



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

12.2 Praktikum 1

```
P12 > DoubleLinkedList > J Node25.java > ...
1 package P12.DoubleLinkedList;
2
3 public class Node25 {
4     int data;
5     Node25 prev, next;
6
7     Node25(Node25 prev, int data, Node25 next){
8         this.prev = prev;
9         this.data = data;
10        this.next = next;
11    }
12 }
13
```

```
package P12.DoubleLinkedList;

public class DoubleLinkedList25 {
    Node25 head;
    int size;

    public DoubleLinkedList25(){
        head = null;
        size = 0;
    }

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

    public void addFirst(int item){
        if (isEmpty()){
            head = new Node25(prev:null, item, next:null);
        }else{
            Node25 newNode = new Node25(prev:null, item, head);
            head.prev = newNode;
            head = newNode;
        }
        size++;
    }

    public void addLast(int item){
        if (isEmpty()){
            addFirst(item);
        }else{
            Node25 current = head;
            while (current.next != null){
                current = current.next;
            }
            Node25 newNode = new Node25(current, item, next:null);
            current.next = newNode;
            size++;
        }
    }
}
```

```
public void add(int item, int index) throws Exception {
    if (isEmpty()){
        addFirst(item);
    }else if (index < 0 || index > size){
        throw new Exception(message:"Nilai indeks di luar batas");
    }else{
        Node25 current = head;
        int i = 0;
        while(i < index){
            current = current.next;
            i++;
        }
        if(current.prev == null){
            Node25 newNode = new Node25(prev:null, item, current);
            current.prev = newNode;
            head = newNode;
        }else{
            Node25 newNode = new Node25(current.prev, item, current);
            newNode.prev = current.prev;
            newNode.next = current;
            current.prev.next = newNode;
            current.prev = newNode;
        }
        size++;
    }
}

public int size(){
    return size;
}

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



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

```
public void print(){
    if(!isEmpty()){
        Node25 tmp = head;
        while (tmp != null){
            System.out.print(tmp.data+"\t");
            tmp = tmp.next;
        }
        System.out.println(x:"\nberhasil diisi");
    }else{
        System.out.println(x:"Linked List Kosong");
    }
}
```

```
package P12.DoubleLinkedList;

public class DoubleLinkedList25Main {
    Run | Debug
    public static void main(String[] args) throws Exception {
        DoubleLinkedList25 dll = new DoubleLinkedList25();
        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());
    }
}
```

12.2.2 Verifikasi Hasil Percobaan

```
PS D:\PrakASD_1F_25> d.; cd 'd:\50a0db594819182e\redhat.java\jdt_
Linked List Kosong
Size : 0
=====
7      3      4
berhasil diisi
Size : 3
=====
7      40     3      4
berhasil diisi
size : 4
=====
Linked List Kosong
Size : 0
PS D:\PrakASD_1F_25>
```

11.2.3 Question

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

Answer:

Single linked list memiliki satu buah pointer yaitu next, sedangkan Double linked list memiliki 2 buah pointer yaitu next dan prev.

2. Perhatikan class Node, di dalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?

Answer:



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

Pointer next untuk menunjuk ke node setelahnya, dan pointer prev untuk menunjuk pada node sebelumnya.

3. Perhatikan konstruktor pada class DoubleLinkedLists. Apa kegunaan inisialisasi atribut head dan size seperti pada gambar berikut ini?

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

Answer :

Potongan kode tersebut berfungsi sebagai kondisi awal dari program tersebut, fungsi inisialisasi head == null untuk head belum menyimpan pada node, sedangkan fungsi inisialisasi size == 0 untuk menyimpan jumlah data pada linked list.

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

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

Answer :

Karena method addFirst() diatas digunakan untuk menambah node pada awal atau index 0, dimana nantinya node yang dimasukkan tersebut prevnya bernilai null karena tidak merujuk atau menyimpan nilai dari node manapun.

5. Perhatikan pada method addFirst(). Apakah arti statement head.prev = newNode ?

Answer :

Potongan kode diatas berfungsi untuk pointer prev pada data yang sudah ada sebelumnya akan merujuk ke node yang baru.

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);
```

Answer :

Karena method addLast() diatas digunakan untuk menambahkan node pada index terakhir, dimana parameter prev dengan current digunakan untuk menyimpan node sebelumnya, dan nantinya terjadi proses penambahan data pada index akhir, sedangkan parameter next dengan null karena tidak merujuk pada node manapun.

