**LAPORAN PRAKTIKUM**

**MATA KULIAH ALGORITMA DAN STRUKTUR DATA**

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

**PERTEMUAN - 12 - Double Linked List**

****

**Nama : M. Zidna Billah Faza**

**NIM : 2341760030**

**Prodi : D-IV Sistem Informasi Bisnis**

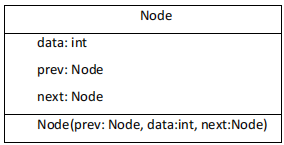
**JURUSAN TEKNOLOGI INFORMASI**

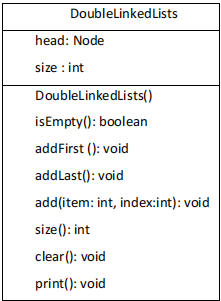
**POLITEKNIK NEGERI MALANG**

**2024**

**Percobaan 1**

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.

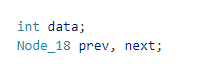




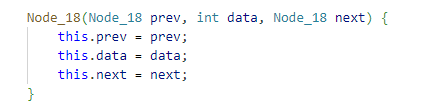
1. Buat paket baru dengan nama doublelinkedlists
2. Buat class di dalam paket tersebut dengan nama Node



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



1. Selanjutnya tambahkan konstruktor default pada class Node sesuai diagram di atas.



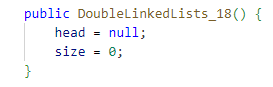
1. Buatlah sebuah class baru bernama DoubleLinkedLists pada package yang sama dengan node seperti gambar berikut:



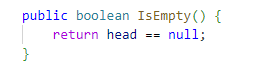
1. Pada class DoubleLinkedLists tersebut, deklarasikan atribut sesuai dengan diagram class di atas.



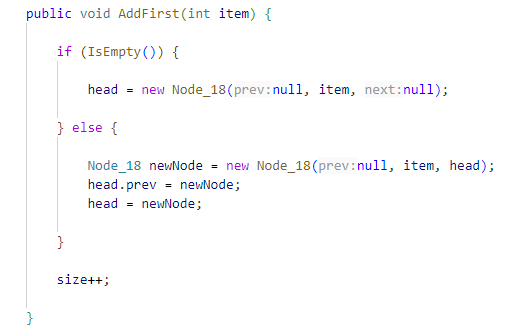
1. Selajuntnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut



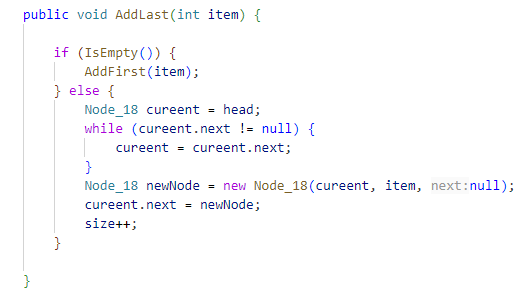
1. Buat method isEmpty(). Method ini digunakan untuk memastikan kondisi linked list kosong



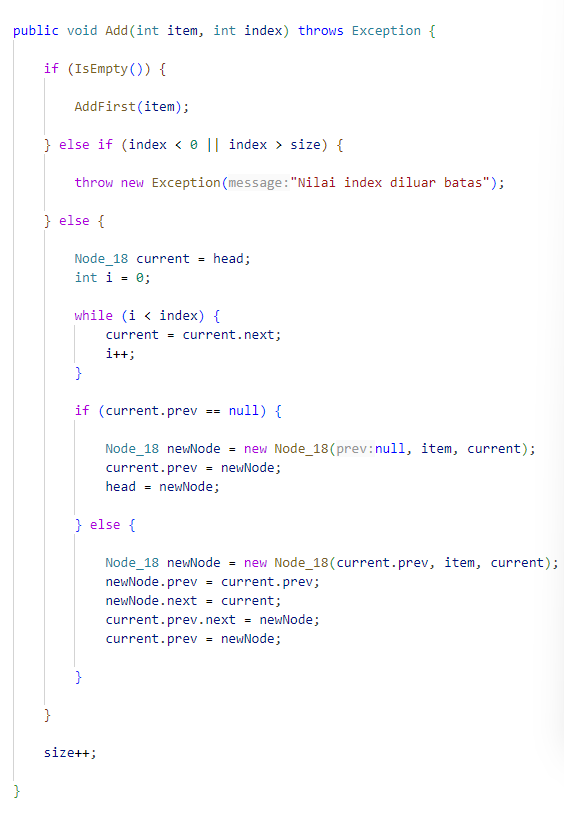
1. Kemudian, buat method addFirst(). Method ini akan menjalankan penambahan data di bagian depan linked list.



