



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

LAPORAN PRAKTIKUM ALGORITMA DAN STRUKTUR DATA

12.2. Kegiatan Praktikum 1

12.2.1. Percobaan 1

Node :

```
src > P12 > doublelinkedlists13 > J Node13.java > ...  
1 package P12.doublelinkedlists13;  
2  
Codeium: Refactor | Explain  
3 public class Node13 {  
4     int data;  
5     Node13 prev, next;  
6  
7     public Node13(Node13 prev, int data, Node13 next) {  
8         this.prev = prev;  
9         this.data = data;  
10        this.next = next;  
11    }  
12  
13 }  
14
```

Double Linked Lists :

```
src > P12 > doublelinkedlists13 > J DoubleLinkedLists13.java > DoubleLinkedLists13 > add(int, int)  
1 package P12.doublelinkedlists13;  
2  
Codeium: Refactor | Explain  
3 public class DoubleLinkedLists13 {  
4     Node13 head;  
5     int size;  
6  
7     public DoubleLinkedLists13() {  
8         head = null;  
9         size = 0;  
10    }  
11  
Codeium: Refactor | Explain | Generate Javadoc | X  
12 public boolean isEmpty() {  
13     return head == null;  
14 }  
15  
Codeium: Refactor | Explain | Generate Javadoc | X  
16 public void addFirst(int item) {  
17     if (isEmpty()) {  
18         head = new Node13(prev:null, item, next:null);  
19     } else {  
20         Node13 newNode = new Node13(prev:null, item, head);  
21         head.prev = newNode;  
22         head = newNode;  
23     }  
24     size++;  
25 }  
26  
Codeium: Refactor | Explain | Generate Javadoc | X  
27 public void addLast(int item) {  
28     if (isEmpty()) {  
29         addFirst(item);  
30     } else {  
31         Node13 current = head;  
32         while (current.next != null) {  
33             current = current.next;  
34         }  
35         Node13 newNode = new Node13(current, item, next:null);  
36         current.next = newNode;  
37         size++;  
38     }  
39 }  
40
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

```
Codeium: Refactor | Explain | Generate Javadoc | X
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(message:"Nilai indeks di luar batas");
46     } else {
47         Node13 current = head;
48         int i = 0;
49         while (i < index) {
50             current = current.next;
51             i++;
52         }
53         if (current.prev == null) {
54             Node13 newNode = new Node13(prev:null, item, current);
55             current.prev = newNode;
56             head = newNode;
57         } else {
58             Node13 newNode = new Node13(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 }
67
Codeium: Refactor | Explain | Generate Javadoc | X
68 public int size() {
69     return size;
70 }
71
Codeium: Refactor | Explain | Generate Javadoc | X
72 public void clear() {
73     head = null;
74     size = 0;
75 }
76
Codeium: Refactor | Explain | Generate Javadoc | X
77 public void print() {
78     if (!isEmpty()) {
79         Node13 current = head;
80         while (current != null) {
81             System.out.print(current.data + "\t");
82             current = current.next;
83         }
84         System.out.println(x:"\nberhasil diisi");
85     } else {
86         System.out.println(x:"Linked Lists Kosong");
87     }
88 }
89 }
90
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

Main :

```
src > P12 > doublelinkedlists13 > J DoubleLinkedListsMain13.java > ...
1  package P12.doublelinkedlists13;
2
   Codeium: Refactor | Explain
3  public class DoubleLinkedListsMain13 {
   Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
4      public static void main(String[] args) throws Exception {
5          DoubleLinkedLists13 dll = new DoubleLinkedLists13();
6          dll.print();
7          System.out.println("Size : " + dll.size());
8          System.out.println(x:"=====");
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(x:"=====");
15         dll.add(item:40, index:1);
16         dll.print();
17         System.out.println("Size : " + dll.size());
18         System.out.println(x:"=====");
19         dll.clear();
20         dll.print();
21         System.out.println("Size : " + dll.size());
22         System.out.println(x:"=====");
23     }
24 }
25
```

Output :

```
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 E:\KULIAH 2\Pratikum Algoritma dan Struktur Data\PrakASD_1F_13>
```

Question :

1. Jelaskan perbedaan antara single linked list dengan double linked lists!
Jawab : Single Linked List terdapat pointer head dan tail dan hanya dapat terhubung dengan node selanjutnya. Sedangkan Double Linked Lists memiliki pointer next dan prev serta memiliki dua arah ke node sesudah dan sebelumnya.
2. Perhatikan class Node, di dalamnya terdapat atribut next dan prev. Untuk apakah atribut tersebut?
Jawab : Untuk menghubungkan alamat node ke node setelahnya dan node sebelumnya.