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

```
while (i < index) {  
    current = current.next;  
    i++;  
}  
if (current.prev == null) {  
    Node newNode = new Node(null, item, current);  
    current.prev = newNode;  
    head = newNode;  
} else {  
    Node newNode = new Node(current.prev, item, current);  
    current.prev.next = newNode;  
    current.prev = newNode;  
}
```

Answer:



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

Bagian kode yang ditandai dengan kotak kuning bertanggung jawab untuk menambahkan node baru di awal linked list dan memastikan bahwa semua pointer diatur dengan benar untuk menjaga struktur double linked list.

12.3 Praktikum 2

```
public void removeFirst() throws Exception {
    if (isEmpty()) {
        throw new Exception("Linked List Masih kosong, tidak dapat dihapus!");
    } else if (size == 1) {
        removeLast();
    } else {
        head = head.next;
        head.prev = null;
        size--;
    }
}

public void removeLast() throws Exception {
    if (isEmpty()) {
        throw new Exception("Linked list masih kosong, tidak dapat dihapus!");
    } else if (head.next == null) {
        head = null;
        size--;
        return;
    }
    Node2 current = head;
    while (current.next.next != null) {
        current = current.next;
    }
    current.next = null;
    size--;
}
```

```
public void remove(int index) throws Exception {
    if (isEmpty() || index >= size) {
        throw new Exception("Nilai indeks di luar batas");
    } else if (index == 0) {
        removeFirst();
    } else {
        Node2 current = head;
        int i = 0;
        while (i < index) {
            current = current.next;
            i++;
        }
        if (current.next == null) {
            current.prev.next = null;
        } else if (current.prev == null) {
            current = current.next;
            current.prev = null;
            head = current;
        } else {
            current.prev.next = current.next;
            current.next.prev = current.prev;
        }
        size--;
    }
}
```

```
package org.bonanza.praktikum2;

import java.util.Scanner;

public class DoubleLinkedListMain {
    @SuppressWarnings("resource")
    public static void main(String[] args) throws Exception {
        DoubleLinkedList dll = new DoubleLinkedList25();
        dll.print();
        dll.addLast(item:50);
        dll.addLast(item:40);
        dll.addLast(item:10);
        dll.addLast(item:20);
        dll.print();
        System.out.println("Size : " + dll.size());
        System.out.println("x:-----");
        dll.removeFirst();
        dll.print();
        System.out.println("Size : " + dll.size());
        System.out.println("x:-----");
        dll.removeLast();
        dll.print();
        System.out.println("Size : " + dll.size());
        System.out.println("x:-----");
        dll.remove(index:1);
        dll.print();
        System.out.println("Size : " + dll.size());
    }
}
```

12.3.2 Verifikasi Hasil Percobaan

```
PS D:\PrakASD_1F_25> & 'C:\Program Files\Java\jdk-11.0.2\bin\java.exe' -cp 'D:\PrakASD_1F_25_3f0e4e3b\bin' org.bonanza.praktikum2.DoubleLinkedListMain
Linked List Kosong
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 D:\PrakASD_1F_25>
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

11.3.3 Pertanyaan

1. Apakah maksud statement berikut pada method removeFirst()?

```
head = head.next;  
head.prev = null;
```

Answer:

head = head.next bermaksud untuk memindahkan pointer head ke node berikutnya, secara efektif menghapus referensi ke node pertama sebelumnya. head.prev = null bermaksud untuk memastikan bahwa node baru pertama tidak memiliki node sebelumnya.

2. Bagaimana cara mendeteksi posisi data ada pada bagian akhir pada method removeLast()?

Answer :

Dengan cara mencari posisi node yang pointer next nya bernilai null, jika sudah ditemukan maka bisa dipastikan node tersebut berada pada index terakhir.

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;
```

Answer :

Potongan kode diatas tidak cocok untuk perintah remove, karena potongan kode tersebut cocok digunakan untuk removeFirst dimana tmp menyimpan data setelah head, selanjutnya head.next menyimpan data pada tmp.next, kemudian pinter prev pada tmp.next menunjuk ke head.

