**POLITEKNIK NEGERI MALANG**

**TEKNOLOGI INFORMASI**

**TEKNIK INFORMATIKA**



**Nama: Abdul Rahman Hanif Darmawan**

**NIM: 244107020232**

**Kelas: TI-1A**

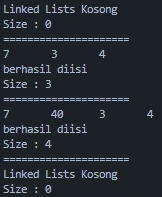
**Prodi: D4-TEKNIK INFORMATIKA**

# JOBSHEET 12

# Double Linked Lists

12.2.2 Verifikasi Hasil Percobaan





12.2.3 Pertanyaan Percobaan

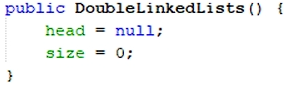
1. Jelaskan perbedaan antara single linked list dengan double linked lists!

* Single linked list hanya memiliki pointer ke node berikutnya (next), sedangkan double linked list memiliki pointer ke node berikutnya (next) dan node sebelumnya (prev).

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

* next: Menunjuk ke node berikutnya dalam daftar dan prev: Menunjuk ke node sebelumnya dalam daftar. (Khusus untuk double linked list).

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



* head = null;: Mengindikasikan bahwa daftar kosong (belum ada elemen) dan size = 0;: Mengatur jumlah elemen dalam daftar menjadi nol karena daftar masih kosong.

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

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

* Karena node baru yang ditambahkan di awal (addFirst) akan menjadi node pertama dalam daftar, sehingga tidak ada node sebelumnya yang ditunjuk oleh prev.

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

* Jika daftar tidak kosong, statement ini mengatur pointer prev dari node yang saat ini menjadi head untuk menunjuk ke newNode yang baru saja ditambahkan di depannya. Ini mengikat node lama ke node baru.

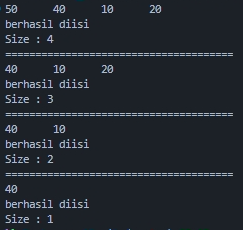
1. Perhatikan isi method addLast(), apa arti dari pembuatan object Node dengan mengisikan parameter prev dengan current, dan next dengan null?

Node newNode = new Node(current, item, null);

* prev dengan current: Node baru akan ditambahkan di akhir daftar, jadi node current (yang merupakan node terakhir sebelum penambahan) akan menjadi node sebelumnya.
* next dengan null: Node baru ini akan menjadi node terakhir, sehingga tidak ada node berikutnya yang ditunjuk oleh next.

12.3.2 Verifikasi Hasil Percobaan





12.3.3 Pertanyaan Percobaan

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

head = head.next;

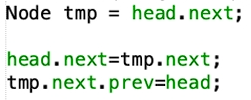
head.prev = null;

* Baris pertama menggeser head ke node berikutnya, menjadikan node kedua sebagai kepala baru. Baris kedua memastikan node kepala baru tidak memiliki referensi prev

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

* Dideteksi dengan mengiterasi sampai current.next.next == null. Yang berarti current.next adalah node terakhir.

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



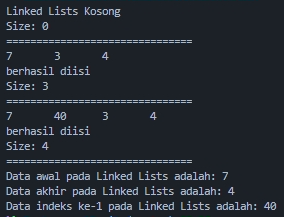
* Kode ini hanya cocok untuk menghapus node kedua. Jika head.next adalah null (list hanya punya satu node) atau tmp.next adalah null (list hanya punya dua node), kode ini akan menyebabkan NullPointerException.

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



* Kode ini menyambungkan node sebelum current dengan node setelah current, dan sebaliknya. Ini efektif "melewati" node current, sehingga menghapusnya dari list tanpa mengubah head atau tail secara langsung.

12.4.2 Verifikasi Hasil Percobaan



12.4.3 Pertanyaan Percobaan

1. Jelaskan method size() pada class DoubleLinkedLists!

* Method size() mengembalikan jumlah elemen yang saat ini ada dalam DoubleLinkedLists.

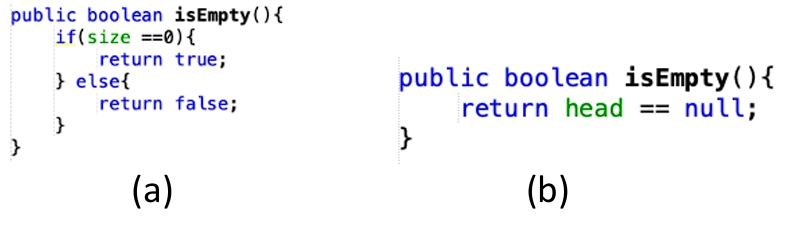
1. Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke- 1!

* Untuk membuat indeks dimulai dari 1, Perlu menyesuaikan semua logika yang berinteraksi dengan indeks (seperti add, remove, get) dengan mengrangi 1 dari nilai indeks yang diterima sebelum digunakan untuk traversing atau perhitungan (misalnya, index - 1).

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

* Pada Double Linked List, fungsi add juga memperbarui pointer prev pada node baru dan/atau node tetangga, selain pointer next. Pada Single Linked List, fungsi add hanya memperbarui pointer next.

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



* **(a):** Menentukan apakah list kosong berdasarkan nilai variabel size. Jika size adalah 0, maka list dianggap kosong.
* **(b):** Menentukan apakah list kosong berdasarkan apakah head (pointer ke node pertama) adalah null. Jika head adalah null, berarti tidak ada node dalam list.

12.5 Tugas Praktikum

Tugas 1

Node.java



VaksinasiQueue.java

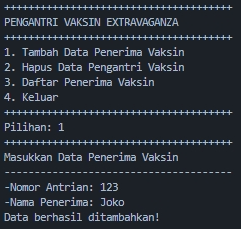
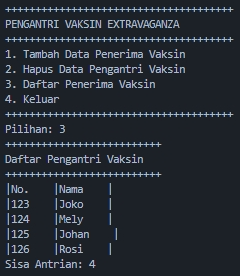
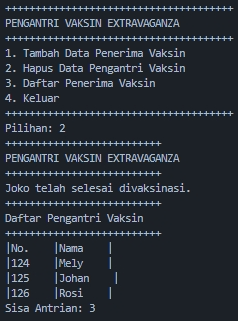




VaksinMain.java



Output

Tugas 2

Film.java



Node.java



DoubleLinkedList.java







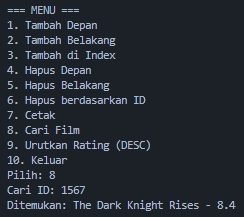
FilmMain.java





Output

Link Github

<https://github.com/baynobu/ALSD/tree/d14333c9b5e2c439eec889cd0314b5b7850af202/Pertemuan%2013>