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

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

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

Jawab : **head = null** : Untuk menandakan bahwa Linked List awal masih kosong.

dan **size = 0** : Untuk memastikan bahwa Linked List masih kosong dan dapat dimulai dari nol, serta untuk menandakan ukuran yang berarada di 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);
```

Jawab : karena pada addFirst ini menginputkan data pertama yang berarti data itu berada di head dan tidak memiliki nilai prev. Oleh karena itu di prev adalah null.

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

Jawab : untuk mengatur prev dari node yang saat ini mejadi head untuk menunjukkan ke newNode yaitu, menghubungkan node sebelumnya (head) ke node baru sebagai node sebelumnya.

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);
```

Jawab : Untuk membuat node baru dengan prev yang menghubungkan ke data linked list paling akhir, lalu menginputkan nilai item dan next yang bernilai null karena data ini ditempatkan menjadi paling akhir, maka bernilai null yang berarti tidak ada node setelahnya.

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;  
}
```

jelaskan maksud dari bagian yang ditandai dengan kotak kuning.

Jawab : bagian ini untuk menginputkan node baru di bagian pertama linked list. Lalu menghubungkan node lama dengan node yang baru sebagai node sebelumnya(prev) dan mengubah head menjadi newNode yang berarti node baru menjadi head.



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

12.3. Kegiatan Praktikum 2

12.3.1. Tahapan Percobaan

Double Linked Lists :

```
90 public void removeFirst() throws Exception {  
91     if (isEmpty()) {  
92         throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus!");  
93     } else if (size == 1) {  
94         head = null;  
95     } else {  
96         head = head.next;  
97         head.prev = null;  
98         size--;  
99     }  
100 }  
101  
Codeium: Refactor | Explain | Generate Javadoc | X  
102 public void removeLast() throws Exception {  
103     if (isEmpty()) {  
104         throw new Exception(message:"Linked List masih kosong< tidak dapat dihapus!");  
105     } else if (head.next == null) {  
106         head = null;  
107         size--;  
108         return;  
109     }  
110     Node13 current = head;  
111     while (current.next.next != null) {  
112         current = current.next;  
113     }  
114     current.next = null;  
115     size--;  
116 }  
117
```

```
118 public void remove(int index) throws Exception {  
119     if (isEmpty() || index >= size) {  
120         throw new Exception(message:"Nilai indeks di luar batas");  
121     } else if (index == 0) {  
122         removeFirst();  
123     } else {  
124         Node13 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 }  
143 }  
144
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

Main :

```
src > P12 > doublelinkedlists13 > J DoubleLinkedListsMain13.java > DoubleLinkedListsMain13 > main(String[])
1 package P12.doublelinkedlists13;
2
3 Codeium: Refactor | Explain
4 public class DoubleLinkedListsMain13 {
5     Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
6     public static void main(String[] args) throws Exception {
7         DoubleLinkedLists13 dll = new DoubleLinkedLists13();
8         dll.addLast(item:50);
9         dll.addLast(item:40);
10        dll.addLast(item:10);
11        dll.addLast(item:20);
12        dll.print();
13        System.out.println("Size : " + dll.size());
14        System.out.println(x:"=====");
15        dll.removeFirst();
16        dll.print();
17        System.out.println("Size : " + dll.size());
18        System.out.println(x:"=====");
19        dll.removeLast();
20        dll.print();
21        System.out.println("Size : " + dll.size());
22        System.out.println(x:"=====");
23        dll.remove(index:1);
24        dll.print();
25        System.out.println("Size : " + dll.size());
26    }
}
```

Output :

```
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 E:\KULIAH 2\Pratikum Algoritma dan Struktur Data\PrakASD_1F_13>
```

Question :

1. Apakah maksud statement berikut pada method **removeFirst()**?
head = head.next;
head.prev = null;

Jawab : Bagian ini mengubah head ke node selanjutnya dan mengubah prev yang menjadi head saat ini menjadi null.