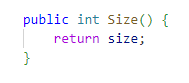
1. Selain itu pembuatan method addLast() akan menambahkan data pada bagian belakang linked list.



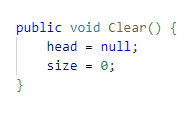
1. Untuk menambahkan data pada posisi yang telah ditentukan dengan indeks, dapat dibuat dengan method add(int item, int index)



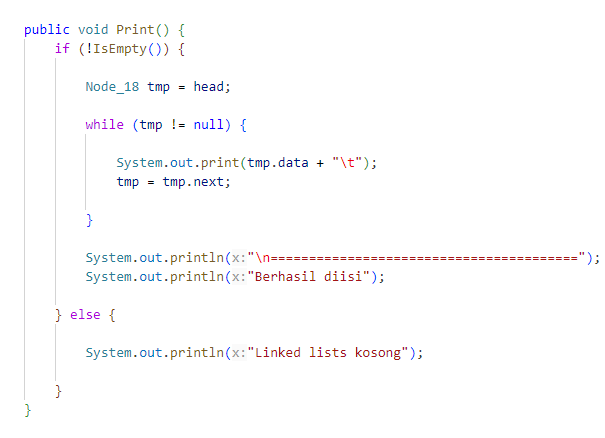
1. Jumlah data yang ada di dalam linked lists akan diperbarui secara otomatis,sehingga dapat dibuat method size() untuk mendapatkan nilai dari size.



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



1. 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.



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

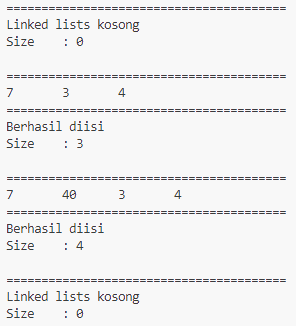


1. Pada main class pada langkah 16 di atas buatlah object dari class DoubleLinkedLists kemudian eksekusi potongan program berikut ini.



1. Verifikasi Hasil Percobaan





**Pertanyaan Percobaan 1**

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

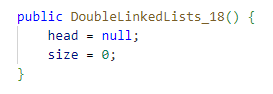
Single linked list adalah struktur data di mana setiap node memiliki dua bagian: data yang disimpan dan referensi ke node berikutnya dalam urutan. Artinya, penelusuran hanya bisa dilakukan ke depan.

Sementara itu, double linked list memiliki tiga bagian: data, referensi ke node sebelumnya, dan referensi ke node berikutnya. Dengan demikian, double linked list memungkinkan penelusuran ke depan dan ke belakang. Namun, double linked list memerlukan lebih banyak ruang memori untuk menyimpan referensi tambahan.

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

Atribut next dan prev dalam kelas Node digunakan untuk menunjukkan referensi ke node berikutnya (next) dan node sebelumnya (prev) dalam suatu struktur data yang berurutan

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



Inisialisasi atribut head dengan null dan size dengan 0 dalam konstruktor kelas DoubleLinkedLists\_18 bertujuan untuk menetapkan kondisi awal dari linked list ganda

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



Ini karena node baru akan menjadi node pertama dalam linked list, sehingga tidak memiliki node sebelumnya (prev)

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

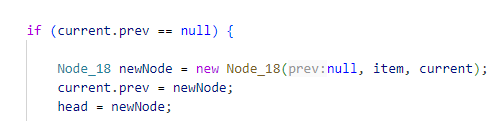
Statement tersebut memiliki arti bahwa kita mengatur referensi prev dari node yang sebelumnya menjadi node baru yang ditambahkan

1. Perhatikan isi method addLast(), apa arti dari pembuatan object Node dengan mengisikan parameter prev dengan current, dan next dengan null?



Pembuatan objek Node dengan mengisikan parameter prev dengan current dan next dengan null dalam metode addLast() memiliki arti bahwa kita membuat node baru di akhir linked list dan mengatur node sebelumnya (node terakhir sebelum penambahan) sebagai node sebelumnya dari node baru sehingga node baru akan berada di linked list terakhir

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

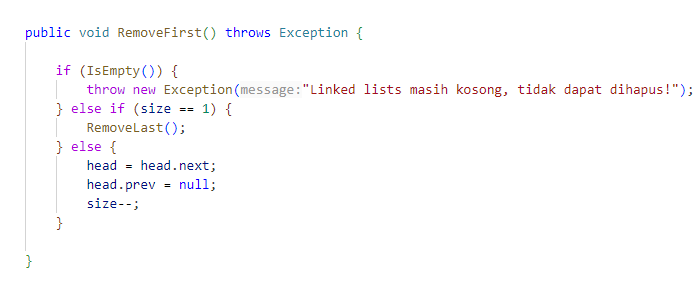


jelaskan maksud dari bagian tersebut.

