**Laporan Praktikum Algoritma & Struktur Data**

**Jobsheet 12 – Double Linked list**

****

Nama: Aqil Rahmat Alifiandi

NIM: 2341760099

Prodi: D-IV Sistem Informasi Bisnis

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2023/2024**

**PERCOBAAN 1**

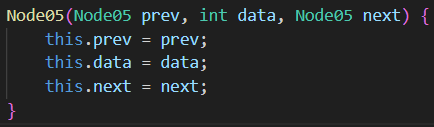
1. Buat class di dalam paket tersebut dengan nama Node05

****

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.

****

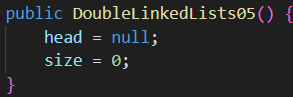
1. Buatlah sebuah class baru bernama DoubleLinkedLists pada package yang sama dengan node

****

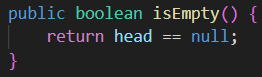
1. Pada class DoubleLinkedLists tersebut, deklarasikan atribut sesuai dengan diagram class di atas

****

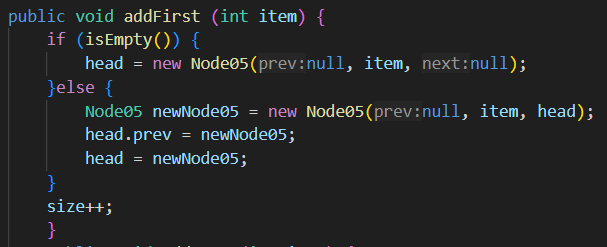
1. Selajuntnya, buat konstruktor pada class DoubleLinkedLists sesuai gambar berikut.

****

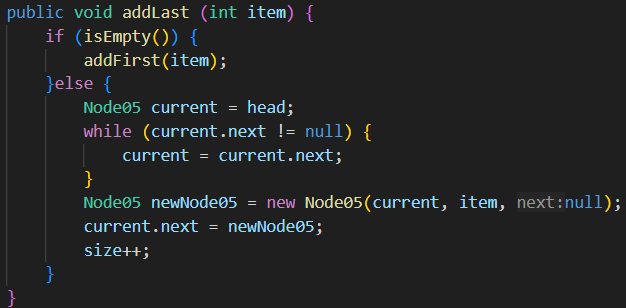
1. Buat method isEmpty(). Method ini digunakan untuk memastikan kondisi linked list kosong.

****

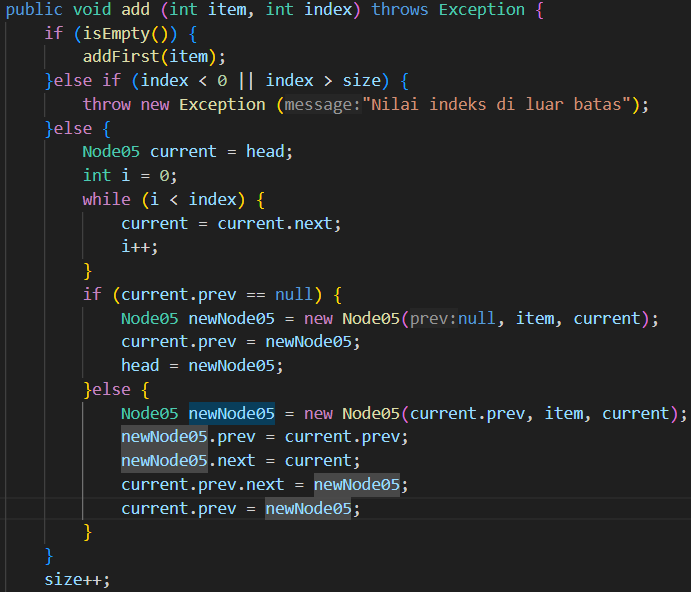
1. Kemudian, buat method addFirst(). Method ini akan menjalankan penambahan data di bagian depan linked list

****

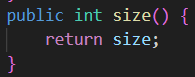
1. Selain itu pembuatan method addLast() akan menambahkan data pada bagian belakang linked list.

