

# LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA



2022



## Daftar Isi

<b>PERCOBAAN PRAKTIKUM 1</b>	<b>4</b>
LANGKAH 1	4
LANGKAH 2	4
LANGKAH 3	5
LANGKAH 4	5
LANGKAH 5	5
LANGKAH 6	5
LANGKAH 7	6
LANGKAH 8	6
LANGKAH 9	6
LANGKAH 10	6
LANGKAH 11	7
LANGKAH 12	7
LANGKAH 13	7
LANGKAH 14	8
LANGKAH 15	8
LANGKAH 16	8
LANGKAH 17	9
VERIFIKASI HASIL PERCOBAAN	9
<i>Pertanyaan Percobaan</i>	9
<i>Jawaban Percobaan</i>	10
<b>PERCOBAAN PRAKTIKUM 2</b>	<b>11</b>
LANGKAH 1	11
LANGKAH 2	11
LANGKAH 3	12
LANGKAH 4	12
VERIFIKASI HASIL PERCOBAAN	13
<i>Pertanyaan Percobaan</i>	13
<i>Jawaban Percobaan</i>	14
<b>PERCOBAAN PRAKTIKUM 3</b>	<b>15</b>
LANGKAH 1	15
LANGKAH 2	15
LANGKAH 3	16



## PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

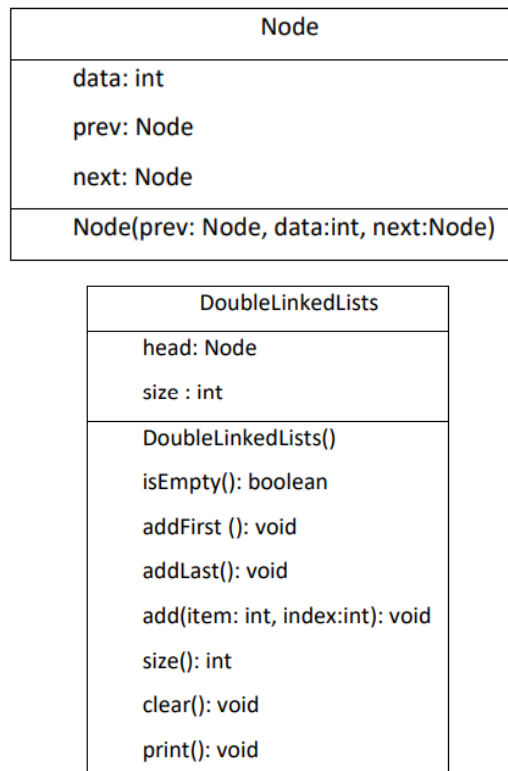
LANGKAH 4.....	16
VERIFIKASI HASIL PERCOBAAN .....	16
<i>Pertanyaan Percobaan</i> .....	17
<i>Jawaban Percobaan</i> .....	17
<b>TUGAS PRAKTIKUM .....</b>	<b>18</b>
SOAL NOMOR 1.....	18
JAWABAN NOMOR 1.....	19
<i>Source code Class Soal1DataAntrian</i> .....	19
<i>Source code Class Soal1DLLMainRidwan</i> .....	19
<i>Source code Class Soal1DLLRidwan</i> .....	20
<i>Source code Class Soal1NodeRidwan</i> .....	21
<i>Output Menu Awal</i> .....	21
<i>Output Penambahan Data</i> .....	22
<i>Output Cetak Data</i> .....	22
<i>Output Hapus Data</i> .....	23
<i>Output Keluar</i> .....	23
SOAL NOMOR 2 .....	24
JAWABAN NOMOR 2.....	25
<i>Source code Class Soal2DataFilm</i> .....	25
<i>Source code Class Soal2DLLMainRidwan</i> .....	25
<i>Source code Class Soal2DLLRidwan</i> .....	27
<i>Source code Class Soal2NodeRidwan</i> .....	30
<i>Output Menu Awal</i> .....	31
<i>Output Tambah Data Awal</i> .....	31
<i>Output Tambah Data Akhir</i> .....	32
<i>Output Tambah Data Index Tertentu</i> .....	33
<i>Output Hapus Data Pertama</i> .....	33
<i>Output Hapus Data Terakhir</i> .....	34
<i>Output Hapus Data Tertentu</i> .....	34
<i>Output Cetak</i> .....	34
<i>Output Cari ID Film</i> .....	36
<i>Output Urut Data Rating-DESC</i> .....	37
<i>Output Keluar</i> .....	37

## Percobaan Praktikum 1

Pada percobaan 1 ini akan dibuat class Node dan class DoubleLinkedLists yang didalamnya terdapat operasi-operasi untuk menambahkan data dengan beberapa cara (dari bagian depan linked list, belakang ataupun indeks tertentu pada linked list).

