

Laporan Praktikum Alogaritma dan Struktur Data

Jobsheet 12 : Double Linked List

Alogaritma dan Struktur Data

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



Eka Putri Natalya Kabelen

2341760107

SIB 1E

PROGRAM STUDI SISTEM INFORMASI BISNIS

JURUSAN TEKNOLOGI INFORMASI

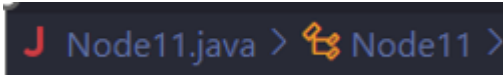
POLITEKNIK NEGERI MALANG

2024

12.2 Kegiatan Praktikum

1 12.2.1 Percobaan 1

1. Buat paket baru dengan nama doublelinkedlists
2. Buat class di dalam paket tersebut dengan nama Node



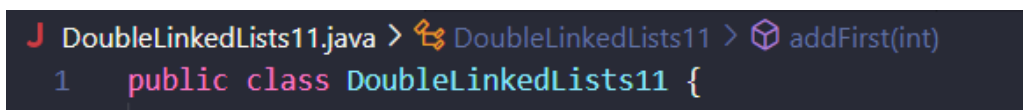
```
J Node11.java > Node11 >
```

3. Di dalam class tersebut, deklarasikan atribut sesuai dengan diagram class di atas. dan Selanjutnya tambahkan konstruktor default pada class Node sesuai diagram di atas.



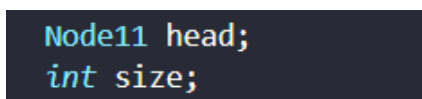
```
J Node11.java > Node11 > data
1  /**
2   * Node11
3   */
4  public class Node11 {
5
6      int data;
7      Node11 prev, next;
8
9      Node11(Node11 prev, int data, Node11 next) {
10         this.prev = prev;
11         this.data = data;
12         this.next = next;
13     }
14 }
```

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



```
J DoubleLinkedLists11.java > DoubleLinkedLists11 > addFirst(int)
1  public class DoubleLinkedLists11 {
```

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



```
Node11 head;
int size;
```

6. Selajuntnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut.

```

6      public DoubleLinkedLists11() {
7          head = null;
8          size = 0;
9      }

```

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

```

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

```

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

```

15     public void addFirst (int item) {
16         if (isEmpty()) {
17             head = new Node11(prev:null, item, next:null);
18         }else {
19             Node11 newNode11 = new Node11(prev:null, item, head);
20             head.prev = newNode11;
21             head = newNode11;
22         }
23         size++;
24     }

```

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

```

25     public void addLast (int item) {
26         if (isEmpty()) {
27             addFirst(item);
28         }else {
29             Node11 current = head;
30             while (current.next != null) {
31                 current = current.next;
32             }
33             Node11 newNode11 = new Node11(current, item, next:null);
34             current.next = newNode11;
35             size++;
36         }
37     }

```

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

```

38 public void add (int item, int index) throws Exception {
39     if (isEmpty()) {
40         addFirst(item);
41     } else if (index < 0 || index > size) {
42         throw new Exception (message:"Nilai indeks di luar batas");
43     } else {
44         Node11 current = head;
45         int i = 0;
46         while (i < index) {
47             current = current.next;
48             i++;
49         }
50         if (current.prev == null) {
51             Node11 newNode11 = new Node11(prev:null, item, current);
52             current.prev = newNode11;
53             head = newNode11;
54         } else {
55             Node11 newNode11 = new Node11(current.prev, item, current);
56             newNode11.prev = current.prev;
57             newNode11.next = current;
58             current.prev.next = newNode11;
59             current.prev = newNode11;
60         }
61     }
62     size++;
63 }

```

11. Jumlah data yang ada di dalam linked lists akan diperbarui secara otomatis, sehingga dapat dibuat method `size()` untuk mendapatkan nilai dari `size` Dan Selanjutnya dibuat method `clear()` untuk menghapus semua isi linked lists, sehingga linked lists dalam kondisi kosong.

```

64 public int size() {
65     return size;
66 }
67 public void clear() {
68     head = null;
69     size = 0;
70 }

```

12. 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.

```

71 public void print() {
72     if (!isEmpty()) {
73         Node11 tmp = head;
74         while (tmp != null) {
75             System.out.print(tmp.data + "\t");
76             tmp = tmp.next;
77         }
78         System.out.println(x:"\nBerhasil diisi");
79     } else {
80         System.out.println(x:"Linked Lists Kosong");
81     }
82 }
83 }

