**LAPORAN PRAKTIKUM**

**Politeknik Negeri Malang**

**Praktikan**

**ALGORITMA DAN STRUKTUR DATA**

****

2022

2141720183

RIDWAN CAESAR RIZQI KARISMA BIWARNI

TI 1C

Daftar Isi

[Percobaan Praktikum 1 4](#_Toc103849900)

[Langkah 1 4](#_Toc103849901)

[Langkah 2 4](#_Toc103849902)

[Langkah 3 5](#_Toc103849903)

[Langkah 4 5](#_Toc103849904)

[Langkah 5 5](#_Toc103849905)

[Langkah 6 5](#_Toc103849906)

[Langkah 7 6](#_Toc103849907)

[Langkah 8 6](#_Toc103849908)

[Langkah 9 6](#_Toc103849909)

[Langkah 10 6](#_Toc103849910)

[Langkah 11 7](#_Toc103849911)

[Langkah 12 7](#_Toc103849912)

[Langkah 13 7](#_Toc103849913)

[Langkah 14 8](#_Toc103849914)

[Langkah 15 8](#_Toc103849915)

[Langkah 16 8](#_Toc103849916)

[Langkah 17 9](#_Toc103849917)

[Verifikasi Hasil Percobaan 9](#_Toc103849918)

[Pertanyaan Percobaan 9](#_Toc103849919)

[Jawaban Percobaan 10](#_Toc103849920)

[Percobaan Praktikum 2 11](#_Toc103849921)

[Langkah 1 11](#_Toc103849922)

[Langkah 2 11](#_Toc103849923)

[Langkah 3 12](#_Toc103849924)

[Langkah 4 12](#_Toc103849925)

[Verifikasi Hasil Percobaan 13](#_Toc103849926)

[Pertanyaan Percobaan 13](#_Toc103849927)

[Jawaban Percobaan 14](#_Toc103849928)

[Percobaan Praktikum 3 15](#_Toc103849929)

[Langkah 1 15](#_Toc103849930)

[Langkah 2 15](#_Toc103849931)

[Langkah 3 16](#_Toc103849932)

[Langkah 4 16](#_Toc103849933)

[Verifikasi Hasil Percobaan 16](#_Toc103849934)

[Pertanyaan Percobaan 17](#_Toc103849935)

[Jawaban Percobaan 17](#_Toc103849936)

[Tugas Praktikum 18](#_Toc103849937)

[Soal Nomor 1 18](#_Toc103849938)

[Jawaban Nomor 1 19](#_Toc103849939)

[Source code Class Soal1DataAntrian 19](#_Toc103849940)

[Source code Class Soal1DLLMainRidwan 19](#_Toc103849941)

[Source code Class Soal1DLLRidwan 20](#_Toc103849942)

[Source code Class Soal1NodeRidwan 21](#_Toc103849943)

[Output Menu Awal 21](#_Toc103849944)

[Output Penambahan Data 22](#_Toc103849945)

[Output Cetak Data 22](#_Toc103849946)

[Output Hapus Data 23](#_Toc103849947)

[Output Keluar 23](#_Toc103849948)

[Soal Nomor 2 24](#_Toc103849949)

[Jawaban Nomor 2 25](#_Toc103849950)

[Source code Class Soal2DataFilm 25](#_Toc103849951)

[Source code Class Soal2DLLMainRidwan 25](#_Toc103849952)

[Source code Class Soal2DLLRidwan 27](#_Toc103849953)

[Source code Class Soal2NodeRidwan 30](#_Toc103849954)

[Output Menu Awal 31](#_Toc103849955)

[Output Tambah Data Awal 31](#_Toc103849956)

[Output Tambah Data Akhir 32](#_Toc103849957)

[Output Tambah Data Index Tertentu 33](#_Toc103849958)

[Output Hapus Data Pertama 33](#_Toc103849959)

[Output Hapus Data Terakhir 34](#_Toc103849960)

[Output Hapus Data Tertentu 34](#_Toc103849961)

[Output Cetak 34](#_Toc103849962)

[Output Cari ID Film 36](#_Toc103849963)

[Output Urut Data Rating-DESC 37](#_Toc103849964)

[Output Keluar 37](#_Toc103849965)

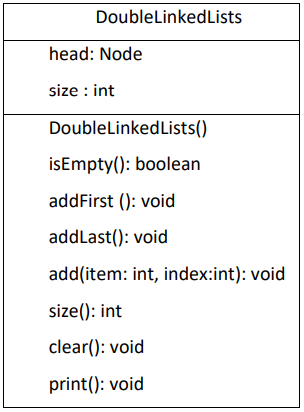
# Percobaan Praktikum 1

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

## Langkah 1

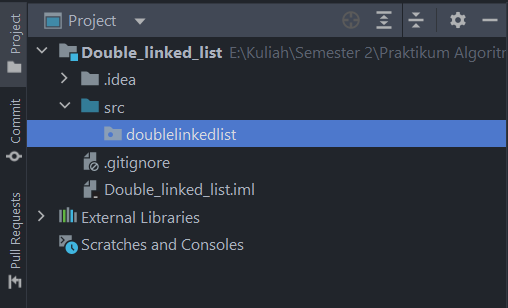
Perhatikan diagram class Node dan class DoublelinkedLists di bawah ini! Diagram class ini yang selanjutnya akan dibuat sebagai acuan dalam membuat kode program DoubleLinkedLists.