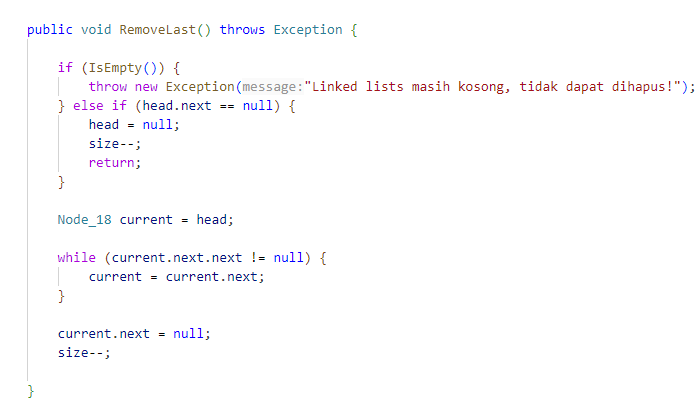
Potongan kode tersebut bertujuan untuk menambahkan node baru pada indeks tertentu dalam linked list. Jika indeks yang dimasukkan adalah 0 atau linked list kosong, maka node baru akan ditambahkan di awal linked list. Jika tidak, node baru akan dimasukkan di tengah linked list sesuai dengan index yang kita inputkan

**Percobaan 2**

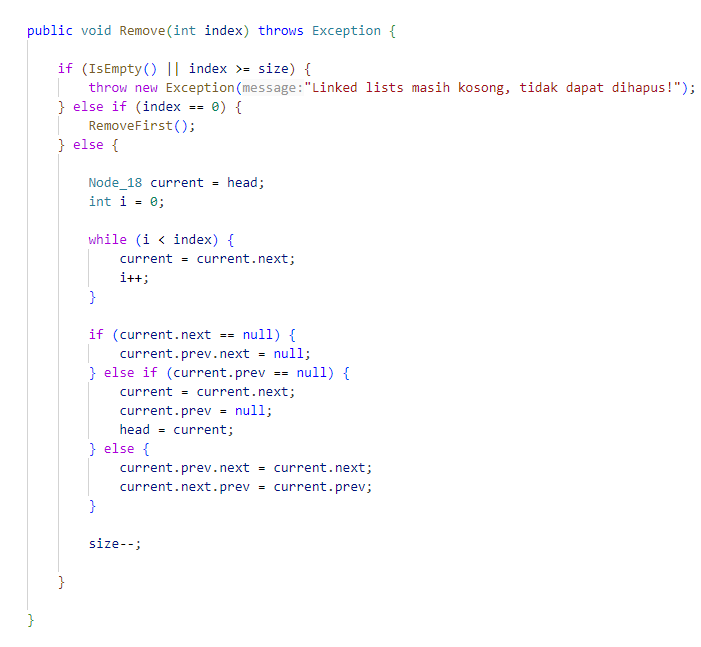
1. Buatlah method removeFirst() di dalam class DoubleLinkedLists



1. Tambahkan method removeLast() di dalam class DoubleLinkedLists.



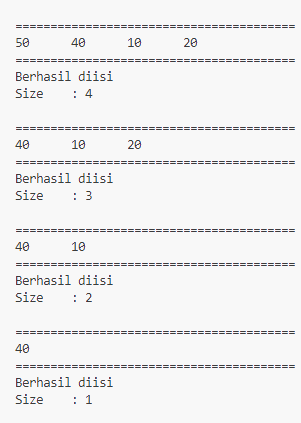
1. Tambahkan pula method remove(int index) pada class DoubleLinkedLists dan amati hasilnya



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



1. Verifikasi Hasil Percobaan



**Pertanyaan Percobaan 2**

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

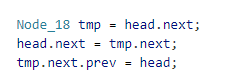


Statement tersebut digunakan untuk menggeser "head" list ke elemen berikutnya setelah yang pertama dihapus. Namun, setelah langkah ini, elemen sebelumnya yang merupakan elemen pertama sebelumnya masih memiliki referensi ke elemen yang baru menjadi kepala. Oleh karena itu, pernyataan head.prev = null; dibutuhkan untuk memutuskan referensi mundur dari elemen baru yang menjadi head.

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

Dengan mengecek apakah data setelah head adalah null. Jika setelah head adalah null maka data pada head merupakan data terakhir.