```

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

```
J DoubleLinkedLists11Main.java > DoubleLinkedLists11Main
1 public class DoubleLinkedLists11Main {
2
```

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

```
run | Debug
3 public static void main(String[] args) {
4     DoubleLinkedLists11 dll = new DoubleLinkedLists11();
5     dll.print();
6     System.out.println("Size: " + dll.size());
7     System.out.println(x:"=====");
8     dll.addFirst(item:3);
9     dll.addLast(item:4);
10    dll.addFirst(item:7);
11    dll.print();
12    System.out.println("Size: " + dll.size());
13    System.out.println(x:"=====");
14    try {
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    } catch (Exception e) {
23        e.printStackTrace();
24    }
25    System.out.println(x:"=====");
26    System.out.println(x:"-----");
27    System.out.println(x:"BUILD SUCCESS");
28    System.out.println(x:"-----");
29    }
30 }
```

15. 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
=====
Linked Lists Kosong
Size: 0
=====
-----
BUILD SUCCESS
-----
PS C:\Users\ASUS\Documents\SEMESTER 2\Algoritma&StrukturData\JOBSHEET 12>
```

12.2.3 Pertanyaan Percobaan

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

Jawab :

Perbedaan antara single linked list dengan double linked lists yaitu berada pada pointer atau penghubung antar node. Yang dimana pada single linked list hanya bisa next pointer dan tidak bisa Kembali, sehingga diharuskan traversal dari awal atau head untuk merubah/mengakses data tertentu. Sedangkan, double linked lists pada pointer atau penghubung antar node ada dua, yaitu next dan previous yang dimana mempermudah programmer untuk memodifikasi suatu data karena tidak perlu traversal seperti single linked list.

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

Jawab :

Atribut next dan prev tersebut berfungsi sebagai pointer atau penghubung antar node.

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 : menandakan bahwa awalnya linked lists tidak memiliki node apapun dan memberi tahu bahwa linked list kosong saat objek 'DoubleLinkedLists' pertama kali dibuat. size = 0 : menandakan bahwa ukuran (jumlah node) linked list awalnya 0 yang mengindikasikan bahwa tidak ada elemen dalam linked list. sehingga kesimpulannya konstruktor 'DoubleLinkedLists' mempersiapkan linked list kosong yang dapat digunakan untuk menyimpan dan mengelola elemen- elemen baru yang ditambahkan nanti.

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 node yang diawal atau head ditandai dengan prev yang mengarah ke null. Sehingga, pada pembuatan object dari konstruktor tersebut pada Node prev akan diberi nilai null.

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

Arti statement tersebut untuk menghubungkan head yang lama dengan Node baru yang akan ditambahkan diawal yaitu newNode. Sehingga prev dari head yang awalnya bernilai null dihubungkan ke Node 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);

Jawab :

Arti dari pengisian nilai Node prev pada pembuatan konstruktor tersebut adalah untuk menghubungkan Node newNode yang akan ditaruh paling belakang dengan Node current yang sementara berada dipaling belakang dan pengisian nilai Node next diisi null mengindikasikan bahwa Node tersebut akan berada di paling belakang sehingga ketika di next akan bernilai null. Nilai Node Current berada di paling belakang setelah dilakukannya traversal untuk mencari Node mana yang berada di paling belakang.

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);  
    newNode.prev = current.prev;  
    newNode.next = current;  
    current.prev.next = newNode;  
    current.prev = newNode;  
}
```

jelaskan maksud dari bagian yang ditandai dengan kotak kuning.

Jawab :

Pada potongan kode yang ditandai tersebut untuk sebuah case dimana Ketika ingin menginput data yang berada paling depan atau awal. Sehingga Node newNode akan diisi konstruktor Node prev = null (mengindikasikan bahwa dia Node paling depan), dan Node next = current (untuk menghubungkan Node newNode kepada node yang sementara berada di paling depan). Lalu, Node current akan dihubungkan ke Node newNode dengan mengganti nilai Node prev pada Node current dengan newNode, sehingga kedua Node tersebut akhirnya terhubung 2 arah baik dengan menggunakan next ataupun prev. Terakhir, karena berada di paling depan maka head akan digantikan ke Node newNode.