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

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

Answer:

Fungsi potongan kode tersebut untuk merubah nilai yang tadinya ada pada current.prev.next atau pointer next pada node sebelumnya akan dipindah pada current.next. sedangkan pada current.next.prev atau pointer prev pada node selanjutnya akan dipindah pada current.prev.



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

12.4 Praktikum 3

```
public int getFirst() throws Exception {
    if(isEmpty()){
        throw new Exception(message:"Linked list Kosong");
    }
    return head.data;
}

public int getLast() throws Exception {
    if(isEmpty()){
        throw new Exception(message:"Linked list Kosong");
    }
    Node<T> tmp = head;
    while (tmp.next != null){
        tmp = tmp.next;
    }
    return tmp.data;
}

public int get(int index) throws Exception{
    if(isEmpty() || index >= size){
        throw new Exception (message:"nilai indeks diluar batas");
    }
    Node<T> tmp = head;
    for (int i =0; i<index; i++){
        tmp = tmp.next;
    }
    return tmp.data;
}
```

```
1 package PoliteknikNegeriMalang;
2
3 public class DoubleLinkedListTest {
4     Run|Debug
5     public static void main(String[] args) throws Exception {
6         DoubleLinkedList dll = new DoubleLinkedList(25);
7
8         dll.print();
9         System.out.println("Size : "+dll.size());
10        System.out.println("x:=====");
11        dll.addFirst(item:3);
12        dll.addLast(item:4);
13        dll.addFirst(item:7);
14        dll.print();
15        System.out.println("Size : "+dll.size());
16        System.out.println("x:=====");
17        dll.add(item:40, index:1);
18        dll.print();
19        System.out.println("Size : "+dll.size());
20        System.out.println("x:=====");
21        System.out.println("Data Awal pada linked list adalah : "+dll.getFirst());
22        System.out.println("Data akhir pada linked list adalah : "+ dll.getLast());
23        System.out.println("Data Pada indeks ke-1 pada linked list adalah : "+dll.get(index:1));
24    }
25 }
```

12.4.2 Verifikasi Hasil Percobaan

```
Linked List Kosong
Size : 0
=====
7      3      4
berhasil diisi
Size : 3
=====
7      40     3      4
berhasil diisi
Size : 4
=====
Data Awal pada linked list adalah : 7
Data akhir pada linked list adalah : 4
Data Pada indeks ke-1 pada linked list adalah : 40
PS D:\PrakASD_1F_25>
```

12.4.3 Pertanyaan

1. Jelaskan method size() pada class DoubleLinkedLists!

Answer :

Method size() berfungsi untuk mereturn nilai menjadi 0 jika dalam kondisi double linked list kosong.

2. Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke- 1!

Answer :

Mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke- 1 dilakukan dengan cara melakukan perulangan dimana yang diinisialisasikan dengan index sama dengan 1.



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

3. Jelaskan perbedaan karakteristik fungsi Add pada Double Linked Lists dan Single Linked Lists!

Answer :

Fungsi add pada double linked list hanya ada 1 fungsi yang berdasarkan indeks sisanya dapat ditambah dari awal atau akhir index. Sedangkan fungsi add pada single linked list terdapat 3 fungsi yaitu insertAfter, insertBefore, dan insertAt.

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)

Answer :

Pada gambar a terdapat if else yang dimana jika size = 0 maka true sedangkan tidak maka false. Dan pada gambar b tidak menggunakan if else, hanya langsung apakah head bernilai null jika benar maka kondisi nya adalah true.