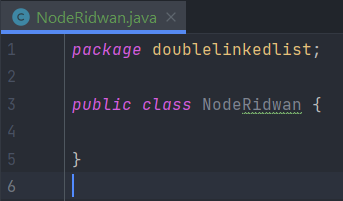
## Langkah 2

Buat paket baru dengan nama doublelinkedlists



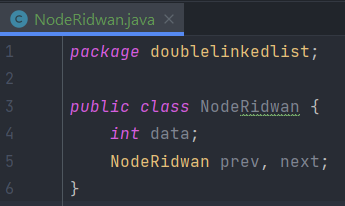
## Langkah 3

Buat class di dalam paket tersebut dengan nama Node



## Langkah 4

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



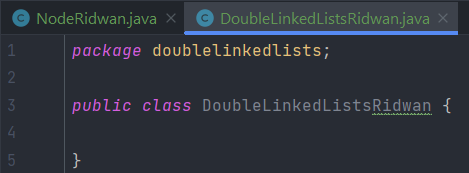
## Langkah 5

Selanjutnya tambahkan konstruktor default pada class Node sesuai diagram di atas



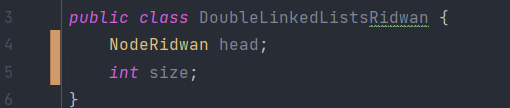
## Langkah 6

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



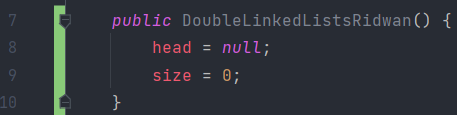
## Langkah 7

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



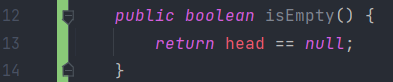
## Langkah 8

Selanjutnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut



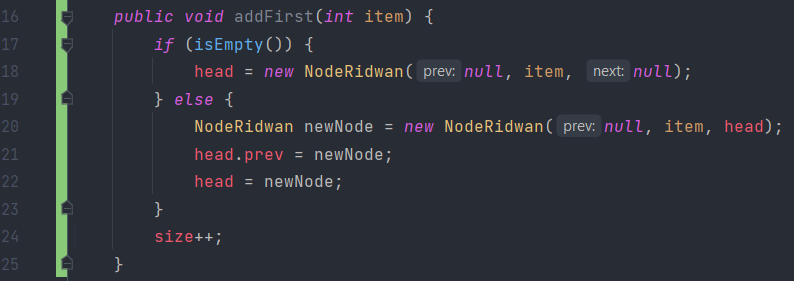
## Langkah 9

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



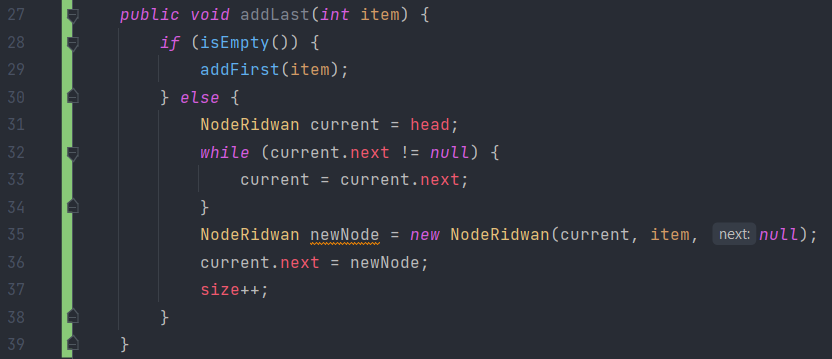
## Langkah 10

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



## Langkah 11

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



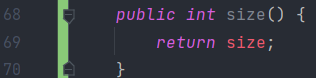
## Langkah 12

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



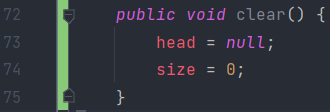
## Langkah 13

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



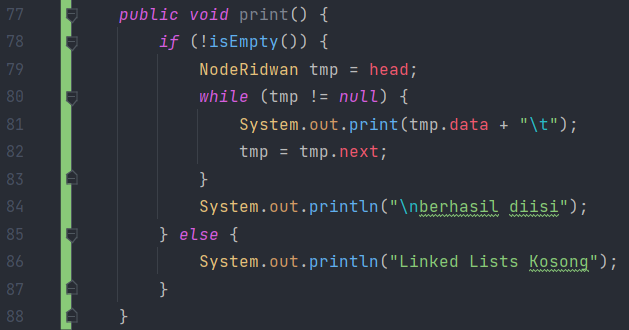
## Langkah 14

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



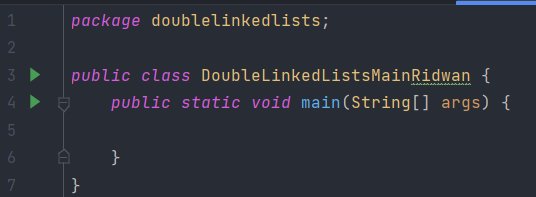
## Langkah 15

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



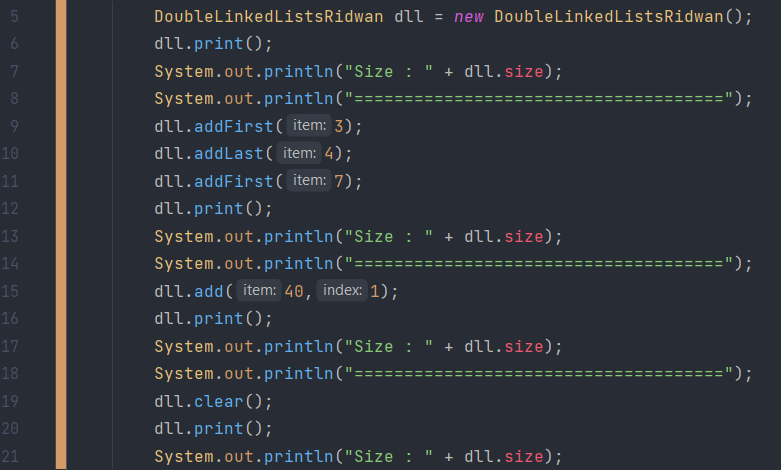
## Langkah 16

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



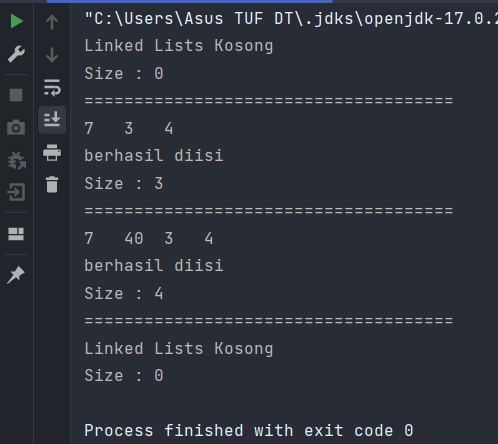
## Langkah 17

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



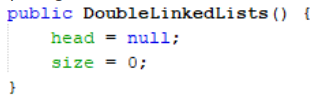
## Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini.



### Pertanyaan Percobaan

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



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



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

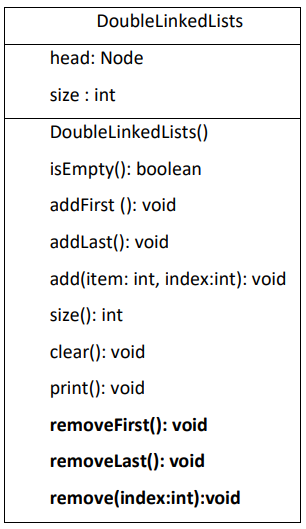


### Jawaban Percobaan

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

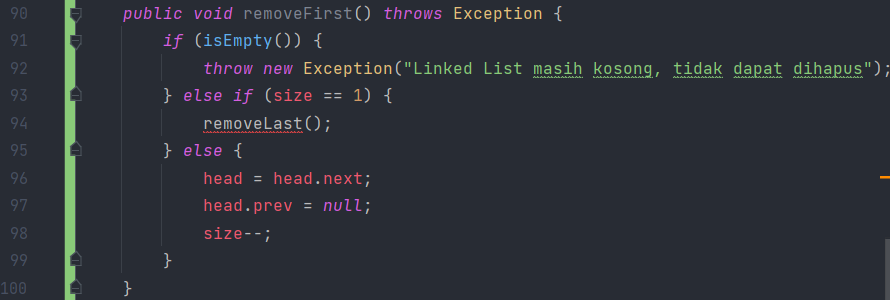
# Percobaan Praktikum 2

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



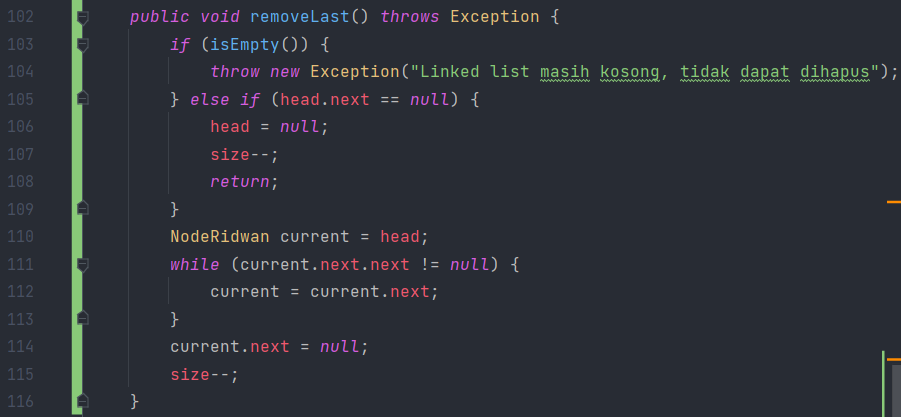
## Langkah 1

Buatlah method removeFirst() di dalam class DoubleLinkedLists



## Langkah 2

Tambahkan method removeLast() di dalam class DoubleLinkedLists.