12.3 Kegiatan Praktikum 2

12.3.1 Tahapan Percobaan

1. Buatlah method `removeFirst()` di dalam class `DoubleLinkedLists`.

```
84 public void removeFirst() throws Exception {  
85     if (isEmpty()) {  
86         throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus!");  
87     } else if (size == 1) {  
88         removeLast();  
89     } else {  
90         head = head.next;  
91         head.prev = null;  
92         size--;  
93     }  
94 }
```

2. Tambahkan method `removeLast()` di dalam class `DoubleLinkedLists`.

```
94 }  
95 public void removeLast() throws Exception {  
96     if (isEmpty()) {  
97         throw new Exception(message:"Linked List masih kosong, tidak dapat dihapus!");  
98     } else if (head.next == null) {  
99         head = null;  
100         size--;  
101         return;  
102     }  
103     Node11 current = head;  
104     while (current.next.next != null) {  
105         current = current.next;  
106     }  
107     current.next = null;  
108     size--;  
109 }
```

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

```
111 public void remove(int index) throws Exception {  
112     if (isEmpty() || index >= size) {  
113         throw new Exception(message:"Nilai indeks di luar batas");  
114     } else if (index == 0) {  
115         removeFirst();  
116     } else {  
117         Node11 current = head;  
118         int i = 0;  
119         while (i < index) {  
120             current = current.next;  
121             i++;  
122         }  
123         if (current.next == null) {  
124             current.prev.next = null;  
125         } else if (current.prev == null) {  
126             current = current.next;  
127             current.prev = null;  
128             head = current;  
129         } else {  
130             current.prev.next = current.next;  
131             current.next.prev = current.prev;  
132         }  
133         size--;  
134     }  
135 }  
136 }
```


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

```
21      System.out.println("Size: " + dll.size());
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(x:"=====");
29      dll.removeFirst();
30      dll.print();
31      System.out.println("Size : "+dll.size());
32      System.out.println(x:"=====");
33      dll.removeLast();
34      dll.print();
35      System.out.println("Size : "+dll.size());
36      System.out.println(x:"=====");
37      dll.remove(index:1);
38      dll.print();
39      System.out.println("Size : "+dll.size());
```

5. 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
=====
-----
BUILD SUCCESS
-----
```

12.3.3 Pertanyaan Percobaan

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

`head = head.next;`

`head.prev = null;`

Jawab :

Maksud dari potongan kode tersebut adalah:

- `head = head.next;` => artinya adalah mengganti nilai `head` ke `head.next`
- `head.prev = null;` => artinya adalah merubah nilai `head.prev` yang tadinya masih ada nilai `Node` nya menjadi `null` untuk memutus pointer ke `Node` yang paling awal sebelumnya.

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

Jawab :

Untuk mendeteksi posisi data pada bagian akhir ditandai dengan potongan kode berikut

```
Node11 current = head;
while (current.next.next != null) {
    current = current.next;
}
current.next = null;
size--;
```

Pada potongan kode diatas menjelaskan akan dilakukan traversal dimulai dari `head` hingga `Node current.next.next` bernilai `null`. Sehingga bisa dilakukan `remove` atau penghapusan pada node terakhir ditandai dengan '`current.next = null`'.

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 :

Potongan program tersebut tidak cocok karena potongan program tersebut hanya menghapus node kedua dalam linked list, bukan node yang sebenarnya ingin dihapus. berikut penjelasannya:

- 1) `Node tmp = head.next;` => baris ini menginisialisasi `tmp` dengan node kedua dalam linked list, ditandai dengan adanya '`head.next`'
- 2) `head.next = tmp.next` => baris ini memperbarui nilai `head.next` untuk melompati `tmp`. ini menghapus koneksi dari `head` ke `tmp` dan langsung menghubungkan `head` ke node setelah `tmp`.

3) `tmp.next.prev = head` => baris ini memperbarui nilai `prev` dari node setelah `tmp` (node ke-3) untuk mentarget `head`. namun, jika `tmp` adalah node kedua, maka `tmp.next` akan menjadi `null` dan ketika mengakses `prev` dari `null` akan terjadi 'NullPointerException' atau data tidak ada.

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

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

Jawab :

Fungsi kode program tersebut adalah untuk menghapus node yang akan kita target. menggambarkan fungsi dari potongan program tersebut adalah untuk menghubungkan node sebelum dan sesudah dari Node yang telah kita target untuk dihapus. Sehingga setelah node tersebut dihapus maka Node sebelum dan sesudah nya akan tetap masih terhubung. Seperti ilustrasi berikut.

12.4 Kegiatan Praktikum 3

12.4.1 Tahapan Percobaan

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