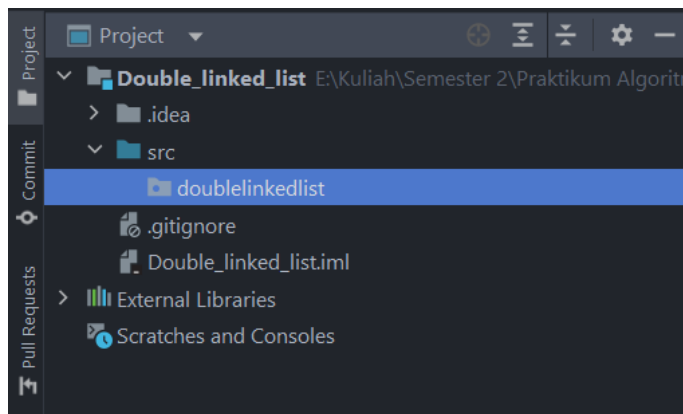
### Langkah 1

Perhatikan diagram class Node dan class DoubleLinkedLists di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program DoubleLinkedLists.



### Langkah 2

Buat paket baru dengan nama doublelinkedlists



## Langkah 3

Buat class di dalam paket tersebut dengan nama Node

```
NodeRidwan.java x
1 package doublelinkedlist;
2
3 public class NodeRidwan {
4
5 }
6
```

## Langkah 4

Di dalam class tersebut, deklarasikan atribut sesuai dengan diagram class di atas.

```
NodeRidwan.java x
1 package doublelinkedlist;
2
3 public class NodeRidwan {
4     int data;
5     NodeRidwan prev, next;
6 }
```

## Langkah 5

Selanjutnya tambahkan konstruktor default pada class Node sesuai diagram di atas

```
7 NodeRidwan(NodeRidwan prev, int data, NodeRidwan next) {
8     this.prev = prev;
9     this.data = data;
10    this.next = next;
11 }
```

## Langkah 6

Buatlah sebuah class baru bernama DoubleLinkedLists pada package yang sama dengan node seperti gambar berikut:

```

NodeRidwan.java x DoubleLinkedListsRidwan.java x
1 package doublelinkedlists;
2
3 public class DoubleLinkedListsRidwan {
4
5 }

```

### Langkah 7

Pada class DoubleLinkedLists tersebut, deklarasikan atribut sesuai dengan diagram class di atas.

```

3 public class DoubleLinkedListsRidwan {
4     NodeRidwan head;
5     int size;
6 }

```

### Langkah 8

Selanjutnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut

```

7 public DoubleLinkedListsRidwan() {
8     head = null;
9     size = 0;
10 }

```

### Langkah 9

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

```

12 public boolean isEmpty() {
13     return head == null;
14 }

```

### Langkah 10

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

```

16 public void addFirst(int item) {
17     if (isEmpty()) {
18         head = new NodeRidwan( prev: null, item, next: null);
19     } else {
20         NodeRidwan newNode = new NodeRidwan( prev: null, item, head);
21         head.prev = newNode;
22         head = newNode;
23     }
24     size++;
25 }

```

### Langkah 11

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

```
27 public void addLast(int item) {
28     if (isEmpty()) {
29         addFirst(item);
30     } else {
31         NodeRidwan current = head;
32         while (current.next != null) {
33             current = current.next;
34         }
35         NodeRidwan newNode = new NodeRidwan(current, item, next: null);
36         current.next = newNode;
37         size++;
38     }
39 }
```

### Langkah 12

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

```
41 public void add(int item, int index) throws Exception {
42     if (isEmpty()) {
43         addFirst(item);
44     } else if (index < 0 || index > size) {
45         throw new Exception("Nilai indeks di luar batas");
46     } else {
47         NodeRidwan current = head;
48         int i = 0;
49         while (i < index) {
50             current = current.next;
51             i++;
52         }
53         if (current.prev == null) {
54             NodeRidwan newNode = new NodeRidwan(prev: null, item, current);
55             current.prev = newNode;
56             head = newNode;
57         } else {
58             NodeRidwan newNode = new NodeRidwan(current.prev, item, current);
59             newNode.prev = current.prev;
60             newNode.next = current;
61             current.prev.next = newNode;
62             current.prev = newNode;
63         }
64     }
65     size++;
66 }
```

### Langkah 13

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

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

#### Langkah 14

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

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

#### Langkah 15

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

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

#### Langkah 16

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

```
1     package doublelinkedlists;  
2  
3     public class DoubleLinkedListsMainRidwan {  
4         public static void main(String[] args) {  
5  
6         }  
7     }
```