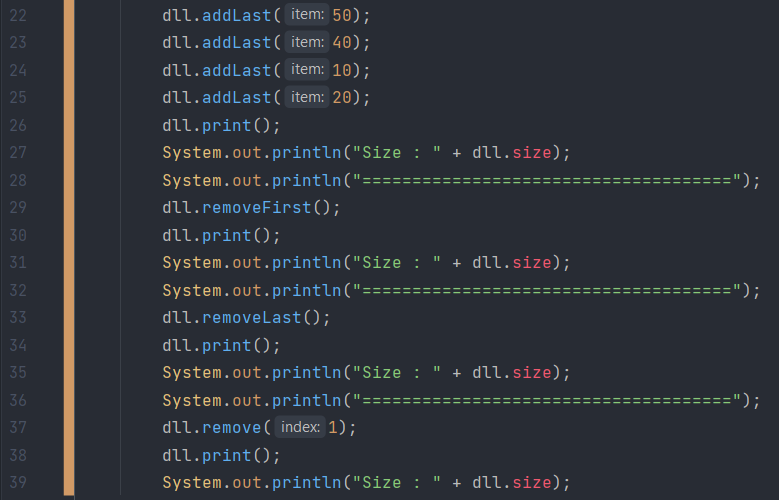
## Langkah 3

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



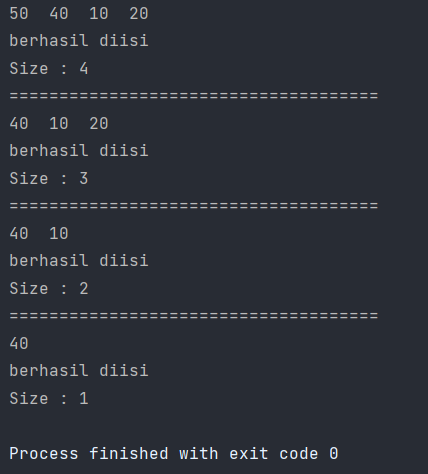
## Langkah 4

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



## Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini

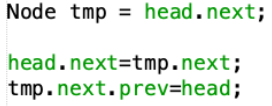


### Pertanyaan Percobaan

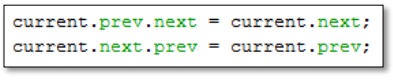
1. Apakah maksud statement berikut pada method removeFirst()?



1. Bagaimana cara mendeteksi posisi data ada pada bagian akhir pada method removeLast()?
2. Jelaskan alasan potongan kode program di bawah ini tidak cocok untuk perintah remove!

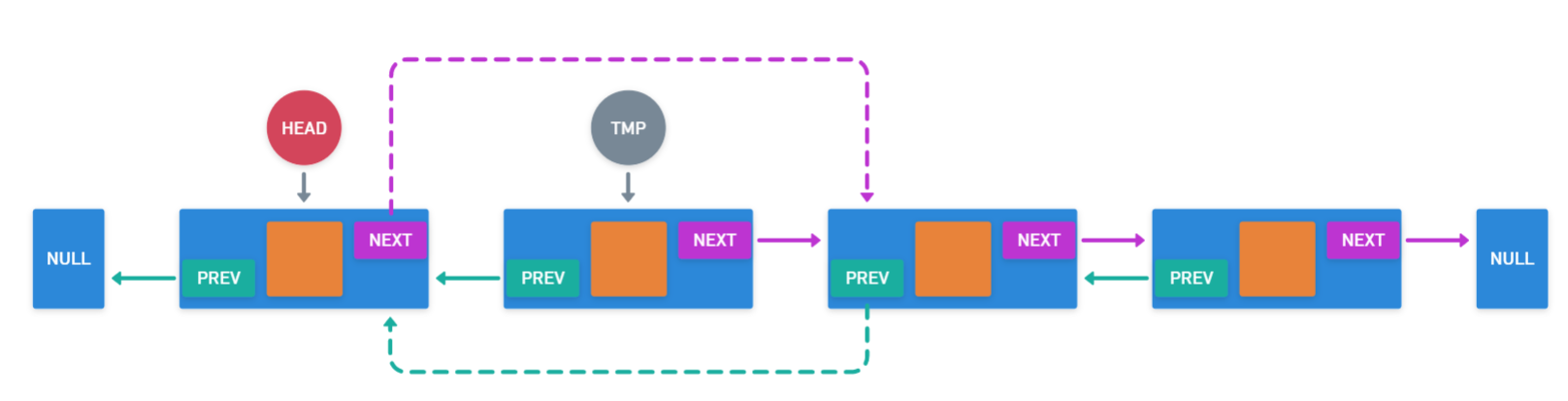


1. Jelaskan fungsi kode program berikut ini pada fungsi remove!

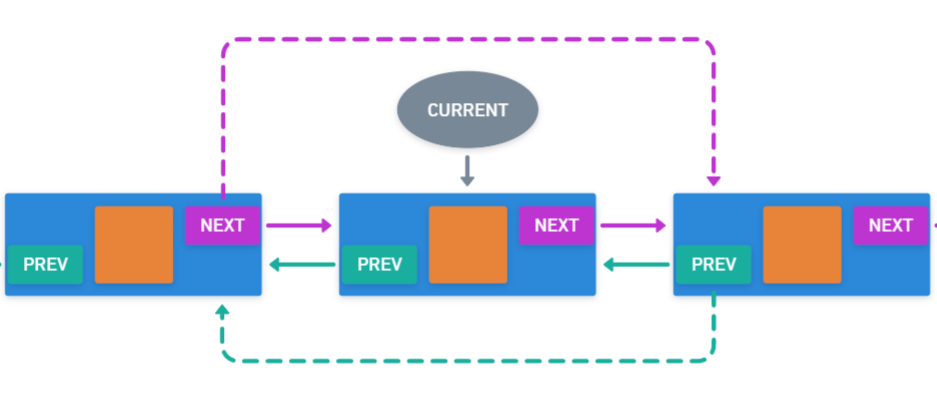


### Jawaban Percobaan

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

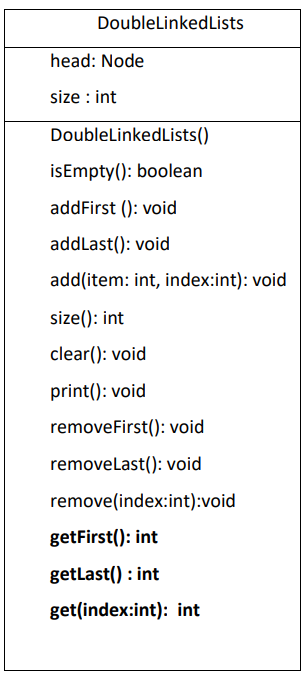


1. Fungsi dari kode program tersebut adalah pointer next yang dimiliki oleh node sebelum current akan menunjuk ke node setelah current dan pointer prev yang dimiliki oleh node setelah current akan menunjuk ke node sebelum current.