****

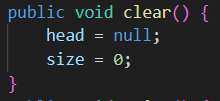
1. Untuk menambahkan data pada posisi yang telah ditentukan dengan indeks, dapat dibuat dengan method add(int item, int index)

****

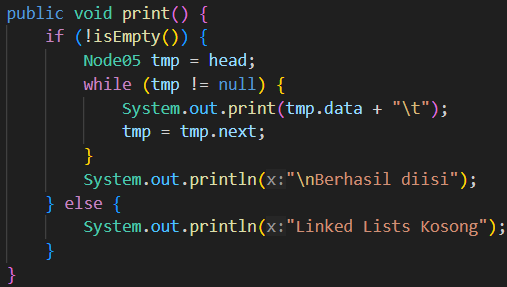
1. Jumlah data yang ada di dalam linked lists akan diperbarui secara otomatis,sehingga dapat dibuat method size() untuk mendapatkan nilai dari size.

****

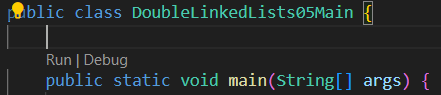
1. Selanjutnya dibuat method clear() untuk menghapus semua isi linked lists, sehingga linked lists dalam kondisi kosong

****

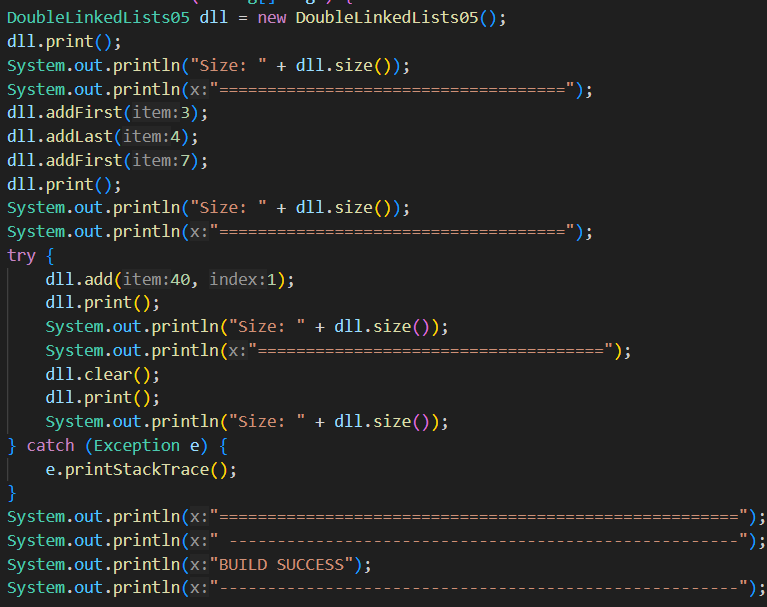
1. Untuk mencetak isi dari linked lists dibuat method print(). Method ini akan mencetak isi linked lists berapapun size-nya. Jika kosong akan dimunculkan suatu pemberitahuan bahwa linked lists dalam kondisi kosong.

****

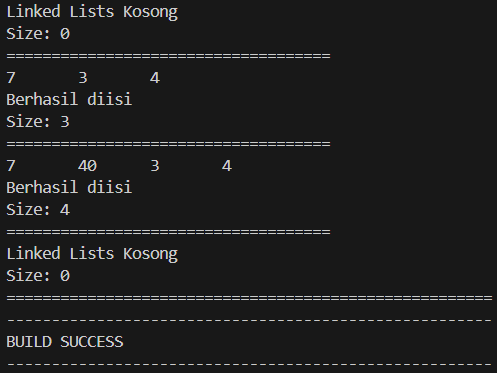
1. Selanjutya dibuat class Main DoubleLinkedListsMain untuk mengeksekusi semua method yang ada pada class DoubleLinkedLists.

****

1. Pada main class pada langkah 14 di atas buatlah object dari class DoubleLinkedLists kemudian eksekusi potongan program berikut ini.

****

**HASIL RUN:**

****

**Pertanyaan Percobaan**

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

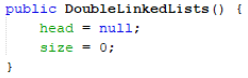
**Jawaban**: **Single Linked List:** Setiap node memiliki dua bagian: data dan pointer "next", Pointer "next" menunjuk ke node berikutnya dalam list.