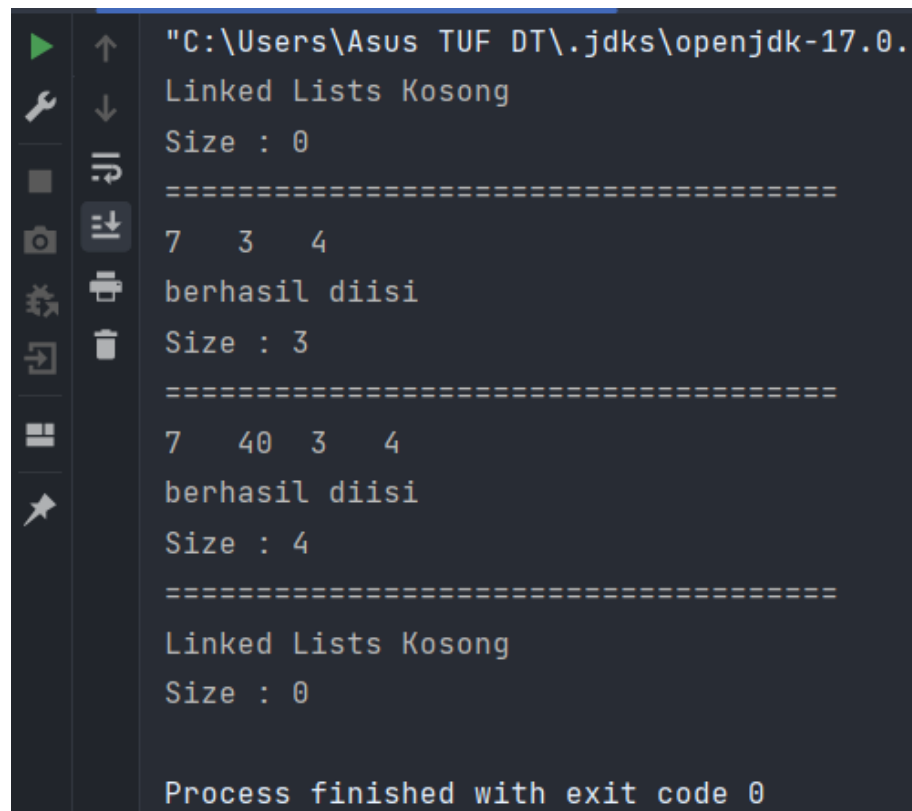
## Langkah 17

Pada main class pada langkah 16 di atas buatlah object dari class DoubleLinkedLists kemudian eksekusi potongan program berikut ini.

```
5      DoubleLinkedListsRidwan dll = new DoubleLinkedListsRidwan();
6      dll.print();
7      System.out.println("Size : " + dll.size);
8      System.out.println("=====");
9      dll.addFirst( item: 3);
10     dll.addLast( item: 4);
11     dll.addFirst( item: 7);
12     dll.print();
13     System.out.println("Size : " + dll.size);
14     System.out.println("=====");
15     dll.add( item: 40, index: 1);
16     dll.print();
17     System.out.println("Size : " + dll.size);
18     System.out.println("=====");
19     dll.clear();
20     dll.print();
21     System.out.println("Size : " + dll.size);
```

## Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.



```
"C:\Users\Asus TUF DT\.jdk\openjdk-17.0.2
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

Process finished with exit code 0
```

## Pertanyaan Percobaan

1. Jelaskan perbedaan antara single linked list dengan double linked lists!

2. Perhatikan class Node, didalamnya 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 mengisi parameter prev dengan current, dan next dengan null?

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

#### Jawaban Percobaan

1. Single linked list memiliki satu variabel yang menunjuk ke node selanjutnya yang biasa disebut dengan next, Sedangkan untuk double linked list memiliki dua variabel yang menunjuk ke node sebelum dan selanjutnya yang biasa disebut prev dan next.
2. Atribut next digunakan sebagai pointer yang menunjuk ke node setelahnya sedangkan atribut prev digunakan sebagai pointer yang menunjuk node sebelumnya.
3. Inisialisasi head bernilai null dan size = 0 karena ketika double linked list pertama kali dibuat akan kosong maka head menunjuk ke null dan size 0.
4. Karena pada addfirst data baru akan ditempatkan urutan paling depan sehingga prev nya menunjuk ke null.
5. Statement tersebut digunakan untuk mengisi pointer pada head saat ini dengan node data baru kemudian node baru akan menjadi head yang baru.
6. Statement tersebut digunakan setelah melakukan proses transvers hingga ditemukan data paling belakang kemudian node baru akan memiliki pointer prev yang menunjuk ke current dan next menunjuk ke null karena data baru ditempatkan paling belakang.

## Percobaan Praktikum 2

Pada praktikum 2 ini akan dibuat beberapa method untuk menghapus isi LinkedLists pada class DoubleLinkedLists. Penghapusan dilakukan dalam tiga cara di bagian paling depan, paling belakang, dan sesuai indeks yang ditentukan pada linkedLists. Method tambahan tersebut akan ditambahkan sesuai pada diagram class berikut ini.

DoubleLinkedLists
head: Node size : int
DoubleLinkedLists() isEmpty(): boolean addFirst (): void addLast(): void add(item: int, index:int): void size(): int clear(): void print(): void <b>removeFirst(): void</b> <b>removeLast(): void</b> <b>remove(index:int):void</b>

### Langkah 1

Buatlah method removeFirst() di dalam class DoubleLinkedLists

```
90 public void removeFirst() throws Exception {  
91     if (isEmpty()) {  
92         throw new Exception("Linked List masih kosong, tidak dapat dihapus");  
93     } else if (size == 1) {  
94         removeLast();  
95     } else {  
96         head = head.next;  
97         head.prev = null;  
98         size--;  
99     }  
100 }
```

### Langkah 2

Tambahkan method removeLast() di dalam class DoubleLinkedLists.

```
102 public void removeLast() throws Exception {
103     if (isEmpty()) {
104         throw new Exception("Linked list masih kosong, tidak dapat dihapus");
105     } else if (head.next == null) {
106         head = null;
107         size--;
108         return;
109     }
110     NodeRidwan current = head;
111     while (current.next.next != null) {
112         current = current.next;
113     }
114     current.next = null;
115     size--;
116 }
```