12.5 Tugas Praktikum

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) Contoh 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  
-*****
```

```
*****  
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
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

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

Answer :

```
package P12.TugasPraktikum.Tugas1;

public class Node25 {
    int nomor;
    String nama;

    Node25 prev, next;

    Node25(Node25 prev, int nomor, String nama, Node25 next){
        this.prev = prev;
        this.nomor = nomor;
        this.nama = nama;
        this.next = next;
    }
}

package P12.TugasPraktikum.Tugas1;

public class AntrianVaksinasi25 {
    Node25 head;
    int size;

    public AntrianVaksinasi25(){
        head = null;
        size = 1;
    }

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

    public void addLast(int nomor, String nama){
        if (isEmpty()){
            head = new Node25(prev:null, nomor, nama, next:null);
        }else{
            Node25 current = head;
            while (current.next != null){
                current = current.next;
            }
            Node25 newNode = new Node25(current, nomor, nama,next:null);
            current.next = newNode;
            size++;
        }
    }

    public void removeFirst() throws Exception{
        if (isEmpty()){
            throw new Exception(message:"Linked List Masih kosong, tidak dapat dihapus!");
        }else if (size == 1){
            removeLast();
        }else{
            System.out.println(head.nama+" telah selesai divaksin");
            head = head.next;
            head.prev = null;
            size--;
        }
    }
}
```




NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

```
public void removeLast() throws Exception {
    if (isEmpty()) {
        throw new Exception("Linked list masih kosong, tidak dapat dihapus!");
    } else if (head.next == null) {
        System.out.println(head.nama + " telah selesai divaksinasi");
        head = null;
        size--;
        return;
    }
    Node current = head;
    while (current.next.next != null) {
        current = current.next;
    }
    current.next = null;
    size--;
}

public void print() {
    System.out.println(x: "=====");
    System.out.println(x: "DAFTAR PENGANTRI VAKSIN ");
    System.out.println(x: "=====");
    System.out.printf(format: "%-6s | %-10s | %n", ...args: "No", "Nama");
    if (isEmpty()) {
        Node tmp = head;
        while (tmp != null) {
            System.out.printf(format: "%-6d | %-10s | %n", tmp.nomor, tmp.nama);
            tmp = tmp.next;
        }
        System.out.println("\nSisa Antrian " + size);
    } else {
        System.out.println(x: "Linked List Kosong");
    }
}
}
```

```
1 package P12.TugasPraktikum.Tugas;
2
3 import java.util.Scanner;
4
5 public class AntrianVaksinasi125Main {
6     public static void menu() {
7         System.out.println(x: "=====");
8         System.out.println(x: "PENGANTRI VAKSIN EXTRAVAGANZA");
9         System.out.println(x: "=====");
10        System.out.println(x: "1. Tambah Data Penerima Vaksin");
11        System.out.println(x: "2. Hapus Data Pengantri Vaksin");
12        System.out.println(x: "3. Daftar Penerima Vaksin");
13        System.out.println(x: "4. Keluar");
14        System.out.println(x: "=====");
15    }
16
17    public static void main(String[] args) throws Exception {
18        Scanner sc25 = new Scanner(System.in);
19
20        AntrianVaksinasi125 dll = new AntrianVaksinasi125();
21
22        int pilih;
23        do {
24            menu();
25            pilih = sc25.nextInt();
26            sc25.nextLine();
27
28            switch (pilih) {
29                case 1:
30                    System.out.print(s: "Nomor Antrian\t: ");
31                    int nim = sc25.nextInt();
32                    sc25.nextLine();
33                    System.out.print(s: "Nama Penerima\t: ");
34                    String nama = sc25.nextLine();
35                    dll.addLast(nim, nama);
36                    break;
37
38                case 2:
39                    dll.removeFirst();
40                    dll.print();
41                    break;
42
43                case 3:
44                    dll.print();
45                    break;
46
47                case 4:
48                    System.exit(status: 0);
49                    break;
50            }
51        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4);
52
53        sc25.close();
54    }
55 }
```

```
PS D:\PrakASD_1F_25> & 'C:\Program Files\Java\jdk-11.0.10\bin\java.exe' -cp 'C:\Users\raktikum\Tugas1\AntrianVaksinasi125Main\build\classes\main\classes' P12.TugasPraktikum.Tugas125Main
+-----+
1 Nomor Antrian : 123
2 Nama Penerima : Joko
+-----+
PENGANTRI VAKSIN EXTRAVAGANZA
+-----+
1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+-----+
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