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

2. Bagaimana cara mendeteksi posisi data ada pada bagian akhir pada method **removeLast()**?
Jawab : menggunakan traverse yang dimana mengecek data sampai data 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;
```

Jawab : kode tersebut hanya menangani penghapusan pada node kedua dari linked list. Jika ingin menghapus node selain node kedua, maka kode ini tidak akan bekerja karena tmp = head.next selalu benar hanya jika kita menghapus node kedua. Kode ini tidak dapat digunakan untuk menghapus node pertama, terakhir, dan node di posisi tertentu.

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

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

Jawab : untuk menghapus node current dari linked list dengan menghubungkan node sebelum current dengan node setelah current.



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

12.4. Kegiatan Praktikum 3

12.4.1. Tahapan Percobaan

Double Linked Lists :

```
144 public int getFirst() throws Exception {
145     if (isEmpty()) {
146         throw new Exception(message:"Linked List kosong");
147     }
148     return head.data;
149 }
150
151 Codeium: Refactor | Explain | Generate Javadoc | X
152 public int getLast() throws Exception {
153     if (isEmpty()) {
154         throw new Exception(message:"Linked List kosong");
155     }
156     Node13 current = head;
157     while (current.next != null) {
158         current = current.next;
159     }
160     return current.data;
161 }
162
163 Codeium: Refactor | Explain | Generate Javadoc | X
164 public int get(int index) throws Exception {
165     if (isEmpty() || index >= size) {
166         throw new Exception(message:"Nilai indeks di luar batas.");
167     }
168     Node13 current = head;
169     for (int i = 0; i < index; i++) {
170         current = current.next;
171     }
172     return current.data;
173 }
```

Main :

```
src > P12 > doublelinkedlists13 > J DoubleLinkedListsMain13.java > ...
1 package P12.doublelinkedlists13;
2
3 Codeium: Refactor | Explain
4 public class DoubleLinkedListsMain13 {
5     Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
6     public static void main(String[] args) throws Exception {
7         DoubleLinkedLists13 dll = new DoubleLinkedLists13();
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 Lists adalah: " + dll.getFirst());
22        System.out.println("Data akhir pada Linked Lists adalah: " + dll.getLast());
23        System.out.println("Data indeks ke-1 pada Linked Lists adalah: " + dll.get(index:1));
24    }
```




NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

Question :

1. Jelaskan method size() pada class DoubleLinkedLists!
Jawab : untuk mengembalikan jumlah node dalam linked list.
2. Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke-1!
Jawab : dengan mengubah variabel l diberi nilai 1 pada perulangan method get.
3. Jelaskan perbedaan karakteristik fungsi **Add** pada Double Linked Lists dan Single Linked Lists!
Jawab : perbedaan add yaitu pada Double Linked List memerlukan dua pointer yaitu next dan prev, sedangkan Single Linked List hanya memerlukan satu pointer next saja.
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)

Jawab :

- a. kode tersebut memeriksa nilai atribut size, jika size adalah 0 berarti tidak ada node di dalam linked list, sehingga linked list dianggap kosong.
- b. kode tersebut memeriksa apakah head dari linked list adalah null, jika head adalah null berarti tidak ada node di dalam linked list, sehingga linked list dianggap kosong.



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

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|
```

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
```

Jawab :



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

Class Node :

```
src > P12 > Vaksinasi13 > J Node13.java > Node13 > Node13(Node13, Antrian13, Node13)
1 package P12.Vaksinasi13;
2
3 Codeium: Refactor | Explain
4 public class Node13 {
5     Antrian13 pengantri;
6     Node13 prev, next;
7
8     public Node13(Node13 prev, Antrian13 pengantri, Node13 next) {
9         this.prev = prev;
10        this.pengantri = pengantri;
11        this.next = next;
12    }
13
14 }
15
```

Class Antrian:

```
src > P12 > Vaksinasi13 > J Antrian13.java > ...
1 package P12.Vaksinasi13;
2
3 Codeium: Refactor | Explain
4 public class Antrian13 {
5     int noAntrian;
6     String nama;
7
8     public Antrian13(int noAntrian, String nama) {
9         this.noAntrian = noAntrian;
10        this.nama = nama;
11    }
12 }
13
```

Class Double Linked Lists :