### Langkah 3

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

```
118 public void remove(int index) throws Exception {
119     if (isEmpty() || index >= size) {
120         throw new Exception("Nilai indeks diluar batas");
121     } else if (index == 0) {
122         removeFirst();
123     } else {
124         NodeRidwan current = head;
125         int i = 0;
126         while (i < index) {
127             current = current.next;
128             i++;
129         }
130         if (current.next == null) {
131             current.prev.next = null;
132         } else if (current.prev == null) {
133             current = current.next;
134             current.prev = null;
135             head = current;
136         } else {
137             current.prev.next = current.next;
138             current.next.prev = current.prev;
139         }
140         size--;
141     }
142 }
```

### Langkah 4

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

```
22     dll.addLast( item: 50);
23     dll.addLast( item: 40);
24     dll.addLast( item: 10);
25     dll.addLast( item: 20);
26     dll.print();
27     System.out.println("Size : " + dll.size);
28     System.out.println("=====");
29     dll.removeFirst();
30     dll.print();
31     System.out.println("Size : " + dll.size);
32     System.out.println("=====");
33     dll.removeLast();
34     dll.print();
35     System.out.println("Size : " + dll.size);
36     System.out.println("=====");
37     dll.remove( index: 1);
38     dll.print();
39     System.out.println("Size : " + dll.size);
```

### Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini

```
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

Process finished with exit code 0
```

### 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;

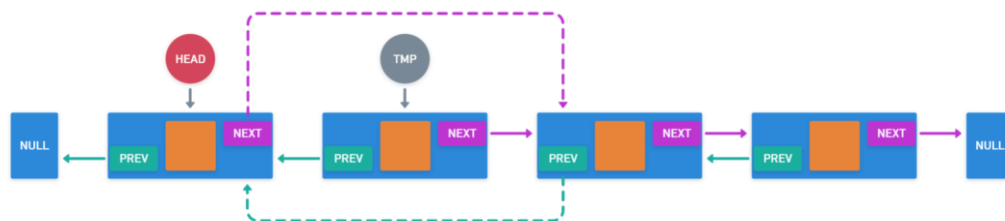
head.next=tmp.next;
tmp.next.prev=head;
```

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

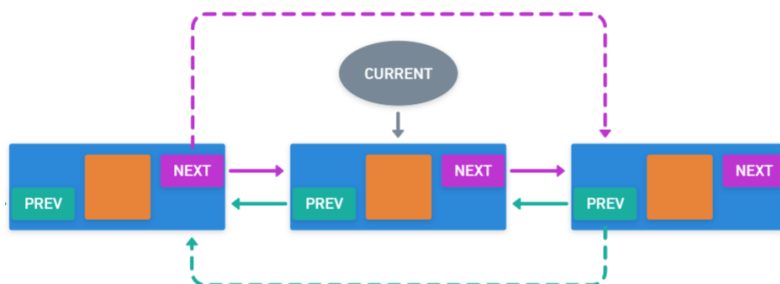
```
current.prev.next = current.next;
current.next.prev = current.prev;
```

### Jawaban Percobaan

1. Statement tersebut digunakan ketika data paling depan akan dihapus maka head diganti ke node yang ditunjuk oleh pointer next head kemudian pointer prev pada head yang baru akan menunjuk ke null karena data sebelumnya sudah tidak digunakan.
2. Untuk mendeteksi posisi data pada bagian akhir digunakan cara transvers variabel current hingga nilai current.next bernilai null.
3. Kode tersebut hanya akan menghapus node setelah head sedangkan untuk method remove kita harus bisa menghapus sesuai dengan index



4. Fungsi dari kode program tersebut adalah pointer next yang dimiliki oleh node sebelum current akan menunjuk ke node setelah current dan pointer prev yang dimiliki oleh node setelah current akan menunjuk ke node sebelum current.



## Percobaan Praktikum 3

Pada praktikum 3 ini dilakukan uji coba untuk mengambil data pada linked list dalam 3 kondisi, yaitu mengambil data paling awal, paling akhir dan data pada indeks tertentu dalam linked list. Method mengambil data dinamakan dengan get. Ada 3 method get yang dibuat pada praktikum ini sesuai dengan diagram class DoubleLinkedLists

DoubleLinkedLists
head: Node size : int
DoubleLinkedLists() isEmpty(): boolean addFirst (): void addLast(): void add(item: int, index:int): void size(): int clear(): void print(): void removeFirst(): void removeLast(): void remove(index:int):void <b>getFirst(): int</b> <b>getLast() : int</b> <b>get(index:int): int</b>

### Langkah 1

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

```

144     public int getFirst() throws Exception {
145         if (isEmpty()) {
146             throw new Exception("Linked list kosong");
147         }
148         return head.data;
149     }
  
```

### Langkah 2

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

```
151     public int getLast() throws Exception {
152         if (isEmpty()) {
153             throw new Exception("Linked list kosong");
154         }
155         NodeRidwan tmp = head;
156         while (tmp.next != null) {
157             tmp = tmp.next;
158         }
159         return tmp.data;
160     }
```

### Langkah 3

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

```
162     public int get(int index) throws Exception {
163         if (isEmpty() || index >= size) {
164             throw new Exception("Nilai indeks di luar batas");
165         }
166         NodeRidwan tmp = head;
167         for (int i = 0; i < index; i++) {
168             tmp = tmp.next;
169         }
170         return tmp.data;
171     }
```

### Langkah 4

Pada main class tambahkan potongan program berikut dan amati hasilnya!

