**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**

****

**Oleh:**

**DZULFIKAR MUHAMMAD AL GHIFARI**

**NIM. 2341760071**

**SIB-1F / 08**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

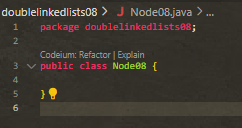
**POLITEKNIK NEGERI MALANG**

**PRAKTIKUM 12**

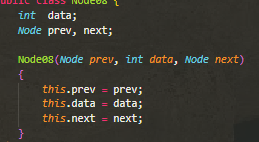
**12.2 Percobaan 1**

**12.2.1 Langkah Langkah percobaan**

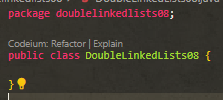
1. Membuat package dan class



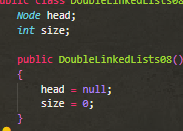
1. Deklarasi variable dan constructor



1. Membuat package dan class double linked lists



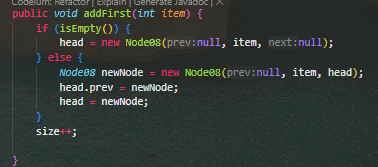
1. Deklarasi variable dan konnstruktor



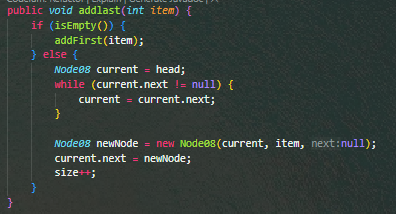
1. Menambahkan method isEmpty



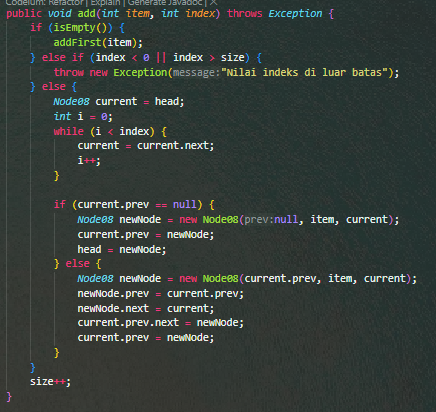
1. Menambahkan method addFirst



1. Menambahkan method addLast



1. Menambahkan method add



9. Membuat method size



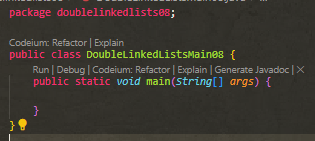
10. Membuat method clear



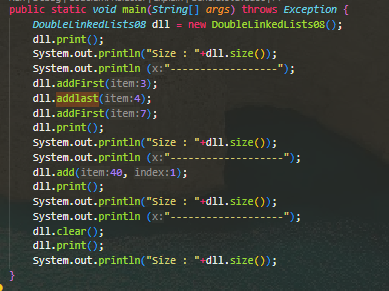
11. Membuat method print



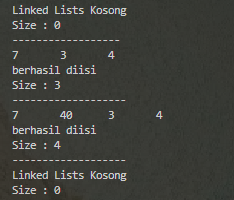
12. Membuat class main



13. Mengisi method main



**VERIFIKASI HASIL PERCOBAAN 12.2.2**



**PERTANYAAN 12.2.3**

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

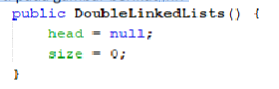
Jika single linked hanya ada satu arah, yaitu dari node awal – akhir, dan hanya memiliki satu node yaitu next.

Double linked memiliki dua arah, yaitu bolak balik antar node, dan memiliki dua node yaitu next dan prev.

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

Attribute next berfungsi untuk menunjukan node berikutnya, prev berfungsi untuk menunjukan node sebelumnya.

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



Head menunjukan pointer node pertama dalam daftar, size menyimpan jumlah node dalam daftar

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

**Node newNode = new Node(null, item, head)**

AddFirst befungsi untuk menambahkan data pada saat pertama kali. Karena isi daftar node pada awal berisikan 0, tidak ada data. Jadi tidak ada yang bisa untuk dilakukan prev

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

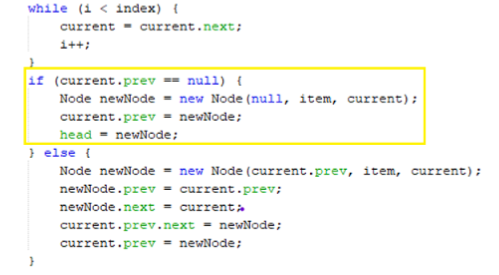
Untuk mengatur node yang sebelumnya menjadi head untuk menunjuk ke node baru yang kita tambahkan newNode