```
src > P12 > Vaksinasi13 > J DoubleLinkedLists13.java > DoubleLinkedLists13
1 package P12.Vaksinasi13;
2
3 Codeium: Refactor | Explain
4 public class DoubleLinkedLists13 {
5     Node13 head;
6     int size;
7
8     public DoubleLinkedLists13() {
9         head = null;
10        size = 0;
11    }
12
13    Codeium: Refactor | Explain | Generate Javadoc | X
14    public boolean isEmpty() {
15        return head == null;
16    }
17
18    Codeium: Refactor | Explain | Generate Javadoc | X
19    void Enqueue(Antrian13 data) {
20        if (isEmpty()) {
21            head = new Node13(prev:null, data, next:null);
22        } else {
23            Node13 current = head;
24            while (current.next != null) {
25                current = current.next;
26            }
27            Node13 newNode = new Node13(current, data, next:null);
28            current.next = newNode;
29            size++;
30        }
31    }
32
33    Codeium: Refactor | Explain | Generate Javadoc | X
34    public void Dequeue() throws Exception {
35        if (isEmpty()) {
36            throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus!");
37        } else if (size == 1) {
38            System.out.println(head.pengantri.nama + " telah selesai divaksinasi.");
39            head = null;
40        } else {
41            System.out.println(head.pengantri.nama + " telah selesai divaksinasi.");
42            head = head.next;
43            head.prev = null;
44            size--;
45        }
46    }
47
48 }
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

```
44 public void print() {  
45     if (isEmpty()) {  
46         Node13 current = head;  
47         System.out.println(x:"+++++++");  
48         System.out.println(x:"Daftar Pengantri Vaksin");  
49         System.out.println(x:"+++++++");  
50         System.out.println(x:"no | nama |");  
51         while (current != null) {  
52             System.out.printf(format:"| %7d| %7s|\n", current.pengantri.noAntrian, current.pengantri.nama);  
53             current = current.next;  
54         }  
55         System.out.println("Sisa Antrian: " + size);  
56     } else {  
57         System.out.println(x:"Linked Lists Kosong");  
58     }  
59 }  
60 }  
61 }
```

Class Main :

```
src > P12 > Vaksinasi13 > J VaksinasiMain13.java > VaksinasiMain13  
1 package P12.Vaksinasi13;  
2  
3 import java.util.Scanner;  
4  
5 Codeium: Refactor | Explain  
6 public class VaksinasiMain13 {  
7     Codeium: Refactor | Explain | Generate Javadoc | X  
8     public static void menu() {  
9         System.out.println(x:"+++++++");  
10        System.out.println(x:"PENGANTRI VAKSIN EXTRAVAGANZA");  
11        System.out.println(x:"+++++++");  
12        System.out.println(x:"1. Tambah Data Penerima Vaksin");  
13        System.out.println(x:"2. Hapus Data Pengantri Vaksin");  
14        System.out.println(x:"3. Daftar Penerima Vaksin");  
15        System.out.println(x:"4. Keluar");  
16        System.out.println(x:"+++++++");  
17        System.out.print(s:"Pilih menu: ");  
18    }  
19  
20    public static void main(String[] args) throws Exception {  
21        Scanner sc = new Scanner(System.in);  
22        DoubleLinkedLists13 antrian = new DoubleLinkedLists13();  
23  
24        while (true) {  
25            menu();  
26            int pilihan = sc.nextInt();  
27  
28            switch (pilihan) {  
29                case 1:  
30                    System.out.println(x:"-----");  
31                    System.out.println(x:"Masukkan Data Penerima Vaksin");  
32                    System.out.println(x:"-----");  
33                    System.out.println(x:"Nomor Antrian: ");  
34                    int noAntri = sc.nextInt();  
35                    sc.nextLine();  
36                    System.out.println(x:"Nama Penerima: ");  
37                    String nama = sc.nextLine();  
38                    Antrian13 antri = new Antrian13(noAntri, nama);  
39                    antrian.Enqueue(antri);  
40                    break;  
41                case 2:  
42                    antrian.Dequeue();  
43                    antrian.print();  
44                    break;  
45                case 3:  
46                    antrian.print();  
47                    break;  
48                case 4:  
49                    System.exit(status:0);  
50                default:  
51                    System.out.println(x:"Pilihan tidak valid, silahkan coba lagi.");  
52            }  
53        }  
54    }
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

Output :

```
+++++
PENGANTRI VAKSIN EXTRAVAGANZA
+++++
1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+++++
Pilih menu: 1
```

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

```
+++++
PENGANTRI VAKSIN EXTRAVAGANZA
+++++
1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar
+++++
Pilih menu: 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
+++++
Pilih menu: 2
Joko telah selesai divaksinasi.
+++++
Daftar Pengantri Vaksin
+++++
| no | nama |
| 124 | Mely |
| 135 | Johan |
| 146 | Rosi |
+++++
Sisa Antrian: 3
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