```
41     dll.print();
42     System.out.println("Size : " + dll.size);
43     System.out.println("=====");
44     dll.addFirst( item: 3);
45     dll.addLast( item: 4);
46     dll.addFirst( item: 7);
47     dll.print();
48     System.out.println("Size : " + dll.size);
49     System.out.println("=====");
50     dll.add( item: 40, index: 1);
51     dll.print();
52     System.out.println("Size : " + dll.size);
53     System.out.println("=====");
54     System.out.println("Data awal pada Linked Lists adalah: " + dll.getFirst());
55     System.out.println("Data akhir pada Linked Lists adalah: " + dll.getLast());
56     System.out.println("Data indeks ke-1 pada Linked Lists adalah: " + dll.get(1));
```

### Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini



```
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

Process finished with exit code 0
```

### 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 Percobaan

1. Method size digunakan untuk mengetahui panjang dari linked list yang akan berkurang ketika di remove dan bertambah ketika ada perintah add.
2. Untuk melakukan inisialisasi indeks pada linked lists digunakan metode transvers yang akan memberi indeks dari head hingga paling belakang linked list dan untuk nilai variabel i diset ke 1 atau inputan indeks +1.
3. Jika pada single linked list ketika menambahkan data maka hanya perlu mengatur pointer next sedangkan pada double linked list harus mengatur pointer next dan prev.
4. Kode program A untuk mengecek apakah linked list kosong menggunakan size sebagai acuan sedangkan kode program B untuk mengecek apakah linked list kosong menggunakan head sebagai acuan.

## Tugas Praktikum

### Soal Nomor 1

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

```

*****
PENGANTRI VAKSIN EXTRAVAGANZA
*****

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar

*****
1
Masukkan Data Penerima Vaksin
Nomor Antrian:
123
Nama Penerima:
Joko
  
```

Cetak Data (**Komponen di area merah harus ada**)

```

*****
PENGANTRI VAKSIN EXTRAVAGANZA
*****

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar

*****
3
Daftar Pengantri Vaksin
*****
|No.   |Nama |
|123   |Joko |
|124   |Mely |
|135   |Johan|
|146   |Rosi |
Sisa Antrian: 4
  
```

Hapus Data (**Komponen di area merah harus ada**)

```

*****
PENGANTRI VAKSIN EXTRAVAGANZA
*****

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar

*****
2
Joko telah selesai divaksinasi.
Daftar Pengantri Vaksin
*****
|No.   |Nama |
|124   |Mely |
|135   |Johan|
|146   |Rosi |
Sisa Antrian: 3
  
```

## Jawaban Nomor 1

## Source code Class Soal1DataAntrian

```
1 package doublelinkedlists;
2
3 public class Soal1DataAntrian {
4     String nama;
5     int nomorAntrian;
6
7     Soal1DataAntrian(String nama, int nomorAntrian) {
8         this.nama = nama;
9         this.nomorAntrian = nomorAntrian;
10    }
11 }
```

## Source code Class Soal1DLLMainRidwan

```
1 package doublelinkedlists;
2 import java.util.Scanner;
3
4 public class Soal1DLLMainRidwan {
5     public static Scanner input = new Scanner(System.in);
6     public static Soal1DLLRidwan DLL = new Soal1DLLRidwan();
7     public static void main(String[] args) throws Exception {
8         int menuInput;
9         do {
10             Menu();
11             menuInput = input.nextInt();
12             switch (menuInput) {
13                 case 1 -> TambahData();
14                 case 2 -> HapusData();
15                 case 3 -> TampilData();
16                 case 4 -> {
17                     System.out.println("System akan dimatikan");
18                     System.exit(status: 0);
19                 }
20             }
21         } while (menuInput > 0 && menuInput < 5);
22     }
```

```

24     public static void Menu() {
25         System.out.println("+++++++");
26         System.out.println("PENGANTRI VAKSIN EXTRAVAGANZA");
27         System.out.println("+++++++");
28         System.out.println();
29         System.out.println("1. Tambah Data Penerima Vaksin");
30         System.out.println("2. Hapus Data Pengantri Vaksin");
31         System.out.println("3. Daftar Penerima Vaksin");
32         System.out.println("4. Keluar");
33         System.out.println("+++++++");
34     }
35
36     public static void TambahData() {
37         int nomorAntrian;
38         String nama;
39         System.out.println("-----");
40         System.out.println("Masukkan Data Penerima Vaksin");
41         System.out.println("-----");
42         System.out.println("Nomor Antrian : ");
43         nomorAntrian = input.nextInt();
44         input.nextLine();
45         System.out.println("Nama Penerima : ");
46         nama = input.nextLine();
47         Soal1DataAntrian newData = new Soal1DataAntrian(nama, nomorAntrian);
48         DLL.addLast(newData);
49     }
50
51     public static void TampilData() {
52         System.out.println("+++++++");
53         System.out.println("Daftar Pengantri Vaksin");
54         System.out.println("+++++++");
55         System.out.println("| No.\t| Nama\t|");
56         DLL.print();
57     }
58
59     public static void HapusData() throws Exception {
60         DLL.removeFirst();
61         TampilData();
62     }
63 }

```

## Source code Class Soal1DLLRidwan

