapus Data Pertama
 5. Hapus Data Terakhir
 6. Hapus Data Tertentu
 7. Cetak
 8. Cari ID Film
 9. Urut Data Rating Film-DESC
 10. Keluar
==================================
8
Cari Data
Masukkan ID Film yang dicari
1567
Data Id Film: 1567 berada di node ke- 3
IDENTITAS:
 ID Film: 1567
 Judul Film: The Dark Knight Rises
 IMDB Rating: 8.4
```

Jawaban

1.
```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\p
ktikum\pertemuan 13'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+Show
rage\01ead246c1411d8a1cfcad47aeb74886\redhat.java\jdt_ws\pertemuan 13_5d738fd\
========================================
     PENGANTRI VAKSIN EXTRAVAGANZA
========================================
1. Tambah antrian
2. Hapus antrian
3. Daftar antrian
4. Keluar
========================================
Masukkan pilihan  :
```

```
========================================
           DAFTAR ANTRIAN
========================================
|No Antrian     |Nama           |
----------------------------------------
|3              |kas            |
|2              |afa            |
|1              |afq            |

========================================
```

```
========================================
Masukkan pilihan  : 2
Antrian nomor 1 dengan nama afq telah divaksinasi
========================================
     PENGANTRI VAKSIN EXTRAVAGANZA
========================================
1. Tambah antrian
2. Hapus antrian
3. Daftar antrian
4. Keluar
========================================
Masukkan pilihan  : 2
Antrian nomor 2 dengan nama afa telah divaksinasi
========================================
     PENGANTRI VAKSIN EXTRAVAGANZA
========================================
1. Tambah antrian
2. Hapus antrian
3. Daftar antrian
4. Keluar
========================================
Masukkan pilihan  :
```

```
PS C:\Muhammad Afiq Firdaus\Semester 2\Algoritma dan Struktur Data\Praktikum\pertemu
ktikum\pertemuan 13'; & 'C:\Program Files\Java\jdk-21\bin\java.exe' '-XX:+ShowCodeDe
rage\01ead246c1411d8a1cfcad47aeb74886\redhat.java\jdt_ws\pertemuan 13_5d738fd\bin' '
================================================================
                    DATA FILM LAYAR LEBAR
================================================================
1. Tambah Film Awal
2. Tambah Film Akhir
3. Tambah Film Index Tertentu
4. Hapus Film Pertama
5. Hapus Film Terakhir
6. Hapus Film Index Tertentu
7. Tampilkan Daftar Film
8. Cari Film Berdasarkan Id Film
9. Urutkan Daftar Film Berdasarkan Rating (Desc)
10. Cari Film Berdasarkan Judul
11. Hapus Film Berdasarkan Judul
12. Keluar
================================================================
Masukkan Pilihan    :
```

2.

```
12. Keluar
================================================================
Masukkan Pilihan    : 9
================================================================
          DAFTAR FILM DESCENDING BERDASARKAN RATING
================================================================
| No Film | Judul Film                          | Rating |
----------------------------------------------------------------
| 1132    | merah                               | 5,00   |
| 1121    | biru                                | 5,00   |
| 1211    | hijau                               | 4,00   |
| 1213    | kuning                              | 4,00   |
================================================================
================================================================
                    DATA FILM LAYAR LEBAR
================================================================
```

```
================================================================
Masukkan Pilihan    : 7
================================================================
                        DAFTAR FILM
================================================================
| No Film | Judul Film                          | Rating |
----------------------------------------------------------------
| 1211    | hijau                               | 4,00   |
| 1132    | merah                               | 5,00   |
| 1121    | biru                                | 5,00   |
| 1213    | kuning                              | 4,00   |
================================================================
```