```
=====
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

=====
PENGANTRI VAKSIN EXTRAVAGANZA
=====
1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
=====
2
=====
Joko telah selesai divaksin
=====
DAFTAR PENGANTRI VAKSIN
=====
| No | Nama |
| 124 | Mely |
| 135 | Johan |
| 146 | Rosi |
=====
Sisa Antrian 3

=====
PENGANTRI VAKSIN EXTRAVAGANZA
=====
1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
=====
4
PS D:\PrakASD_1F_25>
```

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
=====
3
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
=====
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
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

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
```

Answer :

```
package P12.TugasPraktikum.Tugas2;

public class Node25 {
    int id;
    String judulFilm;
    double rating;

    Node25 prev, next;

    Node25(Node25 prev, int id, String judulFilm, double rating, Node25 next){
        this.prev = prev;
        this.id = id;
        this.judulFilm = judulFilm;
        this.rating = rating;
        this.next = next;
    }
}
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

```
package f11; import java.io.*; public class DaftarFilm { private Node head; private int size; public DaftarFilm() { head = null; size = 0; } public boolean isEmpty() { return head == null; } public void addFirst(int id, String judulFilm, double rating) { if (isEmpty()) { head = new Node25(prevnull, id, judulFilm, rating, nextnull); } else { Node25 newnode = new Node25(prevnull, id, judulFilm, rating, head); head.prev = newnode; head = newnode; } size++; } public void addLast(int id, String judulFilm, double rating) { if (isEmpty()) { addFirst(id, judulFilm, rating); } else { Node25 current = head; while (current.next != null) { current = current.next; } Node25 newnode = new Node25(current, id, judulFilm, rating, nextnull); current.next = newnode; size++; } } public void removeLast() throws Exception { if (isEmpty()) { throw new Exception("Linked list masih kosong, tidak dapat dihapus"); } else if (head.next == null) { head = null; size--; return; } Node25 current = head; while (current.next.next != null) { current = current.next; } current.next = null; size--; } public void remove(int index) throws Exception { if (isEmpty() || index > size) { throw new Exception("Nilai indeks di luar batas"); } else if (index == 0) { removeFirst(); } else { Node25 current = head; int i = 0; while (i < index) { current = current.next; i++; } if (current.next == null) { current.prev.next = null; } else if (current.prev == null) { current = current.next; current.prev = null; } else { current.prev.next = current.next; current.next.prev = current.prev; } size--; } } public void search() { Node25 current = null; int index = null; int tempID; Scanner tempID; Scanner tempR; if (head == null) { return; } else { for (current = head; current.next != null; current = current.next) { if (current.rating < index.rating) { tempID = current.rating; current.rating = index.rating; index.rating = tempID; tempID = current.id; current.id = index.id; index.id = tempID; tempID = current.judulFilm; current.judulFilm = index.judulFilm; index.judulFilm = tempID; } } print(); } } } package f11; import java.io.