```

1  package doublelinkedlists;
2
3  public class Soal1DLLRidwan {
4      Soal1NodeRidwan head, tail;
5      int size;
6
7      Soal1DLLRidwan() {
8          head = tail = null;
9          size = 0;
10     }
11
12     public void addLast(Soal1DataAntrian data) {
13         if (head == null) {
14             head = tail = new Soal1NodeRidwan(prev: null, data, next: null);
15         } else {
16             Soal1NodeRidwan newNode = new Soal1NodeRidwan(tail, data, next: null);
17             tail.next = newNode;
18             tail = newNode;
19         }
20         size++;
21     }

```

```

23 public void print() {
24     if (head == null) {
25         System.out.println("Data antrian masih kosong");
26     } else {
27         Soal1NodeRidwan tmp = head;
28         while (tmp != null) {
29             System.out.println("| " + tmp.data.nomorAntrian + "\t| " + tmp.data.nama + "\t|");
30             tmp = tmp.next;
31         }
32         System.out.println("Sisa Antrian: " + size);
33     }
34 }
35
36 public void removeFirst() throws Exception {
37     if (head == null) {
38         throw new Exception("Antrian sudah kosong, tidak dapat dihapus");
39     } else if (size == 1) {
40         System.out.println(head.data.nama + " telah selesai divaksinasi.");
41         head = tail = null;
42         size--;
43     } else {
44         System.out.println(head.data.nama + " telah selesai divaksinasi.");
45         head = head.next;
46         head.prev = null;
47         size--;
48     }
49 }
50 }

```

## Source code Class Soal1NodeRidwan

```

1 package doublelinkedlists;
2
3 public class Soal1NodeRidwan {
4     Soal1DataAntrian data;
5     Soal1NodeRidwan next, prev;
6
7     Soal1NodeRidwan(Soal1NodeRidwan prev, Soal1DataAntrian data, Soal1NodeRidwan next) {
8         this.prev = prev;
9         this.data = data;
10        this.next = next;
11    }
12 }

```

## Output Menu Awal

```

+++++
PENGANTRI VAKSIN EXTRAVAGANZA
+++++

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+++++

```

## Output Penambahan Data

```
+++++
PENGANTRI VAKSIN EXTRAVAGANZA
+++++

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+++++
1
-----
Masukkan Data Penerima Vaksin
-----
Nomor Antrian :
123
Nama Penerima :
Joko
```

## Output Cetak Data

```
+++++
PENGANTRI VAKSIN EXTRAVAGANZA
+++++

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+++++
3
+++++
Daftar Pengantri Vaksin
+++++
| No.  | Nama  |
| 123  | Joko  |
| 124  | Mely  |
| 135  | Johan |
| 146  | Rosi  |
Sisa Antrian: 4
```

## Output Hapus Data

```
+++++
PENGANTRI VAKSIN EXTRAVAGANZA
+++++

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+++++
2
Joko telah selesai divaksinasi.
+++++
Daftar Pengantri Vaksin
+++++
| No.   | Nama |
| 124   | Mely |
| 135   | Johan|
| 146   | Rosi |
Sisa Antrian: 3
```

## Output Keluar

```
+++++
PENGANTRI VAKSIN EXTRAVAGANZA
+++++

1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+++++
4
System akan dimatikan

Process finished with exit code 0
```

## Soal Nomor 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

=====
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

=====
Masukkan Data Film
Urutan ke-
ID Film:
1234
Judul Film:
Death on the Nile
Rating Film:
6.4
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 Nomor 2

## Source code Class Soal2DataFilm

```
1 package doublelinkedlists;
2
3 public class Soal2DataFilm {
4     int id;
5     String judul;
6     double rating;
7
8     Soal2DataFilm(int id, String judul, double rating) {
9         this.id = id;
10        this.judul = judul;
11        this.rating = rating;
12    }
13 }
```

## Source code Class Soal2DLLMainRidwan

```
1 package doublelinkedlists;
2
3 import java.util.Scanner;
4
5 public class Soal2DLLMainRidwan {
6     public static Scanner input = new Scanner(System.in);
7
8     public static void main(String[] args) throws Exception {
9         int menuInput, urutan;
10        int idFilm;
11        String judulFilm;
12        double rating;
13        Soal2DLLRidwan DLL = new Soal2DLLRidwan();
14        do {
15            menu();
16            menuInput = input.nextInt();
17            switch (menuInput) {
18                case 1 -> {
19                    System.out.println("Masukkan Data Film Posisi Awal");
20                    System.out.println("ID Film :");
21                    idFilm = input.nextInt();
22                    input.nextLine();
23                    System.out.println("Judul Film : ");
24                    judulFilm = input.nextLine();
25                    System.out.println("Rating Film : ");
26                    rating = input.nextDouble();
27                    Soal2DataFilm newData = new Soal2DataFilm(idFilm, judulFilm, rating);
28                    DLL.addFirst(newData);
29                }
30            }
31        } while (menuInput != 0);
32    }
33 }
```