**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);**

AddLast befungsi untuk menambahkan data pada urutan terakhir. Karena akan di tempatkan pada posisi terakhir. Jadi tidak ada yang bisa untuk dilakukan next

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



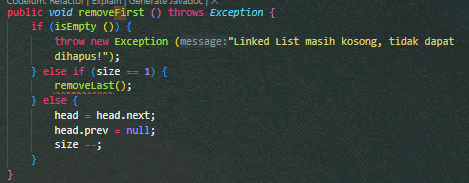
**jelaskan maksud dari bagian yang ditandai dengan kotak kuning**

untuk mendeteksi apakah posisi sebelumnya null, jika ya maka posisi sebelumnya akan di isi oleh data baru, dan head akan ada pada posisi data yang baru saja ditambahkan

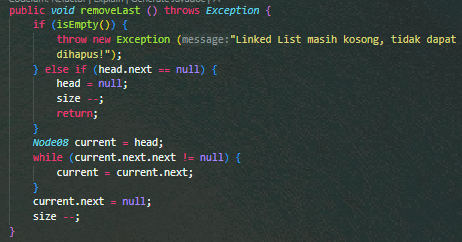
**12.3 Percobaan 2**

**12.3.1 Langkah-langkah Percobaan**

1. Menambahkan method removeFirst



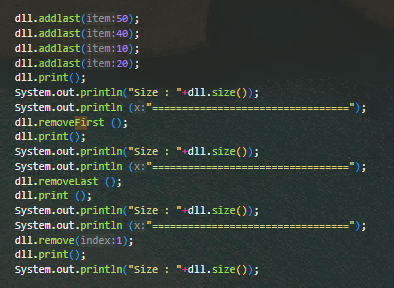
1. Menambahkan method removeLast



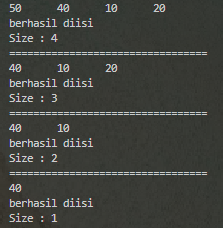
1. Menambahkan method remove



1. Menambahkan kode pada main



**VERIFIKASI HASIL PERCOBAAN 12.3.2**



**PERTANYAAN 12.3.3**

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

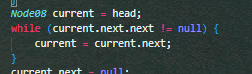
**head = head.next;**

**head.prev = null;**

head=head.next berfungsi untuk memperbarui pointer head yang menunjukan ke node berikutnya setelah node pertama yang akan dihapus.

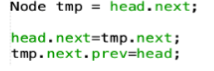
head.prev=null berfungsi untuk menghapus ponter prev karena head berasa pada daftar pertama

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



Dengan menggunakan iterasi loop while, mencari current.next hingga ke daftar yang terakhir. Karena daftar terakhir tidak bisa melakukan .next, jadi data tersebut menunjukan data terakhir, dan ponter akan di simpan di variable current

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



Karena kode program di atas tidak memeriksa kondisi spesifik dari daftar

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



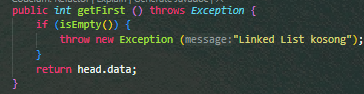
Kode program pertama berfungsi memperbarui pointer next dari node sebelumnya untuk menunjuk ke node berikutnya

Kode program kedua berfungsi memperbarui ponter prev dari node berikutnya untuk menunjuk ke node sebelumnya

