

Jurnal 04: Doubly Linked List

Kerjakanlah soal berikut. Jawaban dinilai berdasarkan dua kriteria: a. Kode sesuai dengan permintaan soal, dan b. Sesi tanya-jawab dengan Asprak. Kerjakan sesuai dengan prinsip Java Coding Style.

Kosuzu merupakan seorang anak yang keluarganya memiliki usaha rental buku. Ia ingin melakukan manajemen peminjaman buku punya keluarganya menggunakan Linked List. Ia ingin agar buku-buku yang ditambahkan secara acak bisa ditelusuri, kemudian diupdate statusnya menjadi dipinjam/tidak dipinjam. Bantulah Kosuzu untuk membuat program tersebut!

Tentunya, setiap buku harus memiliki:

- ISBN
- Judul Buku
- Kategori Buku
- Harga Pinjam
- Status (Apakah sedang dipinjam atau tidak)

Fitur Program:

1. Penambahan buku, menambahkan buku baru ke dalam LinkedList.
2. Penghapusan buku, menghapus buku dari dalam LinkedList berdasarkan ISBN.
3. Edit buku, mengedit data buku di dalam LinkedList. Bisa digunakan untuk mengedit data umum seperti judul, dan juga untuk mengedit status dari dipinjam ke tidak dipinjam, atau sebaliknya.
4. Lihat buku, melihat list seluruh buku dari depan maupun dari belakang.

Persyaratan Implementasi:

- Hanya dibutuhkan dua class di sini, yaitu class POJO untuk Buku, dan class utama/main.
- Data buku yang disimpan **minimal** 5 buku.
- Ketika menyimpan data buku, 2 di antara mereka harus menggunakan addFirst.
- Delete **harus** berdasarkan ISBN.
- List harus bisa ditampilkan dari depan maupun belakang.
- **Wajib** menggunakan iterator.
- Harus ada **dua** data yang diedit.
- Data yang diedit adalah 2 data terakhir pada list, dengan data paling akhir wajib diedit terlebih dahulu. Spesifikasi edit bebas (ingin mengedit stok, nama, atau apapun **kecuali** ISBN).

Java Coding Style

Ikuti gaya penulisan berikut di kode program agar mendapat nilai maksimal

1. Setiap nama kelas, variabel dan konstanta harus dapat menggambarkan isinya.

Misal, membuat variabel untuk menyimpan jumlah penumpang di bus.

Deklarasi:

```
int penumpang = 0;           // Bagus, OK
int p = 0;                   // Tidak OK
```

2. Penulisan nama harus menggunakan huruf besar/kecil yang sesuai.

Nama kelas: UpperCamelCase. Misal: PenumpangBus

Nama variabel: lowerCamelCase. Misal: penumpangBus

Nama konstanta: CAPITAL_CASE. Misal: PENUMPANG_BUS

Nama method: lowerCamelCase. Misal: hitungPenumpangBus

3. Penggunaan tab/spasi ketika berbeda blok program, WAJIB masuk 1 tab atau 4 spasi ke dalam.

```
public class PenumpangBus {
    public static void main(String[] args) {
        int penumpang = 0;
        penumpang = penumpang + 5;
        penumpang = penumpang - 4 + 2;
        penumpang = penumpang - 1;
        penumpang = penumpang - 2 + 3;
        penumpang = penumpang - 2 + 5;
        penumpang = penumpang - 1 + 3;
        System.out.println(penumpang);
    }
}
```

4. Gunakan komentar seperlunya.

Javadoc comment (yang warna biru) hanya untuk mengomentari kelas, method dan variable

```
/** Jumlah penumpang di bus */  
int penumpang = 0;  
  
/* Ada 5 orang penumpang naik ke bus */  
penumpang = penumpang + 5;  
  
// Penumpang turun 4 orang, naik 2  
penumpang = penumpang - 4 + 2;
```