**Double Linked List:** Setiap node memiliki tiga bagian: data, pointer "next", dan pointer "prev", Pointer "next" menunjuk ke node berikutnya.

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

**Jawaban:** Atribut next dan prev tersebut berfungsi sebagai pointer atau penghubung antar node.

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

****

**Jawaban:** head = null : menandakan bahwa awalnya linked lists tidak memiliki node apapun dan memberi tahu bahwa linked list kosong saat objek 'DoubleLinkedLists' pertama kali dibuat. size = 0 : menendakan bahwa ukuran (jumlah node) linked list awalnya 0 yang mengindikasikan bahwa tidak ada elemen dalam linked list. sehingga kesimpulannya konstruktor 'DoubleLinkedLists' mempersiapkan linked list kosong yang dapat digunakan untuk menyimpan dan mengelola elemen- elemen baru yang ditambahkan nanti.

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

****

**Jawaban:** Karena pada node yang diawal atau head ditandai dengan prev yang mengarah ke null. Sehingga, pada pembuatan object dari konstruktor tersebut pada Node prev akan diberi nilai null.

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

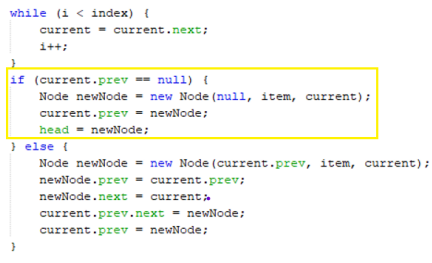
**Jawaban:** Arti statement tersebut untuk menghubungkan head yang lama dengan Node baru yang akan ditambahkan diawal yaitu newNode. Sehingga prev dari head yang awalnya bernilai null dihubungkan ke Node newNode.

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

****

**Jawaban:** Arti dari pengisian nilai Node prev pada pembuatan konstruktor tersebut adalah untuk menghubungkan Node newNode yang akan ditaruh paling belakang dengan Node current yang sementara berada dipaling belakang dan pengisian nilai Node next diisi null mengindikasikan bahwa Node tersebut akan berada di paling belakang sehingga ketika di next akan bernilai null. Nilai Node Current berada di paling belakang setelah dilakukannya traversal untuk mencari Node mana yang berada di paling belakang.