1. Jelaskan alasan potongan kode program di bawah ini tidak cocok untuk perintah remove!



Kode tersebut tidak cocok untuk perintah remove karena itu tidak memperhitungkan kemungkinan bahwa elemen yang dihapus adalah elemen terakhir dalam daftar. Ketika Anda mencoba menghapus elemen terakhir, baris kode tmp.next.prev = head; akan menyebabkan NullPointerException karena tmp.next akan berisi null. Itu artinya, tidak ada properti prev yang bisa diakses dari null. Jadi, potongan kode ini hanya akan berhasil jika elemen yang akan dihapus bukan elemen terakhir.

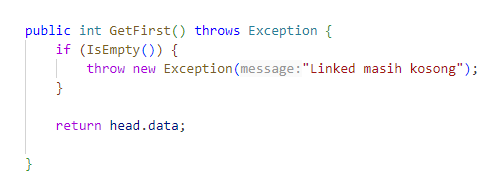
1. Jelaskan fungsi kode program berikut ini pada fungsi remove!



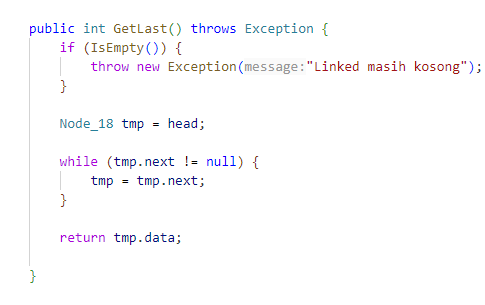
Kode tersebut digunakan untuk mengatur kembali pointer (atau referensi) dari elemen sebelum dan sesudah elemen yang dihapus, sehingga elemen yang dihapus dilewati dalam list dan dihapus dari struktur data

**Percobaan 3**

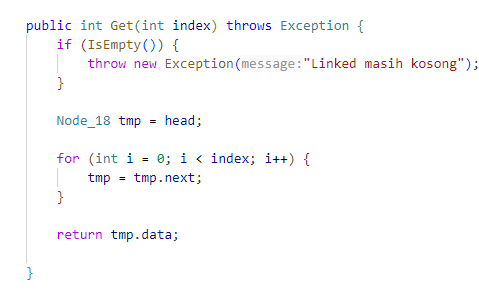
1. Buatlah method getFirst() di dalam class DoubleLinkedLists untuk mendapatkan data pada awal linked lists.



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

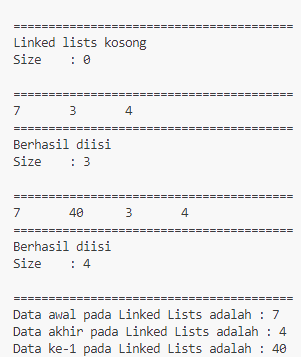


1. Method get(int index) dibuat untuk mendapatkan data pada indeks tertentu



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





**Pertanyaan Percobaan 3**

1. Jelaskan method size() pada class DoubleLinkedLists!

Method size() pada kelas DoubleLinkedLists digunakan untuk mengembalikan jumlah elemen yang saat ini ada dalam linked list.

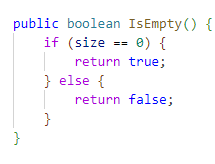
1. Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke-1!

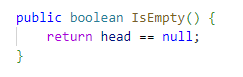
Untuk mengatur indeks pada double linked lists agar dimulai dari indeks ke-1, dapat membuat sebuah variabel tambahan yang bertindak sebagai indeks ke-1, seperti `index`, yang diinisialisasi dengan nilai 1. Saat menambahkan elemen pertama ke linked list, set variabel `index` ini ke 1, dan saat menambahkan elemen baru setelahnya, tingkatkan nilai `index` sebesar 1. Ketika mengakses atau menghapus elemen berdasarkan indeks, hitung indeks aktual dalam linked list dengan mengurangi 1 dari nilai indeks yang digunakan di luar linked list. Ini memungkinkan untuk berinteraksi dengan linked list menggunakan indeks yang lebih intuitif dimulai dari 1.

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

* Double Linked Lists, fungsi Add memungkinkan penambahan elemen di depan, di belakang, atau di tengah linked list dengan mudah karena setiap elemen memiliki referensi ke elemen sebelumnya (prev) dan setelahnya (next).
* Single Linked Lists, fungsi Add sebagai penambahan elemen seringkali hanya dimungkinkan di awal (head) atau di akhir (tail) linked list, karena setiap elemen hanya memiliki referensi ke elemen berikutnya. Untuk menambahkan elemen di tengah linked list, Anda harus mencari elemen sebelumnya, yang memerlukan waktu lebih lama karena Anda harus melakukan pencarian dari awal linked list untuk menemukan elemen tersebut.