```
136 public int getFirst() throws Exception {
137     if (isEmpty()) {
138         throw new Exception(message:"Linked List Kosong");
139     }
140     return head.data;
141 }
```

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

```
142 public int getLast() throws Exception {
143     if (isEmpty()) {
144         throw new Exception(message:"Linked List kosong");
145     }
146     Node11 tmp = head;
147     while (tmp.next != null) {
148         tmp = tmp.next;
149     }
150     return tmp.data;
151 }
```

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

```
2      public int get(int index) throws Exception {
3          if (isEmpty() || index >= size) {
4              throw new Exception(message:"Nilai indeks di luar batas.");
5          }
6          Node11 tmp = head;
7          for (int i = 0; i < index; i++) {
8              tmp = tmp.next;
9          }
10         return tmp.data;
11     }
12 }
13
14
```

4. Pada main class tambahkan potongan program berikut dan amati hasilnya!

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

5. Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.

```
=====
Linked Lists Kosong
Size: 0
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
40
Berhasil diisi
Size: 1
=====
7      3      40      4
Berhasil diisi
Size: 4
=====
7      40      3      40      4
Berhasil diisi
Size: 5
=====
Data awal pada Linked Lists adalah: 7
Data akhir pada Linked Lists adalah: 4
Data indeks ke-1 pada Linked Lists adalah: 40
=====
```

12.4.2 Pertanyaan Percobaan

1. Jelaskan method size() pada class DoubleLinkedLists!

Jawab :

Method size digunakan mengembalikan nilai dari atribut size untuk memberitahukan berapa jumlah data pada linked lists.

2. Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke1!

Jawab :

Pertama adalah mengatur pada konstruktor double linked lists itu sendiri di set menjadi bernilai size = 1;

Kedua, setiap memulai proses traversal langsung mengarahkan nilai indeks nya adalah satu. Sehingga proses perhitungan indeks akan dimulai dari satu.

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

Jawab :

Perbedaan karakteristik nya adalah pada double linked lists diperlukan parameter Node Next dan Prev. Sedangkan pada single linked list hanya memerlukan parameter 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 :

Perbedaan kedua logika tersebut berdasarkan apa yang menjadi perhitungannya. Pada point (a) kondisinya adalah menghitung apakah size = 0. Jika iya maka akan menghasilkan nilai true, jika tidak maka akan menghasilkan nilai false. Sedangkan pada point (b) kondisinya akan melihat apakah nilai head == null. Jika iya maka akan menghasilkan nilai true. Pada point (b) tidak terlihat secara tertulis seperti pada point (a) untuk memberitahukan kondisi apa yang menghasilkan nilai true ataupun false.

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).

Code Program

```
J NodeVaksin.java > NodeVaksin
1 public class NodeVaksin {
2
3     int nomor;
4     String nama;
5
6     NodeVaksin prev, next;
7
8     NodeVaksin(NodeVaksin prev, int nomor, String nama, NodeVaksin next){
9         this.prev = prev;
10        this.nomor = nomor;
11        this.nama = nama;
12        this.next = next;
13    }
14 }
```

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