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

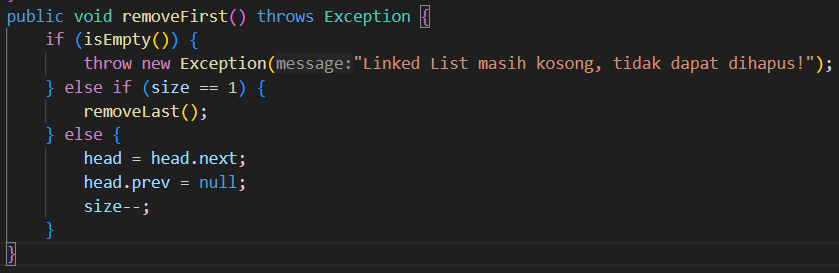
****

jelaskan maksud dari bagian yang ditandai dengan kotak kuning

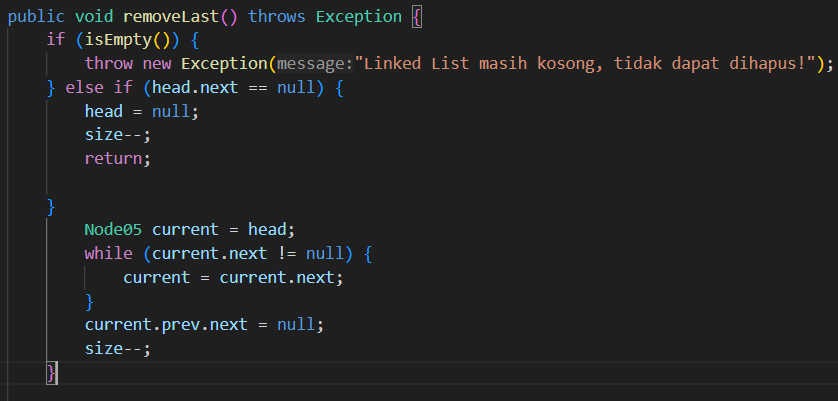
**jawaban:** Pada potongan kode yang ditandai tersebut untuk sebuah case dimana Ketika ingin menginput data yang berada paling depan atau awal. Sehingga Node newNode akan diisi konstruktor Node prev = null (mengindikasikan bahwa dia Node paling depan), dan Node next = current (untuk menghubungkan Node newNode kepada node yang sementara berada di paling depan). Lalu, Node current akan dihubungkan ke Node newNode dengan mengganti nilai Node prev pada Node current dengan newNode, sehingga kedua Node tersebut akhirnya terhubung 2 arah baik dengan menggunakan next ataupun prev. Terakhir, karena berada di paling depan maka head akan digantikan ke Node newNode.

**PERCOBAAN 2**

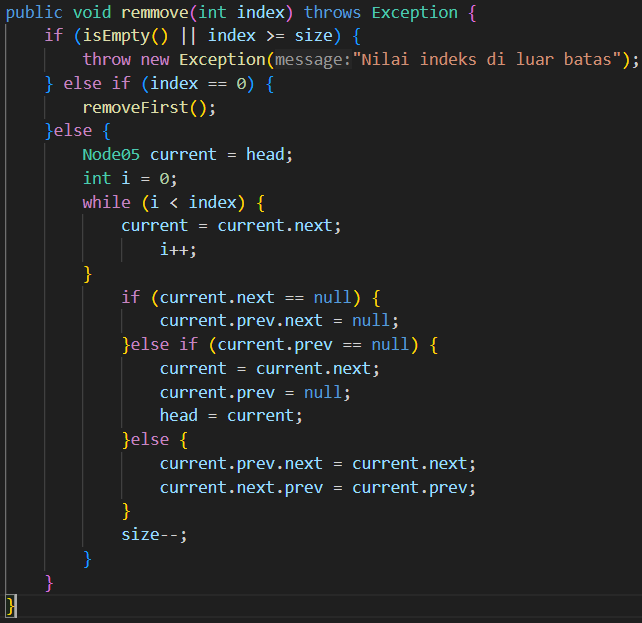
1. Buatlah method removeFirst() di dalam class DoubleLinkedLists.



1. Tambahkan method removeLast() di dalam class DoubleLinkedLists



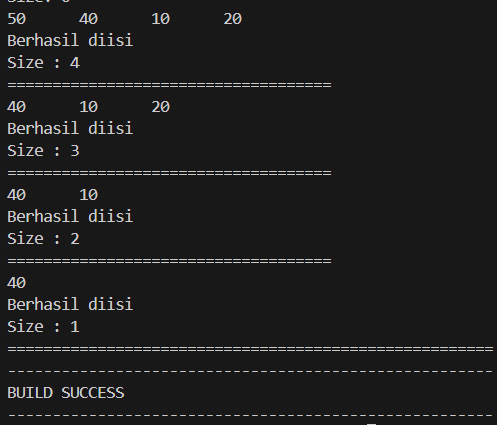
1. Tambahkan pula method remove(int index) pada class DoubleLinkedLists dan amati hasilnya



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



**HASIL RUN:**

****

**Pertanyaan percobaan**

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

****

**Jawaban:**

Maksud dari potongan kode tersebut adalah:

- head = head.next; => artinya adalah mengganti nilai head ke head.next

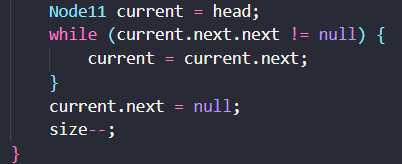
- head.prev = null; => artinya adalah merubah nilai head.prev yang tadinya

masih ada nilai Node nya menjadi null untuk memutus pointer ke Node yang

paling awal sebelumnya.

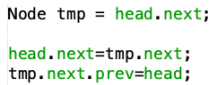
1. Bagaimana cara mendeteksi posisi data ada pada bagian akhir pada method removeLast()?