```

30     case 2 -> {
31         System.out.println("Masukkan Data Posisi Akhir");
32         System.out.println("ID Film :");
33         idFilm = input.nextInt();
34         input.nextLine();
35         System.out.println("Judul Film : ");
36         judulFilm = input.nextLine();
37         System.out.println("Rating Film : ");
38         rating = input.nextDouble();
39         Soal2DataFilm newData = new Soal2DataFilm(idFilm, judulFilm, rating);
40         DLL.addLast(newData);
41     }
42     case 3 -> {
43         System.out.println("Masukkan Data Film");
44         System.out.println("Urutan ke-");
45         System.out.println("ID Film :");
46         idFilm = input.nextInt();
47         input.nextLine();
48         System.out.println("Judul Film : ");
49         judulFilm = input.nextLine();
50         System.out.println("Rating Film : ");
51         rating = input.nextDouble();
52         System.out.println("Data Film ini akan masuk di urutan ke-");
53         urutan = input.nextInt();
54         Soal2DataFilm newData = new Soal2DataFilm(idFilm, judulFilm, rating);
55         DLL.add(newData, index: urutan - 1);
56     }
57     case 4 -> {
58         System.out.println("Hapus Data Pertama");
59         DLL.removeFirst();
60     }
61     case 5 -> {
62         System.out.println("Hapus Data Terakhir");
63         DLL.removeLast();
64     }
65     case 6 -> {
66         System.out.println("Masukkan index data yang ingin dihapus : ");
67         int index = input.nextInt();
68         DLL.remove(index: index - 1);
69     }
70     case 7 -> {
71         System.out.println("Cetak Data");
72         DLL.print();
73     }
74     case 8 -> {
75         System.out.println("Cari Data");
76         System.out.println("Masukkan ID Film yang dicari ");
77         int ID = input.nextInt();
78         DLL.searchID(ID);
79     }
80     case 9 -> {
81         DLL.sortDescRating();
82     }
83     case 10 -> {
84         System.out.println("Program akan ditutup");
85         System.exit(status: 0);
86     }
87 }
88 } while (menuInput > 0 && menuInput < 11);
89 }

```

```

91 public static void menu() {
92     System.out.println("=====");
93     System.out.println("DATA FILM LAYAR LEBAR");
94     System.out.println("=====");
95     System.out.println("1. Tambah Data Awal");
96     System.out.println("2. Tambah Data Akhir");
97     System.out.println("3. Tambah Data Index Tertentu");
98     System.out.println("4. Hapus Data Pertama");
99     System.out.println("5. Hapus Data Terakhir");
100    System.out.println("6. Hapus Data Tertentu");
101    System.out.println("7. Cetak");
102    System.out.println("8. Cari ID Film");
103    System.out.println("9. Urut Data Rating Film-DESC");
104    System.out.println("10. Keluar");
105    System.out.println("=====");
106 }
107 }

```

Source code Class Soal2DLLRidwan

```

1 package doublelinkedlists;
2
3 public class Soal2DLLRidwan {
4     Soal2NodeRidwan head;
5     int size;
6
7     Soal2DLLRidwan() {
8         head = null;
9         size = 0;
10    }
11
12    public boolean isEmpty() {
13        return head == null;
14    }
15
16    public void addFirst(Soal2DataFilm data) {
17        if (isEmpty()) {
18            head = new Soal2NodeRidwan( prev: null, data, next: null);
19        } else {
20            Soal2NodeRidwan newNode = new Soal2NodeRidwan( prev: null, data, head);
21            head.prev = newNode;
22            head = newNode;
23        }
24        size++;
25    }
26
27    public void addLast(Soal2DataFilm data) {
28        if (isEmpty()) {
29            addFirst(data);
30        } else {
31            Soal2NodeRidwan current = head;
32            while (current.next != null) {
33                current = current.next;
34            }
35            Soal2NodeRidwan newNode = new Soal2NodeRidwan(current, data, next: null);
36            current.next = newNode;
37            size++;
38        }
39    }

```

```
41 public void add(Soal2DataFilm data, int index) throws Exception {
42     if (isEmpty()) {
43         addFirst(data);
44     } else if (index < 0 || index > size) {
45         throw new Exception("Nilai indeks di luar batas");
46     } else {
47         Soal2NodeRidwan current = head;
48         int i = 0;
49         while (i < index) {
50             current = current.next;
51             i++;
52         }
53         if (current.prev == null) {
54             Soal2NodeRidwan newNode = new Soal2NodeRidwan( prev: null, data, current);
55             current.prev = newNode;
56             head = newNode;
57         } else {
58             Soal2NodeRidwan newNode = new Soal2NodeRidwan(current.prev, data, current);
59             newNode.prev = current.prev;
60             newNode.next = current;
61             current.prev.next = newNode;
62             current.prev = newNode;
63         }
64     }
65     size++;
66 }
```

```
68 public void removeFirst() throws Exception {
69     if (isEmpty()) {
70         throw new Exception("Data masih kosong, tidak dapat dihapus");
71     } else if (size == 1) {
72         removeLast();
73     } else {
74         dataFilmdihapus(head);
75         head = head.next;
76         head.prev = null;
77         size--;
78     }
79 }
80
81 public void removeLast() throws Exception {
82     if (isEmpty()) {
83         throw new Exception("Data masih kosong, tidak dapat dihapus");
84     } else if (head.next == null) {
85         dataFilmdihapus(head);
86         head = null;
87         size--;
88         return;
89     }
90     Soal2NodeRidwan current = head;
91     while (current.next.next != null) {
92         current = current.next;
93     }
94     dataFilmdihapus(current.next);
95     current.next = null;
96     size--;
97 }
```

