Worksheet W1

Algoritma dan Struktur Data

Teknologi Informasi Kelas C

Nama:

Nabath Nuur Muhammad (245150700111031)

Dosen:

Putra Pandu Adikara, S.Kom., M.Kom.

Program Studi Teknologi Informasi
Jurusan Teknologi Informasi
Universitas Brawijaya

2020

Dari slide materi yang sudah dibagikan saya menulis ulang kode tersebut kemudian saya tambahkan method yang saya perlukan yaitu method untuk menampilkan data dari depan ke belakang (showForward) dan (showBackward). Selain itu saya juga memodifikasi beberapa bagian untuk mempermudah saya dalam modifikasi dan merapikan agar variabel lebih konsisten (semua data saya simpan dalam tipe data Object) supaya lebih mudah saya pahami.

Saya membagi jawaban saya menjadi 2 bagian, bagian pertama yaitu pembahasan kode Singly Linked List dan bagian kedua adalah pembahasan kode Doubly Linked List. Pada masing-masing kode saya menjelaskan method yang diminta dalam halaman latihan pada Tugas 1 yaitu sebanyak 5 method.

A. Singly Linkedlist

1. Jawaban method void insertAt(int index, Object data):

Method ini telah disampaikan dalam slide materi. Langkah kerja method ini:

- Pertama dilakukan pengecekan apakah index valid (tidak boleh < 0 dan tidak boleh lebih besar dari size).
- Jika index == 0, maka method addFirst(data) dipanggil, artinya data dimasukkan di awal.
- Jika index == size, maka method addLast(data) dipanggil, artinya data dimasukkan di akhir.
- Jika tidak, traversal dilakukan dengan for loop dari node pertama (head) sampai ke posisi index-1.
- Setelah ketemu posisi, node baru dibuat (newNode) lalu pointer node sebelumnya diarahkan ke node baru, dan pointer node baru diarahkan ke node berikutnya.
- size ditambah 1 setelah penyisipan berhasil.

2. Jawaban method void removeAt(int index):

- Pada method ini pertama saya memanggil method checkIndex(index) untuk memastikan apakah index valid. Jika tidak valid, maka akan dilempar exception.
- Jika index == 0, maka panggil removeFirst().
- Jika index == size-1, maka panggil removeLast().
- Jika tidak, traversal dilakukan hingga ke node sebelumnya (index-1). Node yang akan dihapus adalah prev.pointer, lalu saya hubungkan prev.pointer ke toRemove.pointer sehingga node target dilewati.
- Ukuran size dikurangi 1 setelah node dihapus.

3. Jawaban method Object get(int index):

- Pertama dilakukan validasi dengan checkIndex(index).
- Pengecekan dilakukan menggunakan for loop dari head sebanyak index kali.
- Setelah mencapai node target, nilai data pada node tersebut dikembalikan (return current.data).

4. Jawaban method boolean contains(Object data):

- Pengecekan Linked List dilakukan dari head menggunakan while loop.
- Setiap node dicek dengan equals. Jika sama, return true.
- Jika sudah sampai akhir (null) tanpa menemukan data, return false.

5. Jawaban method int indexOf(Object data):

- Pengecekan Linked List dilakukan dengan for loop dari head.
- Counter index dimulai dari 0.

- Jika current.data.equals(data), maka return nilai index.
- Jika traversal selesai tanpa menemukan, return -1.

A. Doubly Linkedlist

1. Jawaban method void insertAt(int index, Object data):

- Pertama dilakukan validasi index. Jika index tidak valid, ditangani oleh exception.
- Jika index == 0, maka data dimasukkan dengan addFirst(data).
- Jika index == size, maka data dimasukkan dengan addLast(data).
- Jika tidak, traversal dilakukan dengan for loop dari head sampai node ke-index.
- Node baru disisipkan di posisi tersebut dengan cara:
- newNode.prev = current.prev
- newNode.next = current
- current.prev.next = newNode
- current.prev = newNode
- size ditambah 1.

2. Jawaban method void removeAt(int index):

- Pertama dilakukan validasi index apakah tersedia.
- Jika index == 0, maka dipanggil removeFirst().
- Jika index == size-1, maka dipanggil removeLast().
- Jika tidak, traversal dilakukan dengan for loop menuju node ke-index.
- Setelah ketemu node target, pointer dihubungkan ulang:
- current.prev.next = current.next
- current.next.prev = current.prev
- Node target dilepas, size dikurangi 1.

3. Jawaban method Object get(int index):

- Dilakukan validasi index apakah tersedia.
- Jika index < size/2, traversal dilakukan dari depan (head) dengan for loop hingga mencapai index.
- Jika index >= size/2, traversal dilakukan dari belakang (tail) dengan for loop ke arah kiri.
- Setelah ketemu node target, return current.data.

4. Jawaban method boolean contains(Object data):

- Pengecekan Linked List dilakukan dari head menggunakan while loop.
- Setiap node dicek dengan equals. Jika sama, return true.
- Jika sudah sampai akhir (null) tanpa menemukan data, return false.

5. Jawaban method int indexOf(Object data):

- Pengecekan Linked List dilakukan dengan for loop dari head.
- Counter index dimulai dari 0.
- Jika current.data.equals(data), maka return nilai index.
- Jika traversal selesai tanpa menemukan, return -1.

Refleksi diri

- Saya merasa kesulitan saat memahami slide materi karena Materi Singly Linked List dan Doubly Linked List banyak persamaan dan juga banyak perbedaan, selain itu saya merasa menggunaan tipe data atau nama method di slide materi cukup membingungkan dan tidak konsisten. Sulusinya saya belajar lewat sumber lain di interner.
- Pada Doubly Linked List, saya sering kali bingung dalam menyambungkan antar pointer prev dan next sehingga sering terjadi kesalahan. Dengan perbanyak latihan saya bisa mengatasi masalah ini.
- Saya juga sering salah dalam mengatur size++ dan size-- setelah operasi insert atau remove, sehingga ukuran list tidak sesuai dengan data yang ada. Saya harus lebih teliti dan terus melakukan perbaikan dan pengecekan pada setiap tahap agar kode berjalan sesuai keinginan saya.

Dokumentasi

Berikut saya sertakan hasil dokumentasi kode program saya melalui link GitHub: https://github.com/nabathnm/ASDTugas1.git