```

1  public class doublelinkedlistVaksin {
2
3      NodeVaksin head;
4      int size;
5
6      public doublelinkedlistVaksin(){
7          head = null;
8          size = 1;
9      }
10     public boolean isEmpty(){
11         return head == null;
12     }
13     public void addLast(int nomor, String nama){
14         if (isEmpty()){
15             head = new NodeVaksin(null, nomor, nama, null);
16         }else{
17             NodeVaksin current = head;
18             while (current.next != null){
19                 current = current.next;
20             }
21             NodeVaksin newNodeVaksin = new NodeVaksin(current, nomor, nama, null);
22             current.next = newNodeVaksin;
23             size++;
24         }
25     }
26     public void removeFirst() throws Exception{
27         if(isEmpty()){
28             throw new Exception("Linked List Masih kosong, tidak dapat dihapus!");
29         }else if (size == 1){
30             removeLast();
31         }else{
32             System.out.println(head.nama+"Sudah DiNodeVaksin");
33             head = head.next;
34             head.prev = null;
35             size--;
36         }
37     }
38     public void removeLast() throws Exception {
39         if(isEmpty()){
40             throw new Exception("Linked list masih kosong, tidak dapat dihapus!");
41         }else if (head.next == null){
42             System.out.println(head.nama+"Sudah DiNodeVaksin");
43             head = null;
44             size--;
45             return;
46         }
47         NodeVaksin current = head;
48         while (current.next.next != null){
49             current = current.next;
50         }
51         current.next = null;
52         size--;
53     }
54     public void print(){
55         System.out.println("+++++++");
56         System.out.println(" DAFTAR PENGANTRI NodeVaksin ");
57         System.out.println("+++++++");
58         System.out.println("| No\t| Nama+\t\t|");
59         if(!isEmpty()){
60             NodeVaksin tmp = head;
61             while (tmp != null){
62                 System.out.print("| "+tmp.nomor+"\t| "+tmp.nama+" "+"+\t|\n");
63                 tmp = tmp.next;
64             }
65             System.out.println("\nSisa Antrian "+size);
66         }else{
67             System.out.println("Linked List Kosong");
68         }
69     }
70 }
71
72
73

```

Hasil Running Kode Program

```
+++++
PENGANTRI VAKSIN POLINEMA
+++++
```

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

```
-----
1
Nomor Antrian   : 123
Nama Penerima  : Margareta
```

```
+++++
PENGANTRI VAKSIN POLINEMA
+++++
```

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

```
-----
1
Nomor Antrian   : 124
Nama Penerima   : Putri
```

```
+++++
PENGANTRI VAKSIN POLINEMA
+++++
```

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

```
-----
1
Nomor Antrian   : 125
Nama Penerima   : Annisa
```

```
+++++
PENGANTRI VAKSIN POLINEMA
+++++
```

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

```
-----
1
Nomor Antrian   : 126
Nama Penerima   : Jua
```

```
+++++
PENGANTRI VAKSIN POLINEMA
+++++
```

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

```
-----
1
Nomor Antrian   : 127
Nama Penerima   : Ney
```

```
+++++
PENGANTRI VAKSIN POLINEMA
+++++
```

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

```
-----
1
Nomor Antrian   : 129
Nama Penerima   : Rahmat
```

1
Nomor Antrian : 130
Nama Penerima : Aftril

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

3
+++++
DAFTAR PENGANTRI NodeVaksin
+++++
No	Nama
123	Margareta
124	Putri
125	Annisa
126	Jua
127	Ney
129	Rahmat
130	Aftril

Sisa Antrian 7
+++++
PENGANTRI VAKSIN POLINEMA
+++++
1. Tambah Data Penerima Vaksin
2. Hapus Data Pengantri Vaksin
3. Daftar Penerima Vaksin
4. Keluar

2
MargaretaSudah DiNodeVaksin

+++++
DAFTAR PENGANTRI NodeVaksin
+++++
No	Nama
124	Putri
125	Annisa
126	Jua
127	Ney
129	Rahmat
130	Aftril

Sisa Antrian 6

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

4
PS C:\Users\ASUS\Documents\SEMESTER

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.

Code Program

```
J NodeFilm.java > ...
1  public class NodeFilm {
2
3      int id;
4      String judulFilm;
5      double rating;
6
7      NodeFilm prev, next;
8
9      NodeFilm(NodeFilm prev, int id, String judulFilm, double rating, NodeFilm next){
10         this.prev = prev;
11         this.id = id;
12         this.judulFilm = judulFilm;
13         this.rating = rating;
14         this.next = next;
15     }
16 }
```