# Percobaan Praktikum 3

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



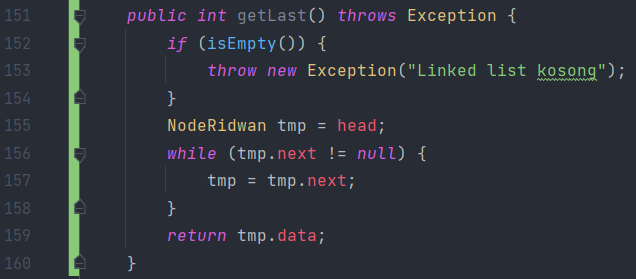
## Langkah 1

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



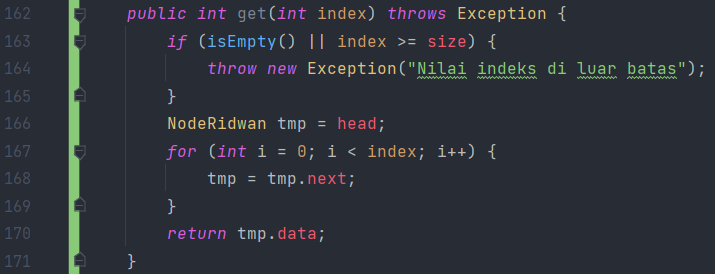
## Langkah 2

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



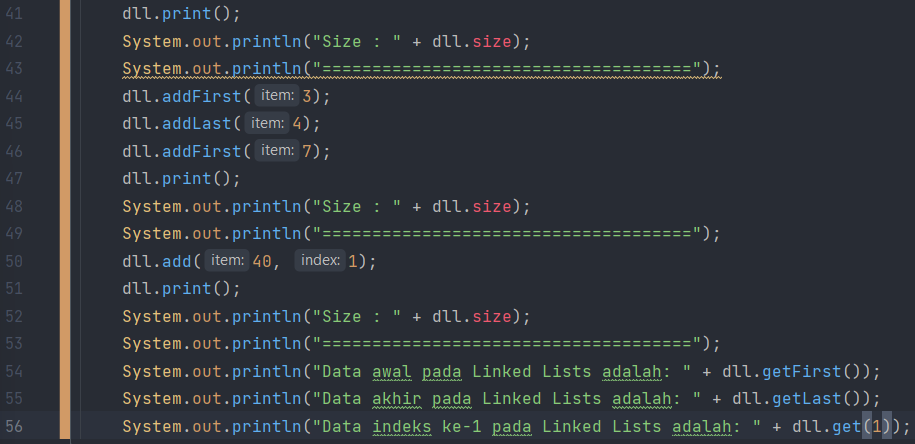
## Langkah 3

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



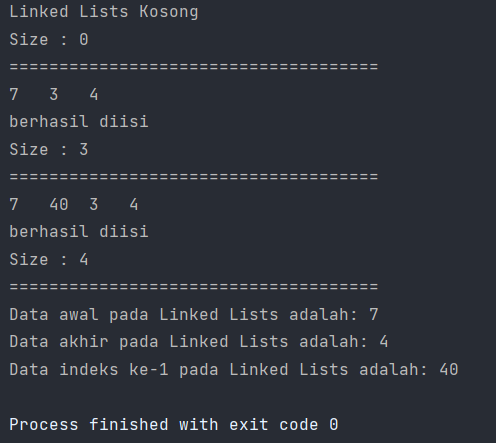
## Langkah 4

Pada main class tambahkan potongan program berikut dan amati hasilnya!