1. Jelaskan perbedaan logika dari kedua kode program di bawah ini!



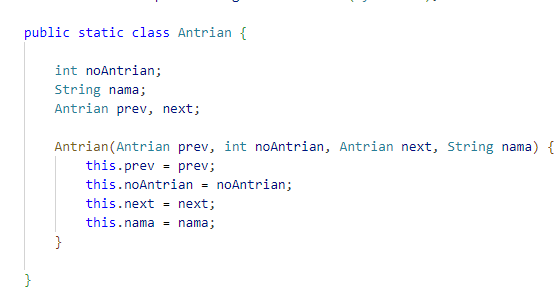


kode kedua langsung memeriksa status head, sementara kode pertama memeriksa ukuran linked list. Meskipun keduanya memberikan hasil yang sama, pendekatan yang lebih langsung menggunakan head terlihat lebih sederhana dan mudah dipahami.

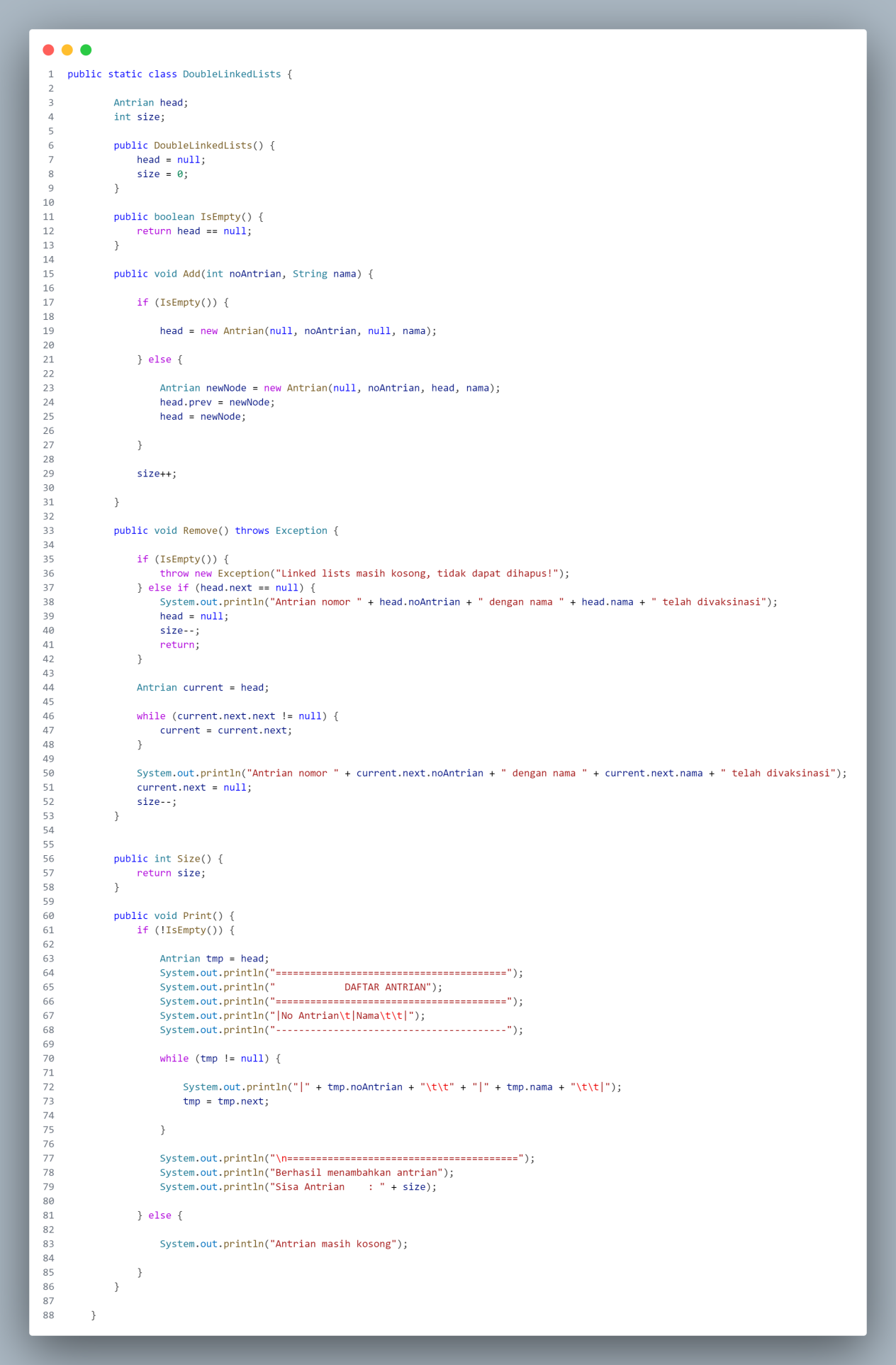
**Tugas Praktikum 1**

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)