```

1  /**
2   * doubleLinkedListFilm
3   */
4  public class doubleLinkedListFilm {
5
6      NodeFilm head;
7      int size;
8
9      public doubleLinkedListFilm(){
10         head = null;
11         size = 0;
12     }
13     public boolean isEmpty(){
14         return head == null;
15     }
16     public void addFirst(int id, String judulFilm, double rating){
17         if (isEmpty()){
18             head = new NodeFilm(null, id, judulFilm, rating, null);
19         }
20         else{
21             NodeFilm newNodeFilm = new NodeFilm(null, id, judulFilm, rating, head);
22             head.prev = newNodeFilm;
23             head = newNodeFilm;
24         }
25         size++;
26     }
27     public void addLast(int id, String judulFilm, double rating){
28         if (isEmpty()){
29             addFirst(id, judulFilm, rating);
30         }
31         else{
32             NodeFilm current = head;
33             while (current.next != null){
34                 current = current.next;
35             }
36             NodeFilm newNodeFilm = new NodeFilm(current, id, judulFilm, rating, null);
37             current.next = newNodeFilm;
38             size++;
39         }
40     }
41     public void add(int id, String judulFilm, double rating, int index) throws Exception {
42         if (isEmpty()){
43             addFirst(id, judulFilm, rating);
44         }
45         else if (index < 0 || index > size){
46             throw new Exception("Nilai indeks di luar batas");
47         }
48         else{
49             NodeFilm current = head;
50             int i = 0;
51             while(i < index){
52                 current = current.next;
53                 i++;
54             }
55             if(current.prev == null){
56                 NodeFilm newNodeFilm = new NodeFilm(null, id, judulFilm, rating, current);
57                 current.prev = newNodeFilm;
58                 head = newNodeFilm;
59             }
60             else{
61                 NodeFilm newNodeFilm = new NodeFilm(current.prev, id, judulFilm, rating, current);
62                 newNodeFilm.prev = current.prev;
63                 newNodeFilm.next = current;
64                 current.prev.next = newNodeFilm;
65                 current.prev = newNodeFilm;
66             }
67             size++;
68         }
69     }
70     public void removeFirst() throws Exception{
71         if(isEmpty()){
72             throw new Exception("Linked List Masih kosong, tidak dapat dihapus!");
73         }
74         else if (size == 1){
75             removeLast();
76         }
77         else{
78             head = head.next;
79             head.prev = null;
80             size--;
81         }
82     }
83     public void removeLast() throws Exception {
84         if(isEmpty()){
85             throw new Exception("Linked list masih kosong, tidak dapat dihapus!");
86         }
87         else if (head.next == null){
88             head = null;
89             size--;
90             return;
91         }
92         NodeFilm current = head;
93         while (current.next.next != null){
94             current = current.next;
95         }
96         current.next = null;
97         size--;
98     }
99 }

```

```

92     public void remove(int index) throws Exception {
93         if (isEmpty() || index >= size){
94             throw new Exception("Nilai indeks di luar batas");
95         }else if (index == 0){
96             removeFirst();
97         }else{
98             NodeFilm current = head;
99             int i=0;
100             while (i<index){
101                 current = current.next;
102                 i++;
103             }
104             if(current.next == null){
105                 current.prev.next = null;
106             }else if(current.prev == null){
107                 current = current.next;
108                 current.prev = null;
109                 head = current;
110             }else{
111                 current.prev.next = current.next;
112                 current.next.prev = current.prev;
113             }
114             size--;
115         }
116     }
117     public void print(){
118         System.out.println("-----");
119         System.out.println("Data Saat Ini Menjadi");
120         if(!isEmpty()){
121             NodeFilm tmp = head;
122             while (tmp != null){
123                 System.out.println("ID \t\t: "+tmp.id);
124                 System.out.println("Judul Film \t: "+tmp.judulFilm);
125                 System.out.println("Rating \t\t: "+tmp.rating);
126                 System.out.println();
127                 tmp = tmp.next;
128             }
129         }else{
130             System.out.println("Linked List Kosong");
131         }
132     }
133     public int findSeqSearch(int cari){
134         NodeFilm tmp = head;
135         int posisi = -1;
136         int index = 0;
137         while (tmp != null){
138             if(tmp.id == cari){
139                 posisi = index;
140                 break;
141             }
142             index++;
143             tmp = tmp.next;
144         }
145         return posisi;
146     }
147     public void tampilPosisi(int x, int pos){
148         if(pos != -1){
149             System.out.println("ID\t\t: "+ x +" ditemukan pada indeks "+pos);
150         }else {
151             System.out.println("ID\t\t"+ x +" tidak ditemukan");
152         }
153     }
154     public void sort(){
155         NodeFilm current = null, index = null;
156         int tempID;
157         String tempJD;
158         Double tempRT;
159         if (head == null) {
160             return;
161         } else {
162             for (current = head; current.next != null; current = current.next) {
163                 for (index = current.next; index != null; index = index.next) {
164                     if (current.rating < index.rating) {
165                         tempRT = current.rating;
166                         current.rating = index.rating;
167                         index.rating = tempRT;
168                         tempID = current.id;
169                         current.id = index.id;
170                         index.id = tempID;
171                         tempJD = current.judulFilm;
172                         current.judulFilm = index.judulFilm;
173                         index.judulFilm = tempJD;
174                     }
175                 }
176             }
177             print();
178         }
179     }
180 }
181
182

```