## Verifikasi Hasil Percobaan

Verifikasi hasil kompilasi kode program Anda dengan gambar berikut ini



### Pertanyaan Percobaan

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



### Jawaban Percobaan

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

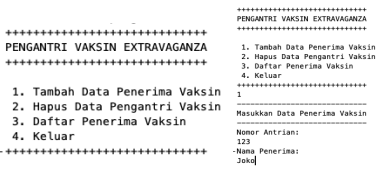
# Tugas Praktikum

## Soal Nomor 1

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

**Ilustrasi Program**

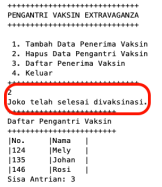
Menu Awal dan Penambahan Data



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

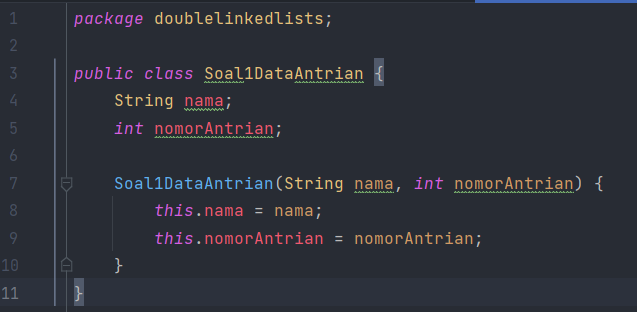


Hapus Data **(Komponen di area merah harus ada)**

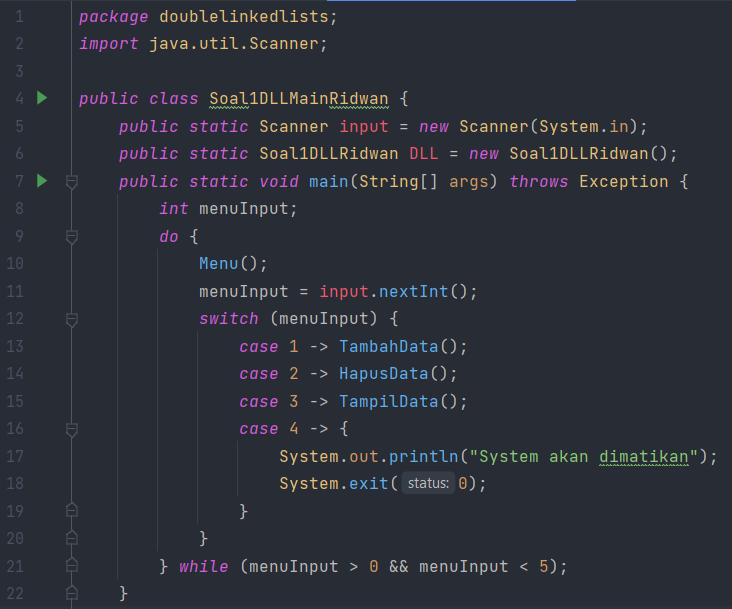


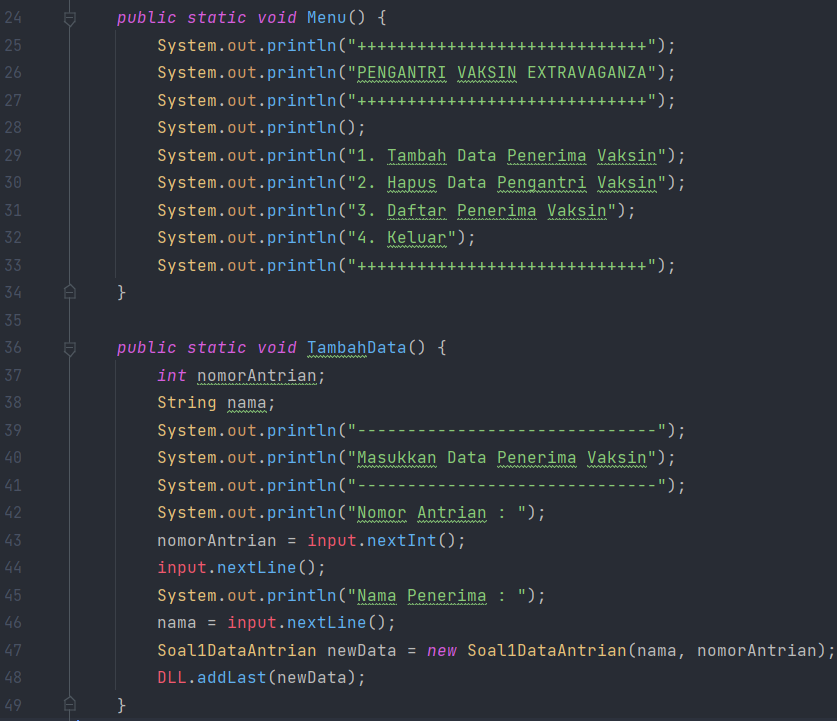