*; import java.util.Scanner; public class DaftarFilmMain { public static void main() { System.out.println("-----"); System.out.println("Data Film Layer Lebar "); System.out.println("-----"); System.out.println(x:"1. Tambah Data Awal"); System.out.println(x:"2. Tambah Data Akhir"); System.out.println(x:"3. Tambah Data Index Tertentu"); System.out.println(x:"4. Hapus Data Pertama"); System.out.println(x:"5. Hapus Data Terakhir"); System.out.println(x:"6. Hapus Data Tertentu"); System.out.println(x:"7. Cetak"); System.out.println(x:"8. Cari ID Film"); System.out.println(x:"9. Urut Data Rating Film-DESC"); System.out.println(x:"10. Keluar"); System.out.println("-----"); } Run/Debug public static void main(String[] args) throws Exception { Scanner sc25 = new Scanner(System.in); DaftarFilm dll = new DaftarFilm25(); int pilih; public static void main(String[] args) throws Exception { case 4: dll.removeFirst(); dll.print(); break; case 5: dll.removeLast(); dll.print(); break; case 6: Scanner out.println(x:"Hapus Data File yang diinginkan"); out.println(x:"index ke - "); int index = sc25.nextInt(); dll.remove(index); dll.print(); break; case 7: dll.print(); break; case 8: Scanner out.println(x:"Cari ID Film yang ingin dicari"); int cari = sc25.nextInt(); int id = dll.findSearch(cari); dll.tampilPosisi(cari, id); break; case 9: Scanner out.println(x:"Data akan diurut secara DESC"); dll.sort(); break; case 10: Scanner out.println(x:"Data akan diurut ke - "); while (pilih != 10); sc25.close(); } }
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

```
=====
1
Masukkan Data Film Posisi Awal
ID Film      : 1222
Judul Film   : Spider-Man: No Way Home
Rating      : 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 Film Posisi Akhir
ID Film      : 1346
Judul Film   : Uncharted
Rating      : 6.7
=====
3
Masukkan Data Film Posisi yang diinginkan
Urutan ke - 3
ID Film      : 1234
Judul Film   : Death on The Nile
Rating      : 6.6
Data Film ini akan masuk di urutan ke - 3
=====
4
Data Saat Ini Menjadi
ID      : 1222
Judul Film : Spider-Man: No Way Home
Rating   : 7.6
=====
ID      : 1765
Judul Film : Skyfall
Rating   : 7.8
=====
ID      : 1567
Judul Film : The Dark Knight Rises
Rating   : 8.4
=====
ID      : 1234
Judul Film : Death on The Nile
Rating   : 6.6
=====
ID      : 1346
Judul Film : Uncharted
Rating   : 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
=====
4
Data Saat Ini Menjadi
ID      : 1765
Judul Film : Skyfall
Rating   : 7.8
=====
ID      : 1567
Judul Film : The Dark Knight Rises
Rating   : 8.4
=====
ID      : 1234
Judul Film : Death on The Nile
Rating   : 6.6
=====
ID      : 1346
Judul Film : Uncharted
Rating   : 6.7
=====
5
Data Saat Ini Menjadi
ID      : 1765
Judul Film : Skyfall
Rating   : 7.8
=====
ID      : 1567
Judul Film : The Dark Knight Rises
Rating   : 8.4
=====
ID      : 1234
Judul Film : Death on The Nile
Rating   : 6.6
=====
ID      : 1346
Judul Film : Uncharted
Rating   : 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
=====
6
hapus Data Film yang diinginkan
index ke - 1
=====
Data Saat Ini Menjadi
ID      : 1765
Judul Film : Skyfall
Rating   : 7.8
=====
ID      : 1234
Judul Film : Death on The Nile
Rating   : 6.6
=====
=====
```



NAMA : SALSABILA MAHDA RUNISHA
NIM : 2341720257
NO ABSEN : 25
KELAS : 1F
MATERI : DOUBLE LINKED LIST

```
=====
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
-----
Data Saat Ini Menjadi
ID : 1222
Judul Film : Spider-Man: No Way Home
Rating : 7.6
ID : 1365
Judul Film : Skyfall
Rating : 7.8
ID : 1567
Judul Film : The Dark Knight Rises
Rating : 8.4
ID : 1234
Judul Film : Death on The Nile
Rating : 6.6
ID : 1346
Judul Film : Uncharted
Rating : 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
=====
9
Data Akan diurut secara DESC
-----
Data Saat Ini Menjadi
ID : 1567
Judul Film : The Dark Knight Rises
Rating : 8.4
ID : 1222
Judul Film : Spider-Man: No Way Home
Rating : 7.6
ID : 1222
Judul Film : Spider-Man: No Way Home
Rating : 7.6
ID : 1346
Judul Film : Uncharted
Rating : 6.7
ID : 1234
Judul Film : Death on The Nile
Rating : 6.6
=====
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
PS D:\PrakASD_1F_25>
```