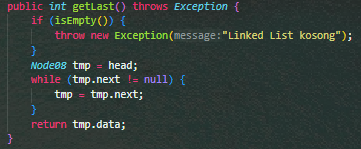
**12.4 Percobaan 2**

**12.4.1 Langkah-langkah Percobaan**

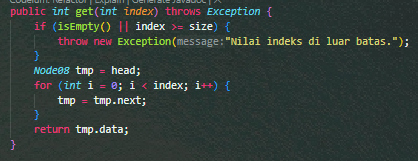
1. Menambahkan method getFIrst



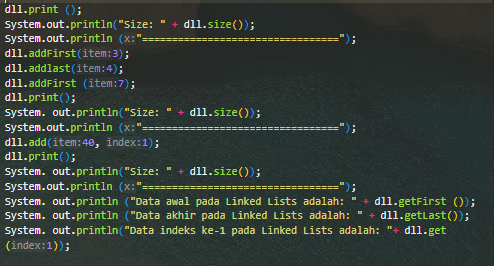
1. Menambahkan method getLast



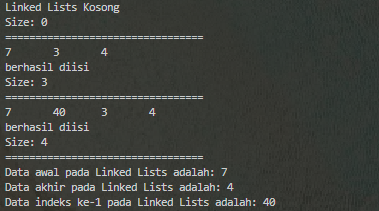
1. Menambahkan method get



1. Menambahkan kode pada main



**VERIFIKASI HASIL PERCOBAAN 12.4.2**



**PERTANYAAN 12.4.3**

1. **Jelaskan method size() pada class DoubleLinkedLists!**

Size() mengembalikan nilai size yang telah di atur pada awal menjalankan program

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

Secara default index umumnya dimulai dari 0. Namun dengan dengan sedikit mengubah pada method add dengan memberikan validasi untuk index seperti dibawah ini

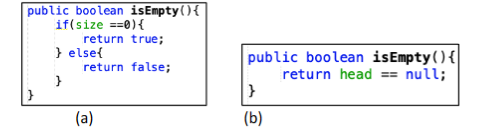


1. **Jelaskan perbedaan karakteristik fungsi Add pada Double Linked Lists dan Single Linked Lists!**

Penambahan pada single linked hanya satu arah dari depan ke belakang, tidak bisa menambahkan elemen di tengah daftar

Penambahan pada double linked memiliki 2 arah, bisa dari depan dan belakang, bisa menyisipkan kode di mana saja

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



pada kode A akan memeriksa jumlah elemen, jika jumlah elemen tersebut bernilai 0 maka akan mengembalikan true, namun jika tidak sama dengan 0 maka akan mengembalikan false

Pada kode B akan memeriksa ponter head apakah berisikan null, jika iya maka akan mengembalikan true, jika tidak null maka kan mengembalikan false

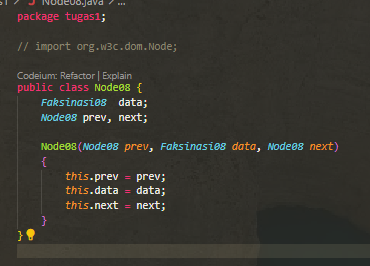
**8.4 TUGAS**

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

Class faksinasi



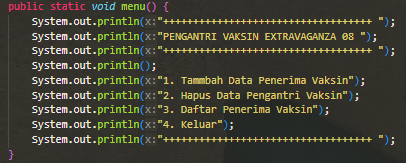
Class Node

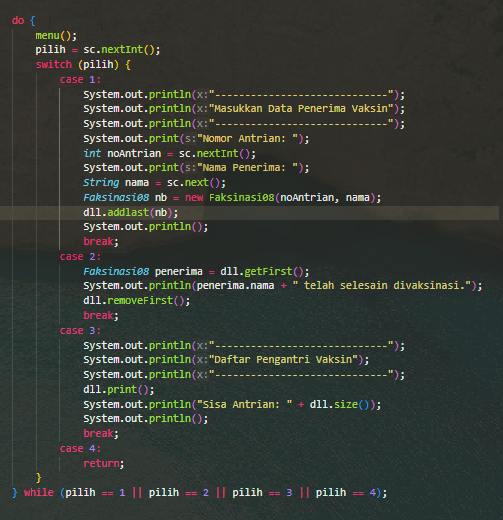


Class double berisikan hasil copy dari percobaan 1-3

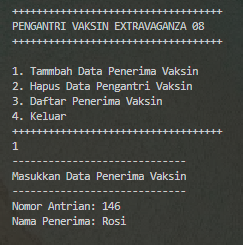
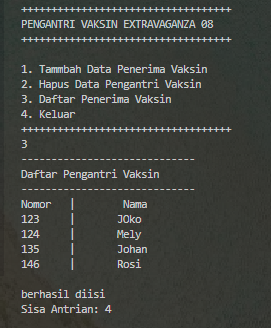


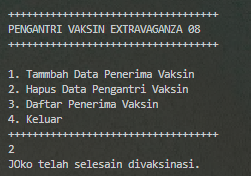
Class Main





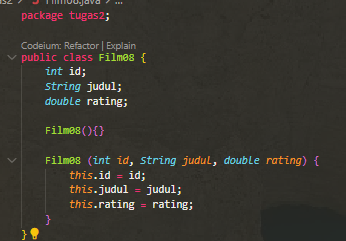
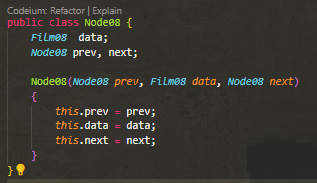
Hasil Compile