## Jawaban Nomor 1

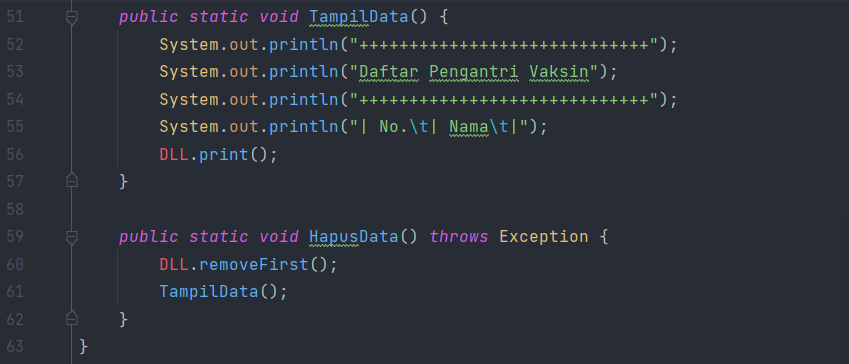
### Source code Class Soal1DataAntrian



### Source code Class Soal1DLLMainRidwan

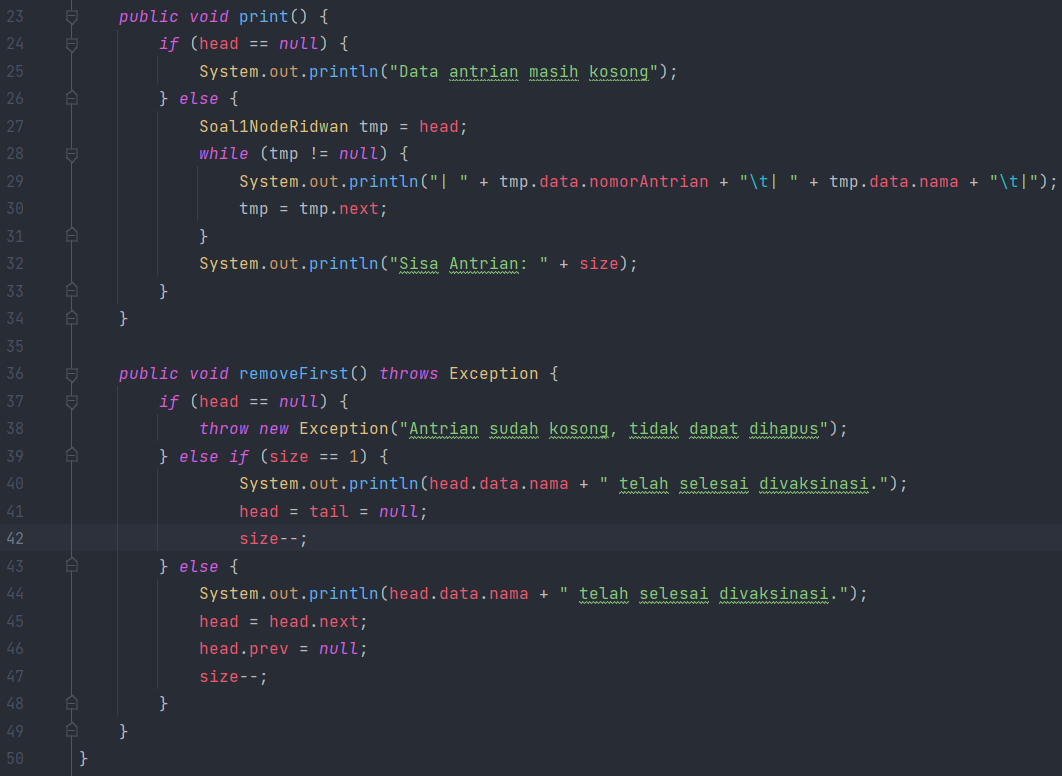




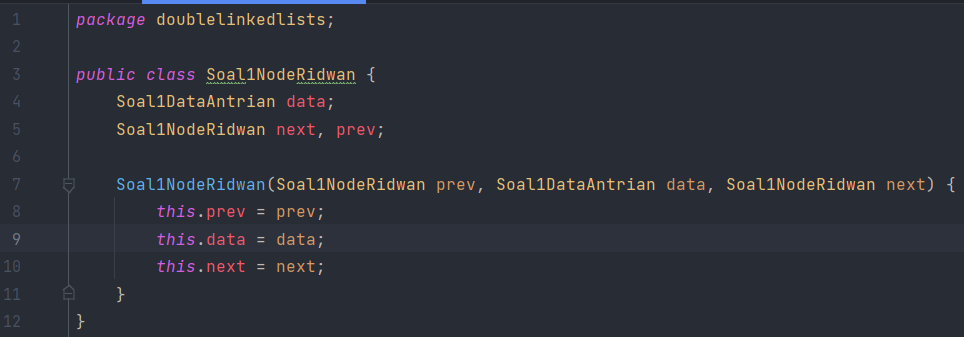


### Source code Class Soal1DLLRidwan

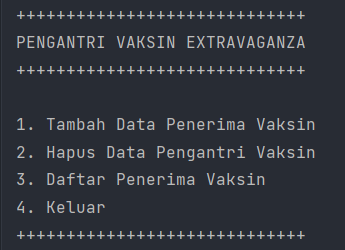




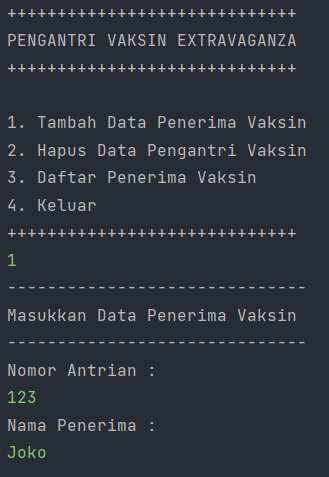
### Source code Class Soal1NodeRidwan



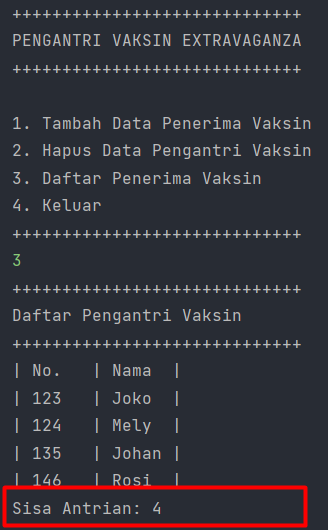
### Output Menu Awal



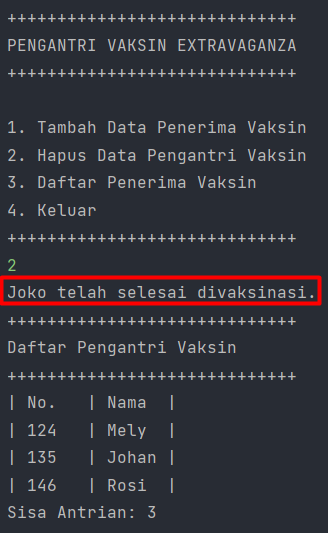
### Output Penambahan Data



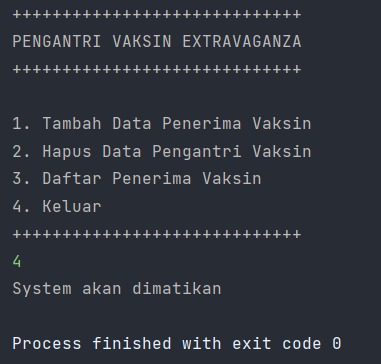
### Output Cetak Data



### Output Hapus Data



### Output Keluar

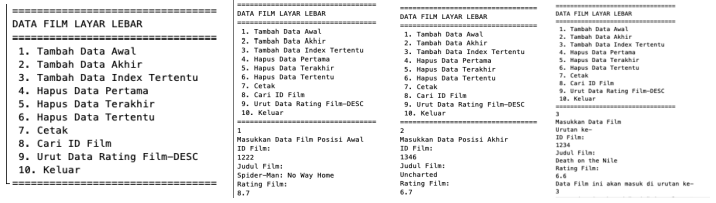