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
=====
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

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
```

Jawab :

Class DaftarFilm :

```
src > P12 > Film13 > J DaftarFilm13.java > ...
1 package P12.Film13;
2
3 Codeium: Refactor | Explain
4 public class DaftarFilm13 {
5     int id;
6     String judul;
7     double rating;
8
9     public DaftarFilm13(int id, String judul, double rating) {
10         this.id = id;
11         this.judul = judul;
12         this.rating = rating;
13     }
14 }
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

Class Node :

```
src > P12 > Film13 > J Node13.java > ...
1  package P12.Film13;
2
3  Codeium: Refactor | Explain
4  public class Node13 {
5      DaftarFilm13 dataFilm;
6      Node13 prev, next;
7
8      public Node13(Node13 prev, DaftarFilm13 dataFilm, Node13 next) {
9          this.prev = prev;
10         this.dataFilm = dataFilm;
11         this.next = next;
12     }
13 }
14
```

Class DLLFilm :

```
src > P12 > Film13 > J DLLFilm13.java > DLLFilm13 > getID(int)
1  package P12.Film13;
2
3  Codeium: Refactor | Explain
4  public class DLLFilm13 {
5      Node13 head;
6      int size;
7
8      public DLLFilm13() {
9          head = null;
10         size = 0;
11     }
12
13     Codeium: Refactor | Explain | Generate Javadoc | X
14     public boolean isEmpty() {
15         return head == null;
16     }
17
18     Codeium: Refactor | Explain | Generate Javadoc | X
19     public int size() {
20         return size;
21     }
22
23     Codeium: Refactor | Explain | Generate Javadoc | X
24     public void addFirst(DaftarFilm13 data) {
25         if (isEmpty()) {
26             head = new Node13(prev:null, data, next:null);
27         } else {
28             Node13 newNode = new Node13(prev:null, data, head);
29             head.prev = newNode;
30             head = newNode;
31         }
32         size++;
33     }
34
35     Codeium: Refactor | Explain | Generate Javadoc | X
36     public void addLast(DaftarFilm13 data) {
37         if (isEmpty()) {
38             addFirst(data);
39         } else {
40             Node13 current = head;
41             while (current.next != null) {
42                 current = current.next;
43             }
44             Node13 newNode = new Node13(current, data, next:null);
45             current.next = newNode;
46             size++;
47         }
48     }
49 }
```




NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

```
44  
Codeium: Refactor | Explain | Generate Javadoc | X  
45 public void add(DaftarFilm13 data, int index) throws Exception {  
46     if (isEmpty()) {  
47         addFirst(data);  
48     } else if (index < 0 || index >= size) {  
49         throw new Exception(message:"Nilai indeks di luar batas.");  
50     } else {  
51         Node13 current = head;  
52         int i = 0;  
53         while (i < index) {  
54             current = current.next;  
55             i++;  
56         }  
57         if (current.prev == null) {  
58             Node13 newNode = new Node13(prev:null, data, current);  
59             current.prev = newNode;  
60             head = newNode;  
61         } else {  
62             Node13 newNode = new Node13(current.prev, data, current);  
63             current.prev.next = newNode;  
64             current.prev = newNode;  
65         }  
66     }  
67     size++;  
68 }  
69  
Codeium: Refactor | Explain | Generate Javadoc | X  
70 public void removeFirst() throws Exception {  
71     if (isEmpty()) {  
72         throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus!");  
73     } else if (size == 1) {  
74         head = null;  
75     } else {  
76         head = head.next;  
77         head.prev = null;  
78         size--;  
79     }  
80 }  
81  
Codeium: Refactor | Explain | Generate Javadoc | X  
82 public void removeLast() throws Exception {  
83     if (isEmpty()) {  
84         throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus!");  
85     } else if (head.next == null) {  
86         head = null;  
87         size--;  
88         return;  
89     }  
90     Node13 current = head;  
91     while (current.next.next != null) {  
92         current = current.next;  
93     }  
94     current.next = null;  
95     size--;  
96 }  
97  
Codeium: Refactor | Explain | Generate Javadoc | X  
98 public void remove(int index) throws Exception {  
99     if (isEmpty() || index >= size) {  
100         throw new Exception(message:"Nilai indeks di luar batas");  
101     } else if (index == 0) {  
102         removeFirst();  
103     } else {  
104         Node13 current = head;  
105         int i = 0;  
106         while (i < index) {  
107             current = current.next;  
108             i++;  
109         }  
110         if (current.next == null) {  
111             current.prev.next = null;  
112         } else if (current.prev == null) {  
113             current = current.next;  
114             current.prev = null;  
115             head = current;  
116         } else {  
117             current.prev.next = current.next;  
118             current.next.prev = current.prev;  
119         }  
120         size--;  
121     }  
122 }  
123
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