**Jawaban:** Untuk mendeteksi posisi data pada bagian akhir ditandai dengan potongan kode berikut

******

Pada potongan kode diatas menjelaskan akan dilakukan traversal dimulai dari head hingga Node current.next.next bernilai null. Sehingga bisa dilakukan remove atau penghapusan pada node terakhir ditandai dengan ‘current.next = null’.

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

****

**Jawaban:** Potongan program tersebut tidak coock karena potongan program tersebut hanya

menghapus node kedua dalam linked list, bukan node yang sebenarnya ingin

dihapus. berikut penjelasannya:

1) Node tmp = head.next; => baris ini menginisialisasi tmp dengan node kedua

dalam linked list, ditandai dengan adanya 'head.next'

2) head.next = tmp.next => baris ini memperbarui nilai head.next untuk melompati

tmp. ini menghapus koneksi dari head ke tmp dan langsung menghubungkan head

ke node setelah tmp.

3) tmp.next.prev = head => baris ini memperbarui nilai prev dari node setelah tmp

(node ke-3) untuk mentarget head. namun, jika tmp adalah node kedua, maka

tmp.next akan menjadi null dan ketika mengakses prev dari null akan terjadi

'NullPointerException' atau data tidak ada.

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

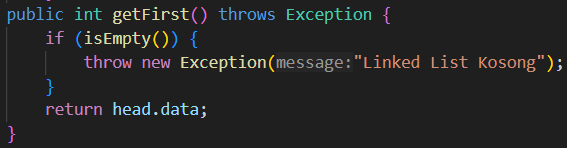
****

**Jawaban:** Fungsi kode program tersebut adalah untuk menghapus node yang akan kita

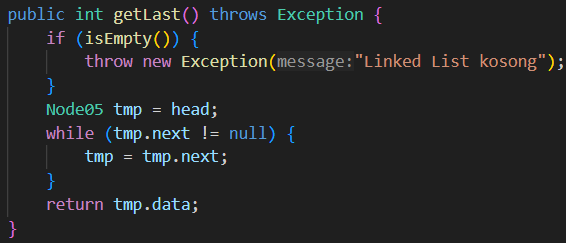
target. menggambarkan fungsi dari potongan program tersebut adalah untuk menghubungkan node sebelum dan sesudah dari Node yang telah kita target untuk dihapus. Sehingga setelah node tersebut dihapus maka Node sebelum dan sesudah nya akan tetap masih terhubung. Seperti ilustrasi berikut.

**PERCOBAAN 3**

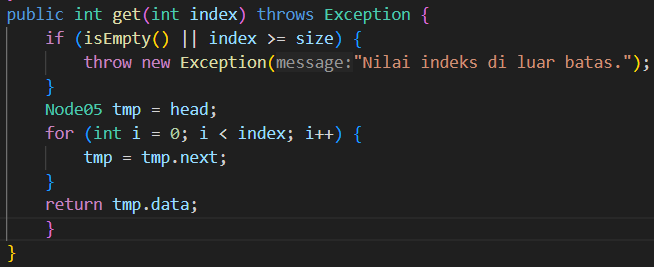
1. Buatlah method getFirst() di dalam class DoubleLinkedLists untuk mendapatkan data pada awal linked lists.



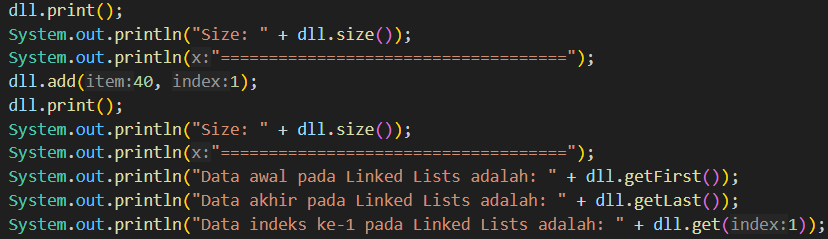
1. Selanjutnya, buatlah method getLast() untuk mendapat data pada akhir linked lists.