## Soal Nomor 2

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

**Contoh Ilustrasi Program**

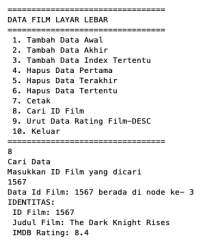
Menu Awal dan Penambahan Data



Cetak Data

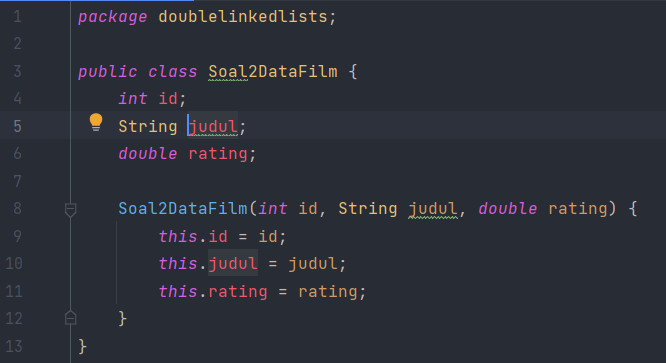


Pencarian Data

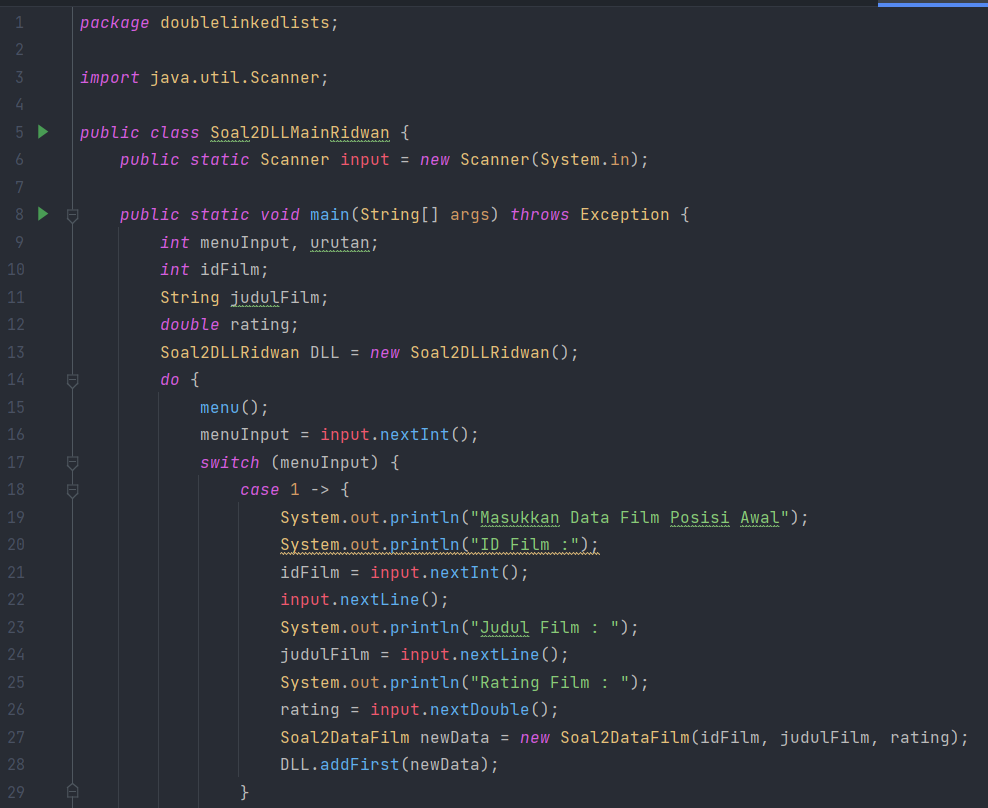


## Jawaban Nomor 2

### Source code Class Soal2DataFilm

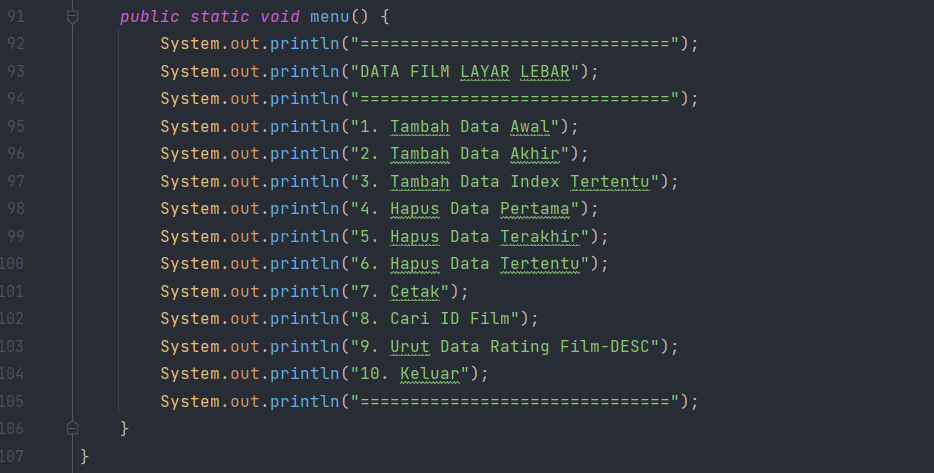


### Source code Class Soal2DLLMainRidwan



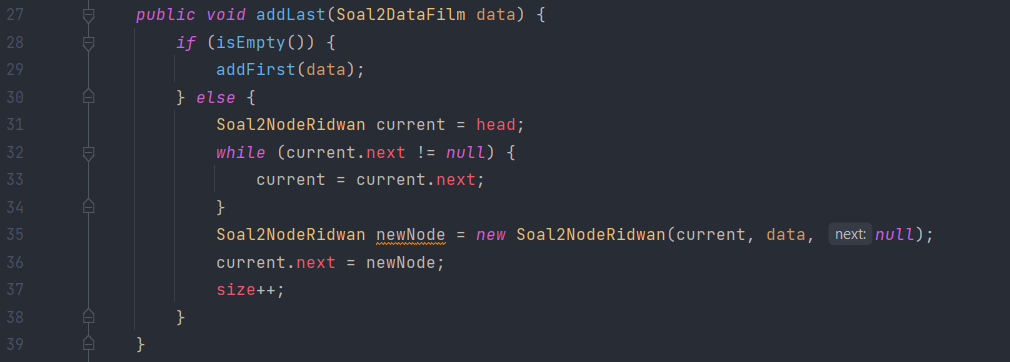






### Source code Class Soal2DLLRidwan

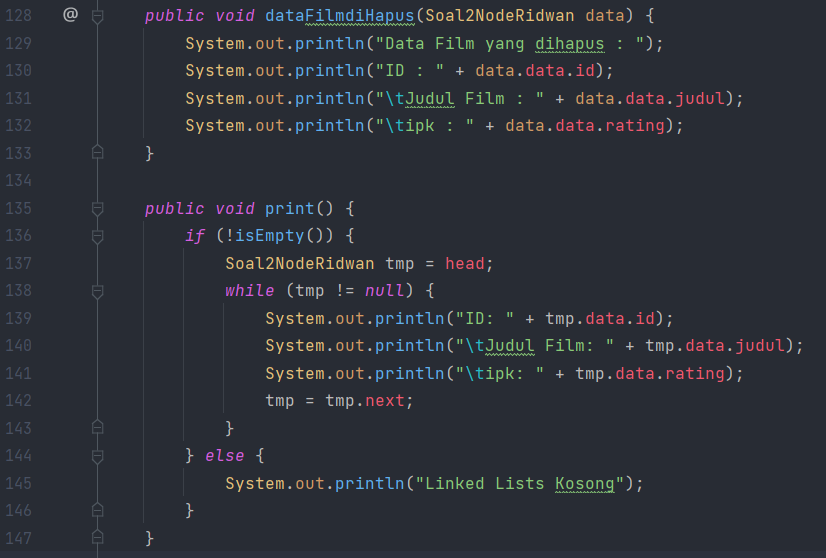


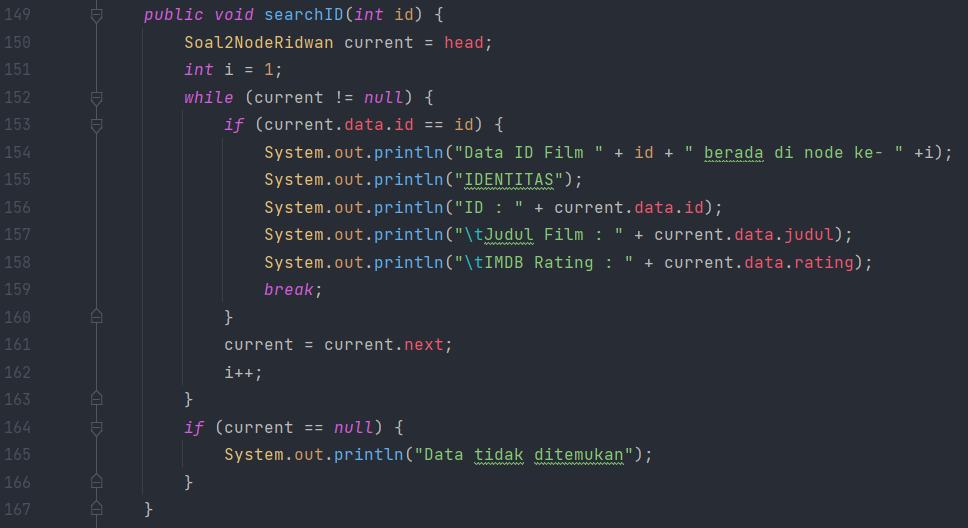


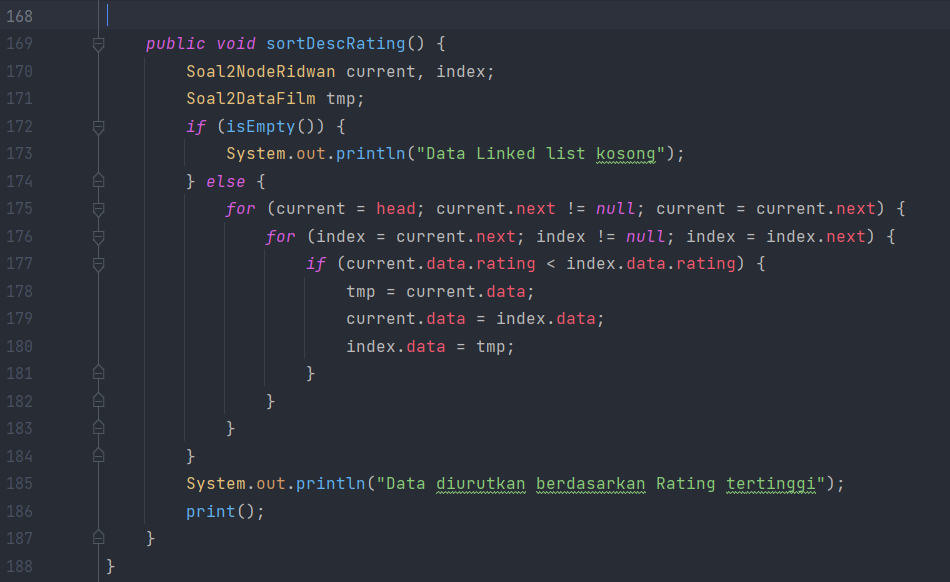




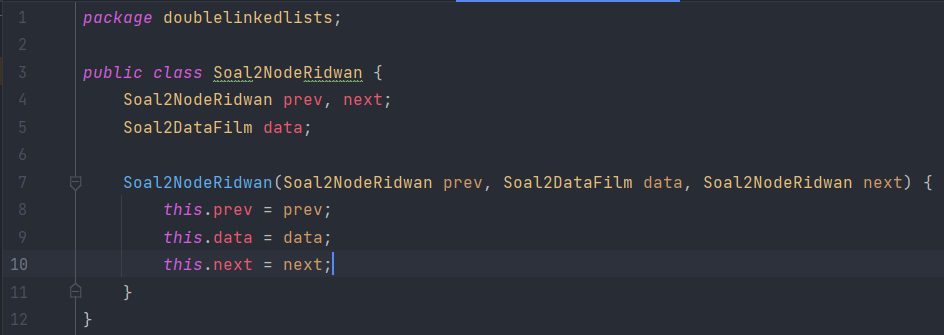




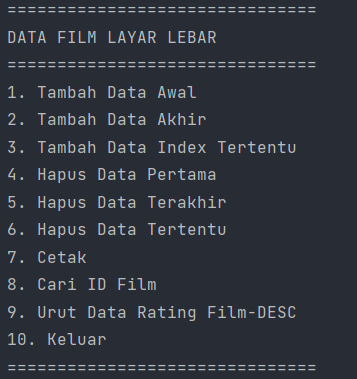