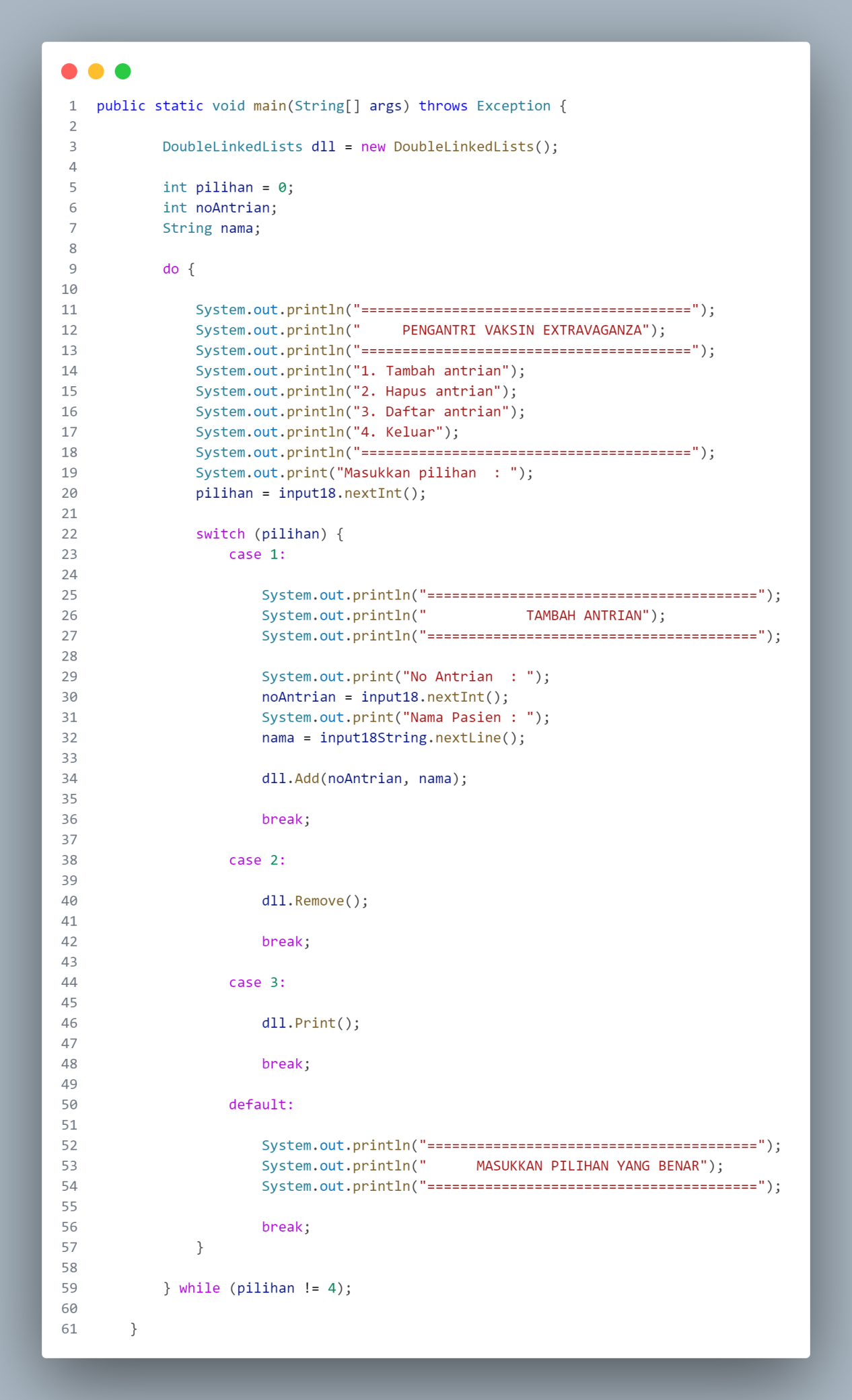
1. Program (class Antrian)



1. Program (class DoubleLinkedLists)



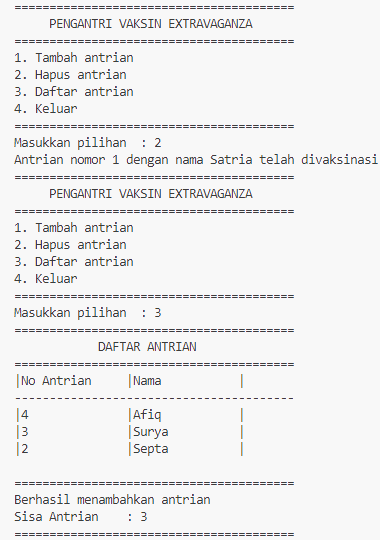
1. Program (class main)