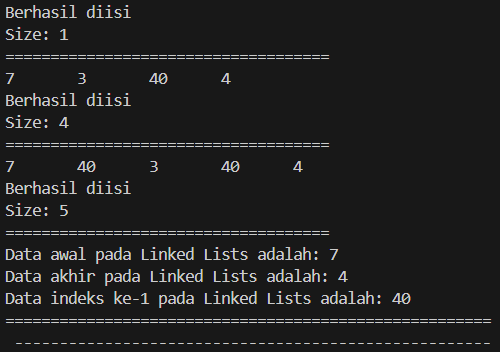
1. Method get(int index) dibuat untuk mendapatkan data pada indeks tertentu



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



**HASIL RUN:**

****

**Pertanyaan Percobaan**

1. Jelaskan method size() pada class DoubleLinkedLists!

**Jawaban:** Method size digunakan mengembalikan nilai dari atribut size untuk memberitahukan berapa jumlah data pada linked lists.

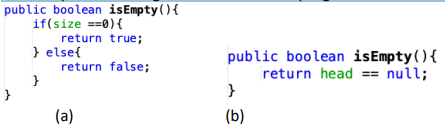
1. Jelaskan cara mengatur indeks pada double linked lists supaya dapat dimulai dari indeks ke1!

**Jawaban:** Pertama adalah mengatur pada konstruktor double linked lists itu sendiri di set menjadi bernilai size = 1; Kedua, setiap memulai proses traversal langsung mengarahkan nilai indeks nya adalah satu. Sehingga proses perhitungan indeks akan dimulai dari satu.

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

**Jawaban:** Perbedaan karakteristik nya adalah pada double linked lists diperlukan parameter Node Next dan Prev. Sedangkan pada single linked list hanya memerlukan parameter Next saja.