```

1 import java.util.Scanner;
2 public class doubleLinkedListFilmMain {
3
4     public static void menu() {
5         System.out.println("=====");
6         System.out.println(" Data Film Layar Lebar ");
7         System.out.println("=====");
8         System.out.println(" 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 Masing Film DESC 10. Keluar ");
9         System.out.println("=====");
10    }
11
12    public static void main(String[] args) throws Exception {
13
14        Scanner sc = new Scanner(System.in);
15        Scanner sd = new Scanner(System.in);
16        Scanner sh = new Scanner(System.in);
17
18        doubleLinkedListFilm dll = new doubleLinkedListFilm();
19
20        int pilih;
21        do {
22            menu();
23            pilih = sc.nextInt();
24            sc.nextLine();
25

```

```

26            switch(pilih){
27                case 1:
28                    System.out.println("Masukkan Data Film Posisi Awal");
29                    System.out.print("ID Film\t\t: ");
30                    int id = sd.nextInt();
31                    System.out.print("Judul Film\t: ");
32                    String judulFilm = sc.nextLine();
33                    System.out.print("Rating\t\t: ");
34                    double rating = sb.nextDouble();
35                    dll.addFirst(id, judulFilm, rating);
36                    sc.nextLine();
37                    break;
38
39                case 2:
40                    System.out.println("Masukkan Data Film Posisi Akhir");
41                    System.out.print("ID Film\t\t: ");
42                    int idD = sd.nextInt();
43                    System.out.print("Judul Film\t: ");
44                    String judulFilmM = sc.nextLine();
45                    System.out.print("Rating\t\t: ");
46                    double ratingG = sb.nextDouble();
47                    dll.addLast(idD, judulFilmM, ratingG);
48                    sc.nextLine();
49                    break;
50
51                case 3:
52                    System.out.println("Masukkan Data Film Posisi yang diinginkan");
53                    System.out.print("Urutan ke - ");
54                    int idx = sd.nextInt();
55                    System.out.print("ID Film\t\t: ");
56                    int idDD = sd.nextInt();
57                    System.out.print("Judul Film\t: ");
58                    String judulFilmMM = sc.nextLine();
59                    System.out.print("Rating\t\t: ");
60                    double ratingGG = sb.nextDouble();
61                    dll.add(idDD, judulFilmMM, ratingGG, idx);
62                    sc.nextLine();
63                    break;
64
65                case 4:
66                    dll.removeFirst();
67                    dll.print();
68                    break;
69
70                case 5:
71                    dll.removeLast();
72                    dll.print();
73                    break;
74
75                case 6:
76                    System.out.println("hapus Data Film Posisi yang diinginkan");
77                    System.out.print("Urutan ke - ");
78                    int index = sd.nextInt();
79                    dll.remove(index);
80                    dll.print();
81                    break;
82
83                case 7:
84                    dll.print();
85                    break;
86
87                case 8:
88                    System.out.println("Cari ID Film Yang ingin dicari");
89                    System.out.print("Masukkan ID\t: ");
90                    int cari = sd.nextInt();
91                    int idF = dll.findSeqSearch(cari);
92                    dll.tampilPosisi(cari, idF);
93                    break;
94
95                case 9:
96                    System.out.println("Data Akan diurut secara DESC");
97                    dll.sort();
98                    break;
99
100                case 10:
101                    System.exit(0);
102                    break;
103            }
104        } while (pilih == 1 || pilih == 2 || pilih == 3 || pilih == 4 || pilih == 5 || pilih == 6 || pilih == 7 || pilih == 8 || pilih == 9 || pilih == 10);
105    }
106 }

```

Hasil Run Code Program

```
+++++
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 : nOway 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

-----
7
-----
Data Saat Ini Menjadi
ID      : 1222
Judul Film : spider man
Rating  : 8.7

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

-----
```

```
2
Masukkan Data Film Posisi Akhir
ID Film      : 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

-----
8
Cari ID Film Yang ingin dicari
Masukkan ID   : 1222
ID            : 1222 ditemukan pada indeks 0
```