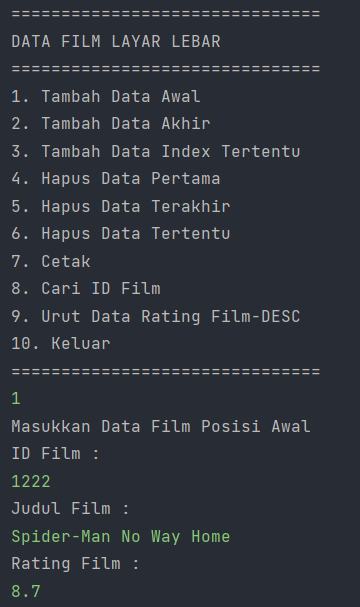
### Source code Class Soal2NodeRidwan



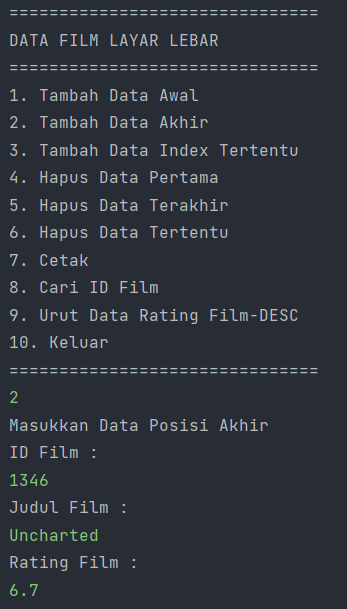
### Output Menu Awal



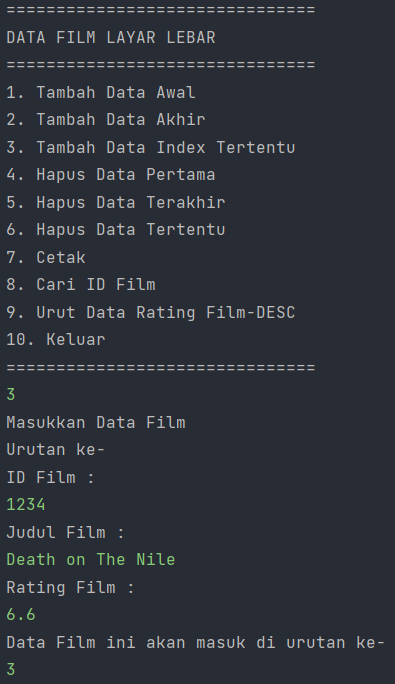
### Output Tambah Data Awal



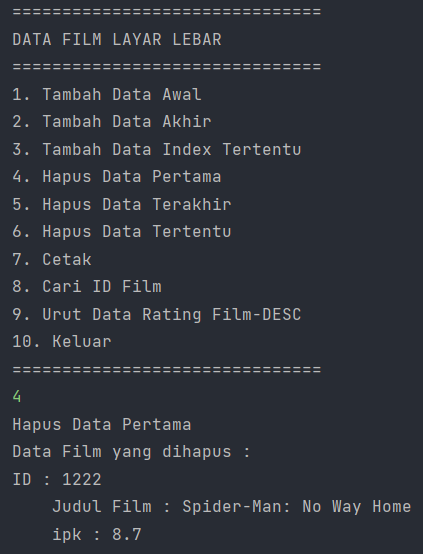
### Output Tambah Data Akhir



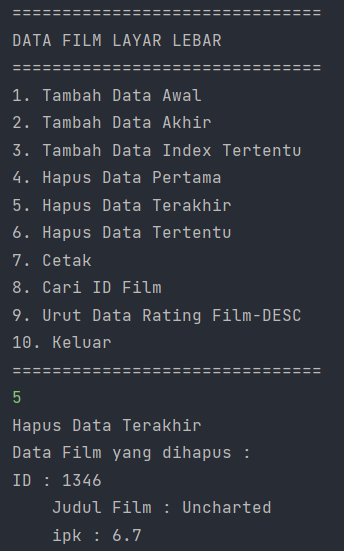
### Output Tambah Data Index Tertentu



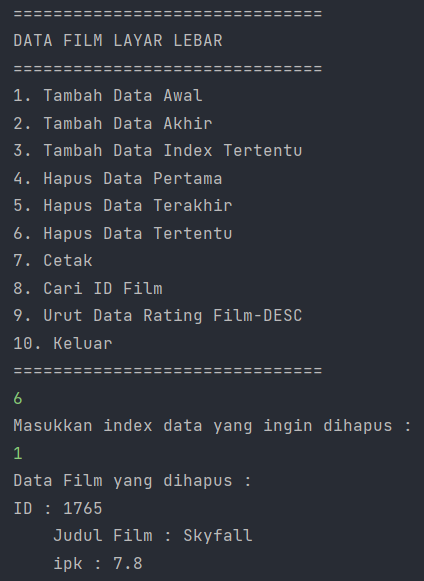
### Output Hapus Data Pertama



### Output Hapus Data Terakhir

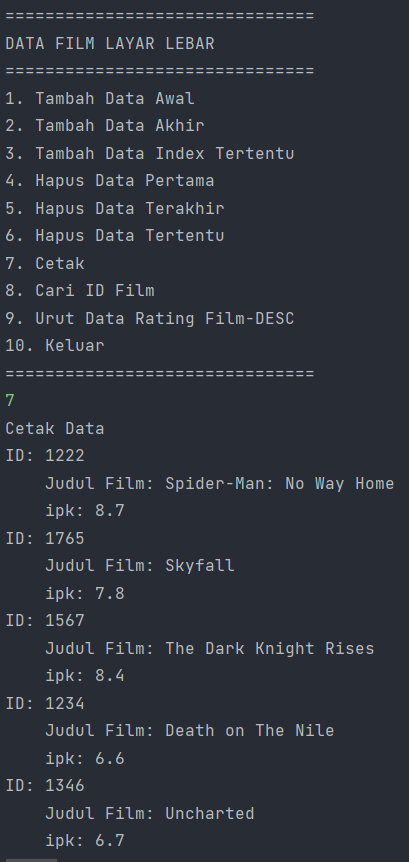


### Output Hapus Data Tertentu

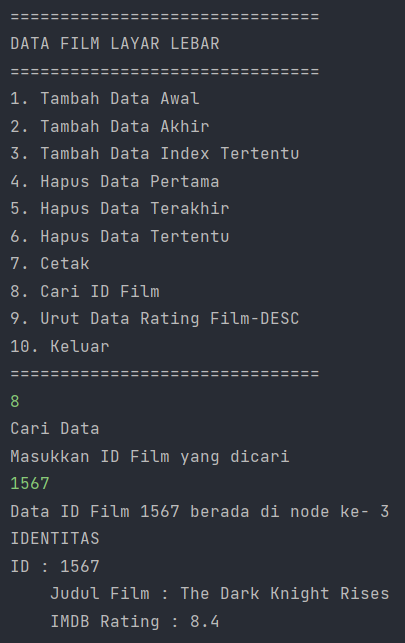


### Output Cetak

Sebelumnya data sudah saya inputkan kembali



### Output Cari ID Film



### Output Urut Data Rating-DESC



### Output Keluar