```
124     public void print() {  
125         if (!isEmpty()) {  
126             Node13 current = head;  
127             System.out.println(x:"Cetak Data");  
128             while (current != null) {  
129                 System.out.println("ID : " + current.dataFilm.id);  
130                 System.out.println("\tJudul Film : " + current.dataFilm.judul);  
131                 System.out.println("\tRating Film : " + current.dataFilm.rating);  
132                 current = current.next;  
133             }  
134         } else {  
135             System.out.println(x:"Data Film Kosong");  
136         }  
137     }  
138  
139     Codeium: Refactor | Explain | Generate Javadoc | X  
140     public void getID (int id) throws Exception {  
141         if (isEmpty()) {  
142             throw new Exception(message:"Data Film Kosong");  
143         } else {  
144             Node13 current = head;  
145             for (int i = 0; i < size; i++) {  
146                 if (current.dataFilm.id == id) {  
147                     System.out.println("Data ID Film: " + id + "berada di node ke- " + (i + 1));  
148                     System.out.println(x:"IDENTITAS: ");  
149                     System.out.println("\tID Film : " + current.dataFilm.id);  
150                     System.out.println("\tJudul Film : " + current.dataFilm.judul);  
151                     System.out.println("\tIMDB Rating : " + current.dataFilm.rating);  
152                     break;  
153                 }  
154                 current = current.next;  
155             }  
156         }  
157     }  
158  
159     public void bubbleSortRating() throws Exception {  
160         if (isEmpty()) {  
161             throw new Exception(message:"Data Film Kosong.");  
162         } else {  
163             for (int i = 0; i < size - 1; i++) {  
164                 Node13 current = head;  
165                 for (int j = 0; j < size - 1; j++) {  
166                     if (current.dataFilm.rating < current.next.dataFilm.rating) {  
167                         DaftarFilm13 tmp = current.dataFilm;  
168                         current.dataFilm = current.next.dataFilm;  
169                         current.next.dataFilm = tmp;  
170                     }  
171                     current = current.next;  
172                 }  
173             }  
174         }  
175     }  
176 }  
177
```

Class DLLFilmMain :