1. Output Tambah Antrian

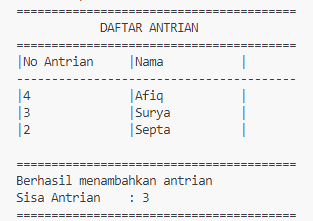


1. Output Hapus Antrian



Pada antrian selanjutnya berganti menjadi septa sesuai dengan urutan

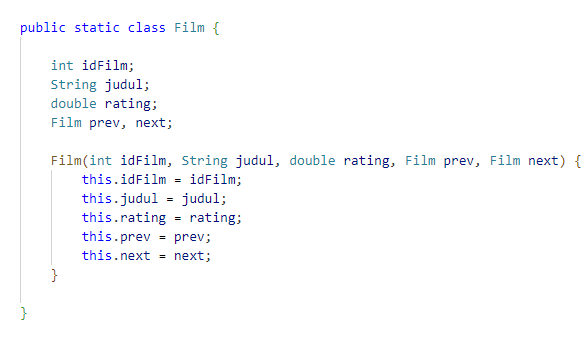
1. Output Daftar Antrian



**Tugas Praktikum 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.

1. Program (class Film)

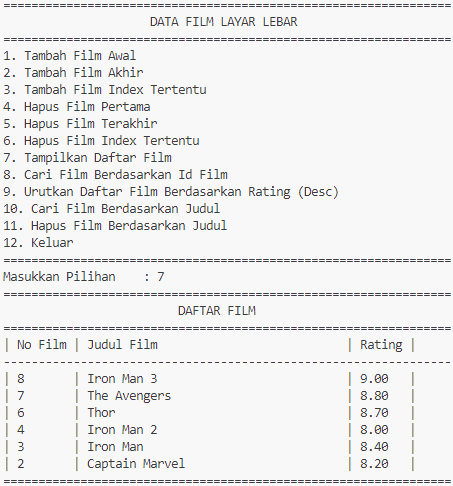


1. Program (class DoubleLinkedLists dan class main)

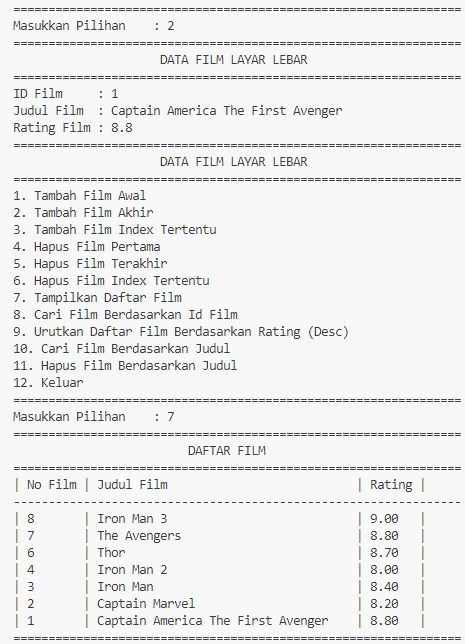
*Mohon maaf ibu Triana karena kode saya pada class ini banyak sehingga tidak dapat saya tampilkan disini. Ibu bisa melihat hasil outputnya atau melihat repository saya yang akan saya sertakan dibawah*