```

99      public void remove(int index) throws Exception {
100          if (isEmpty() || index >= size) {
101              throw new Exception("Nilai indeks diluar batas");
102          } else if (index == 0) {
103              removeFirst();
104          } else {
105              Soal2NodeRidwan current = head;
106              int i = 0;
107              while (i < index) {
108                  current = current.next;
109                  i++;
110              }
111              if (current.next == null) {
112                  dataFilmdiHapus(current);
113                  current.prev.next = null;
114              } else if (current.prev == null) {
115                  dataFilmdiHapus(current);
116                  current = current.next;
117                  current.prev = null;
118                  head = current;
119              } else {
120                  dataFilmdiHapus(current);
121                  current.prev.next = current.next;
122                  current.next.prev = current.prev;
123              }
124              size--;
125          }
126      }

```

```

128      @ public void dataFilmdiHapus(Soal2NodeRidwan data) {
129          System.out.println("Data Film yang dihapus : ");
130          System.out.println("ID : " + data.data.id);
131          System.out.println("\tJudul Film : " + data.data.judul);
132          System.out.println("\tipk : " + data.data.rating);
133      }
134
135      public void print() {
136          if (!isEmpty()) {
137              Soal2NodeRidwan tmp = head;
138              while (tmp != null) {
139                  System.out.println("ID: " + tmp.data.id);
140                  System.out.println("\tJudul Film: " + tmp.data.judul);
141                  System.out.println("\tipk: " + tmp.data.rating);
142                  tmp = tmp.next;
143              }
144          } else {
145              System.out.println("Linked Lists Kosong");
146          }
147      }

```

```

149 public void searchID(int id) {
150     Soal2NodeRidwan current = head;
151     int i = 1;
152     while (current != null) {
153         if (current.data.id == id) {
154             System.out.println("Data ID Film " + id + " berada di node ke- " + i);
155             System.out.println("IDENTITAS");
156             System.out.println("ID : " + current.data.id);
157             System.out.println("\tJudul Film : " + current.data.judul);
158             System.out.println("\tIMDB Rating : " + current.data.rating);
159             break;
160         }
161         current = current.next;
162         i++;
163     }
164     if (current == null) {
165         System.out.println("Data tidak ditemukan");
166     }
167 }

```

```

168 |
169 public void sortDescRating() {
170     Soal2NodeRidwan current, index;
171     Soal2DataFilm tmp;
172     if (isEmpty()) {
173         System.out.println("Data Linked list kosong");
174     } else {
175         for (current = head; current.next != null; current = current.next) {
176             for (index = current.next; index != null; index = index.next) {
177                 if (current.data.rating < index.data.rating) {
178                     tmp = current.data;
179                     current.data = index.data;
180                     index.data = tmp;
181                 }
182             }
183         }
184     }
185     System.out.println("Data diurutkan berdasarkan Rating tertinggi");
186     print();
187 }
188 }

```

## Source code Class Soal2NodeRidwan

```

1 package doublelinkedlists;
2
3 public class Soal2NodeRidwan {
4     Soal2NodeRidwan prev, next;
5     Soal2DataFilm data;
6
7     Soal2NodeRidwan(Soal2NodeRidwan prev, Soal2DataFilm data, Soal2NodeRidwan next) {
8         this.prev = prev;
9         this.data = data;
10        this.next = next;
11    }
12 }

```

## Output Menu Awal

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

## Output Tambah Data Awal

```
=====
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
```

## Output Tambah Data Akhir

```
=====
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
```



## Output Tambah Data Index Tertentu

```
=====
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
```

## Output Hapus Data Pertama

```
=====
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
=====
4
Hapus Data Pertama
Data Film yang dihapus :
ID : 1222
    Judul Film : Spider-Man: No Way Home
    ipk : 8.7
```

## Output Hapus Data Terakhir

```
=====
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
=====
5
Hapus Data Terakhir
Data Film yang dihapus :
ID : 1346
    Judul Film : Uncharted
    ipk : 6.7
```

## Output Hapus Data Tertentu

```
=====
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
=====
6
Masukkan index data yang ingin dihapus :
1
Data Film yang dihapus :
ID : 1765
    Judul Film : Skyfall
    ipk : 7.8
```

## Output Cetak

Sebelumnya data sudah saya inputkan kembali

```
=====
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
```

## Output Cari ID Film

```
=====
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 : 1567
    Judul Film : The Dark Knight Rises
    IMDB Rating : 8.4
```

## Output Urut Data Rating-DESC

```
=====
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
=====
9
Data diurutkan berdasarkan Rating tertinggi
ID: 1222
    Judul Film: Spider-Man: No Way Home
    ipk: 8.7
ID: 1567
    Judul Film: The Dark Knight Rises
    ipk: 8.4
ID: 1765
    Judul Film: Skyfall
    ipk: 7.8
ID: 1346
    Judul Film: Uncharted
    ipk: 6.7
ID: 1234
    Judul Film: Death on The Nile
    ipk: 6.6
```

## Output 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
=====
10
Program akan ditutup

Process finished with exit code 0
```