```
src > P12 > Film13 > J DLLFilmMain13.java > DLLFilmMain13 > main(String[])  
1 package P12.Film13;  
2  
3 import java.util.Scanner;  
4  
5 Codeium: Refactor | Explain  
6 public class DLLFilmMain13 {  
7     Codeium: Refactor | Explain | Generate Javadoc | X  
8     public static void menu() {  
9         System.out.println(x:"=====");  
10        System.out.println(x:"DATA FILM LAYAR LEBAR");  
11        System.out.println(x:"=====");  
12        System.out.println(x:"1. Tambah Data Awal");  
13        System.out.println(x:"2. Tambah Data Akhir");  
14        System.out.println(x:"3. Tambah Data Index Tertentu");  
15        System.out.println(x:"4. Hapus Data Pertama");  
16        System.out.println(x:"5. Hapus Data Terakhir");  
17        System.out.println(x:"6. Hapus Data Tertentu");  
18        System.out.println(x:"7. Cetak");  
19        System.out.println(x:"8. Cari ID Film");  
20        System.out.println(x:"9. Urut Data Rating Film-DESC");  
21        System.out.println(x:"10. Keluar");  
22        System.out.println(x:"=====");  
23        System.out.print(s:"Pilih menu: ");  
24    }  
25 }
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

```
24 public static void main(String[] args) throws Exception {
25     Scanner input = new Scanner(System.in);
26     DLLFilm13 film = new DLLFilm13();
27
28     while (true) {
29         menu();
30         int pilihan = input.nextInt();
31
32         switch (pilihan) {
33             case 1:
34                 System.out.println(x:"-----");
35                 System.out.println(x:"Masukkan Data Film Awal");
36                 System.out.println(x:"-----");
37                 System.out.println(x:"Id Film : ");
38                 int id = input.nextInt();
39                 input.nextLine();
40                 System.out.println(x:"Judul Film : ");
41                 String judul = input.nextLine();
42                 System.out.println(x:"Rating Film : ");
43                 double rating = input.nextDouble();
44                 DaftarFilm13 filmAwal = new DaftarFilm13(id, judul, rating);
45                 film.addFirst(filmAwal);
46                 break;
47             case 2:
48                 System.out.println(x:"-----");
49                 System.out.println(x:"Masukkan Data Film Akhir");
50                 System.out.println(x:"-----");
51                 System.out.println(x:"Id Film : ");
52                 id = input.nextInt();
53                 input.nextLine();
54                 System.out.println(x:"Judul Film : ");
55                 judul = input.nextLine();
56                 System.out.println(x:"Rating Film : ");
57                 rating = input.nextDouble();
58                 DaftarFilm13 filmAkhir = new DaftarFilm13(id, judul, rating);
59                 film.addLast(filmAkhir);
60                 break;
61
62             case 3:
63                 System.out.println(x:"-----");
64                 System.out.println(x:"Masukkan Data Film Urutan ke-");
65                 System.out.println(x:"-----");
66                 System.out.println(x:"Id Film : ");
67                 id = input.nextInt();
68                 input.nextLine();
69                 System.out.println(x:"Judul Film : ");
70                 judul = input.nextLine();
71                 System.out.println(x:"Rating Film : ");
72                 rating = input.nextDouble();
73                 System.out.print(s:"Data Film ini akan masuk di urutan ke- ");
74                 int index = input.nextInt();
75                 DaftarFilm13 filmIndex = new DaftarFilm13(id, judul, rating);
76                 film.add(filmIndex, index);
77                 break;
78             case 4:
79                 film.removeFirst();
80                 break;
81             case 5:
82                 film.removeLast();
83                 break;
84             case 6:
85                 System.out.print(s:"Data Film dihapus pada urutan ke- ");
86                 index = input.nextInt();
87                 film.remove(index - 1);
88                 break;
89             case 7:
90                 film.print();
91                 break;
92             case 8:
93                 System.out.println(x:"Masukkan ID Film yang dicari : ");
94                 id = input.nextInt();
95                 film.getID(id);
96                 break;
97             case 9:
98                 film.bubbleSortRating();
99                 film.print();
100                 break;
101             case 10:
102                 System.exit(status:0);
103             default:
104                 System.out.println(x:"Pilihan tidak valid, silahkan coba lagi.");
105                 break;
106         }
107     }
108 }
109
```



NAMA : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

Output :

```
=====
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
=====
Pilih menu:
```

```
=====
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
=====
Pilih menu: 2
-----
Masukkan Data Film 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
=====
Pilih menu: 1
-----
Masukkan Data Film 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
=====
Pilih menu: 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 : Gilang Purnomo
NIM : 2341720042
NO ABSEN : 13
KELAS : 1F
MATERI : Double Linked Lists

```
=====
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

```
=====
Pilih menu: 7
```

```
Cetak Data
```

```
ID : 1222
```

```
        Judul Film : Spider-Man: No Way Home
        Rating Film : 8.7
```

```
ID : 1765
```

```
        Judul Film : Skyfall
        Rating Film : 7.8
```

```
ID : 1567
```

```
        Judul Film : The Dark Knight Rises
        Rating Film : 8.4
```

```
ID : 1234
```

```
        Judul Film : Death on The Nile
        Rating Film : 6.6
```

```
ID : 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

```
=====
Pilih menu: 9
```

```
Cetak Data
```

```
ID : 1222
```

```
        Judul Film : Spider-Man: No Way Home
        Rating Film : 8.7
```

```
ID : 1567
```

```
        Judul Film : The Dark Knight Rises
        Rating Film : 8.4
```

```
ID : 1765
```

```
        Judul Film : Skyfall
        Rating Film : 7.8
```

```
ID : 1346
```

```
        Judul Film : Uncharted
        Rating Film : 6.7
```

```
ID : 1234
```

```
        Judul Film : Death on The Nile
        Rating Film : 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

```
=====
Pilih menu: 8
```

```
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
```

```
=====
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

```
=====
Pilih menu: 10
```

```
PS E:\KULIAH 2\Pratikum Algoritma dan Struktur Data\PrakASD_1F_13>
```