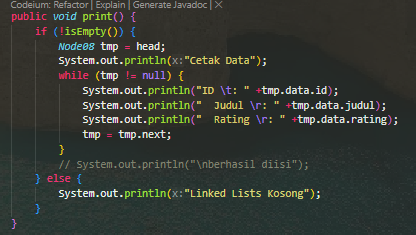


1. Buatlah program daftar film yang terdiri dari id, judul dan rating menggunakan double linked lists, bentuk program memiliki fitur pencarian melalui ID Film dan pengurutan Rating secara descending. Class Film wajib diimplementasikan dalam soal ini.

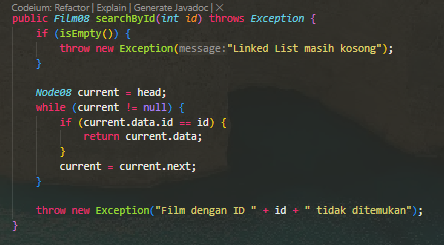
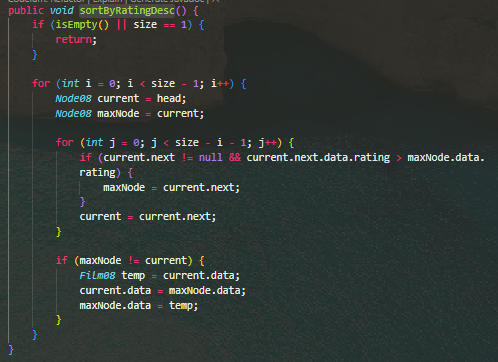
Class film Class node

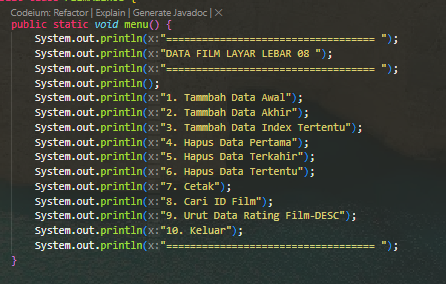
Perubahan pada class doublelinkedlist

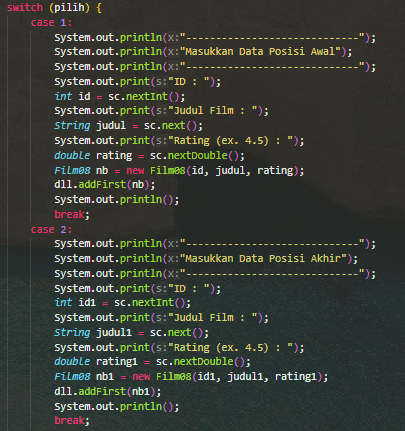
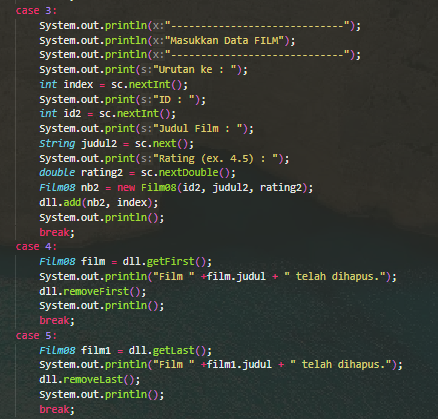


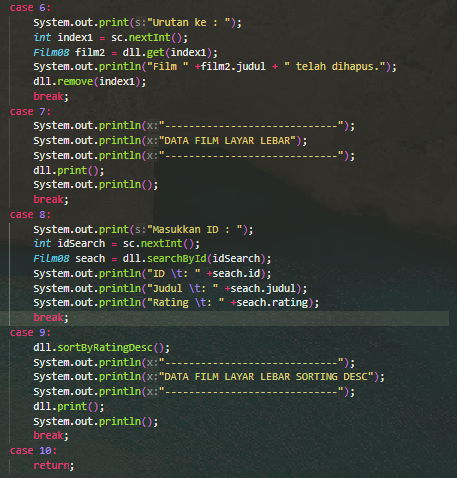
Penambahan pada class double linkedlist

Class main





Hasil compile