1. Output Tambah Film Awal

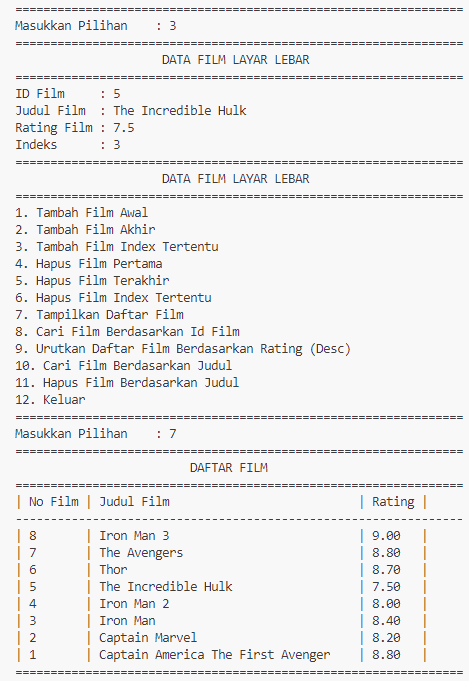




1. Output Tambah Film Akhir



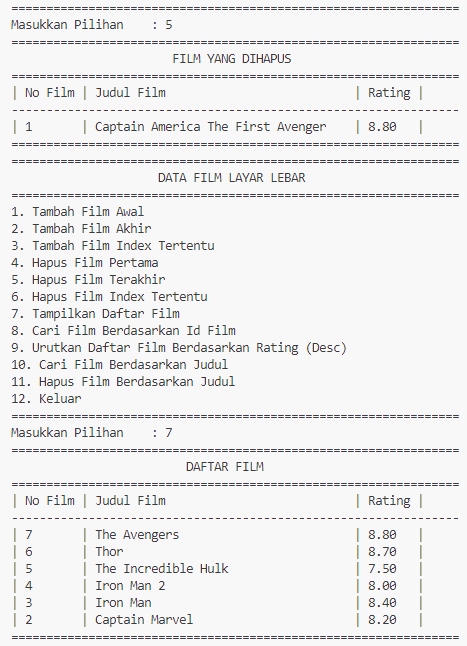
1. Output Tambah Film Index Tertentu



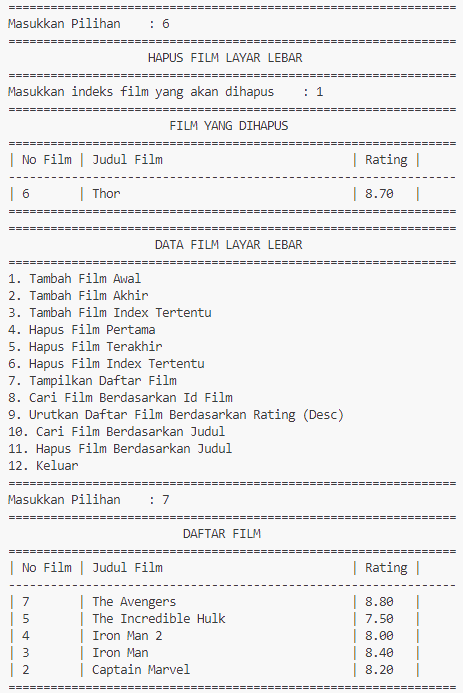
1. Output Hapus Film Pertama



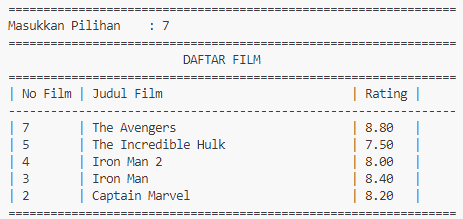
1. Output Hapus Film Terakhir



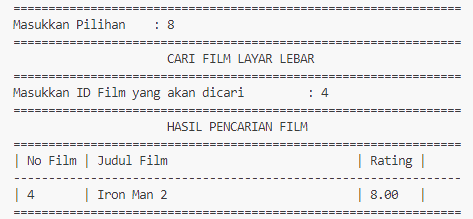
1. Output Hapus Film Index Tertentu



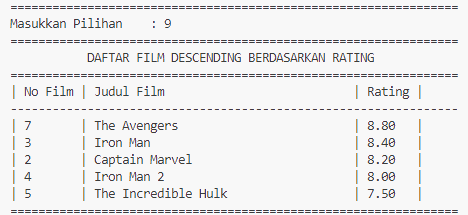
1. Output Tampilkan Daftar Film



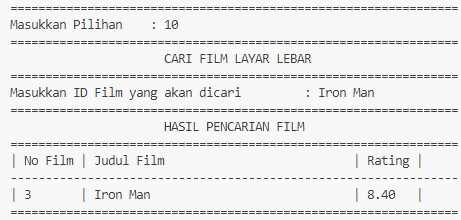
1. Output Cari Film Berdasarkan Id Film



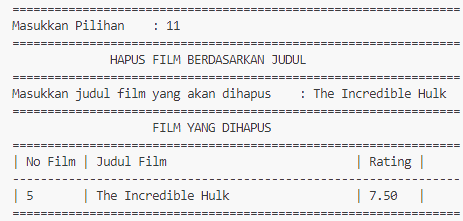
1. Output Urutkan Daftar Film Berdasarkan Rating (Desc)

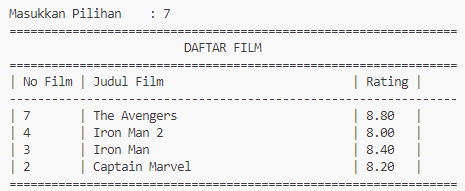


1. Output Cari Film Berdasarkan Judul (Tambahan)



1. Output Hapus Film Berdasarkan Judul (Tambahan)





1. Output Keluar



Repository : <https://github.com/zidnafaz/Praktikum-Algoritma-Struktur-Data>