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

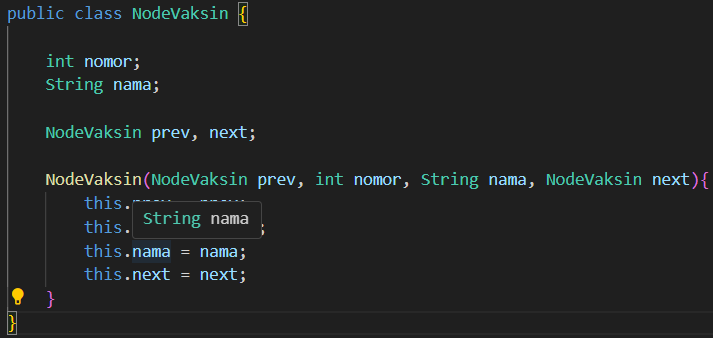
****

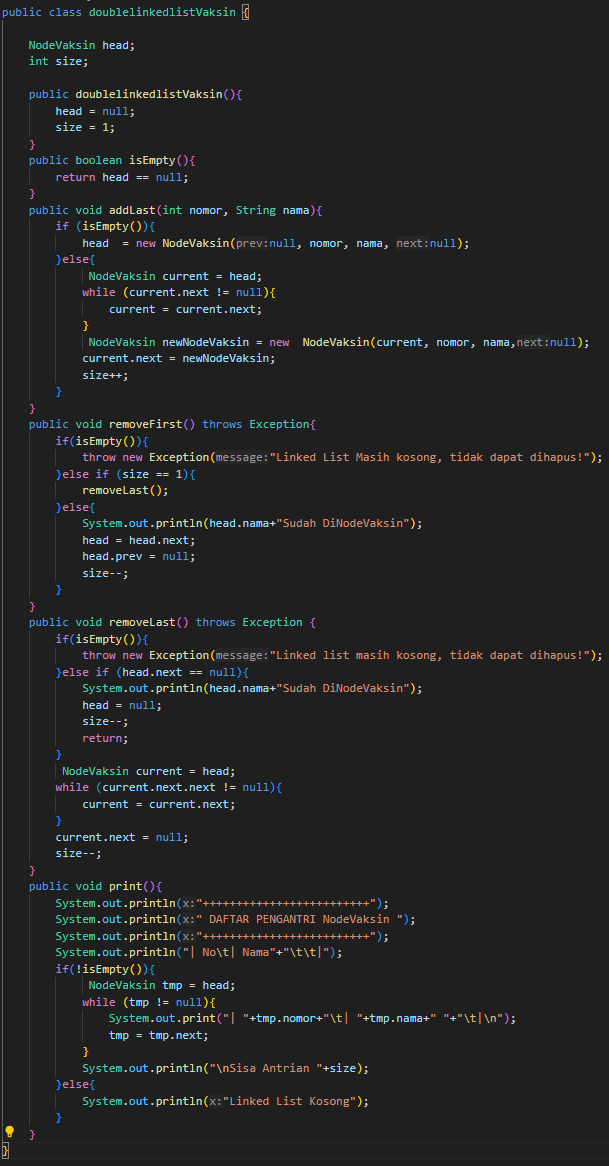
**Jawaban:**

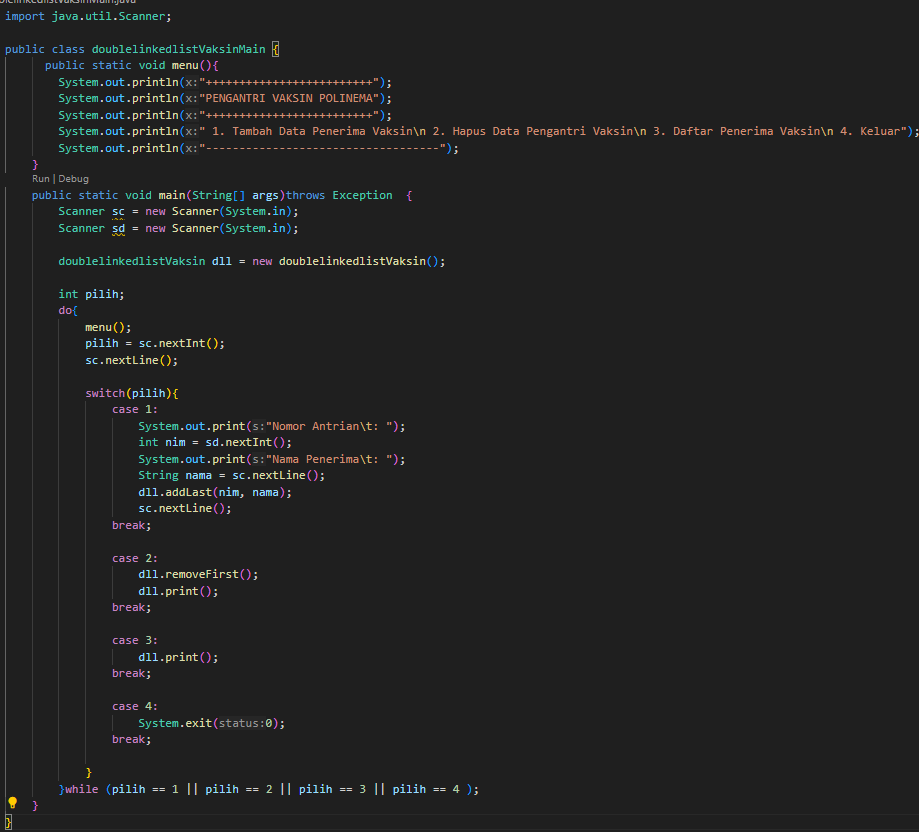
Perbedaan kedua logika tersebut berdasarkan apa yang menjadi perhitungannya. Pada point (a) kondisinya adalah menghitung apakah size = 0. Jika iya maka akan menghasilkan nilai true, jika tidak maka akan menghasilkan nilai false. Sedangkan pada point (b) kondisinya akan melihat apakah nilai head == null. Jika iya maka akan menghasilkan nilai true. Pada point (b) tidak terlihat secara tertulis seperti pada point (b) untuk memberitahukan kondisi apa yang menghasilkan nilai true ataupun false.

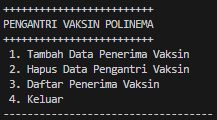
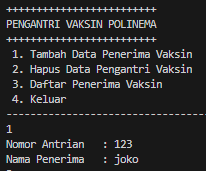
**TUGAS**

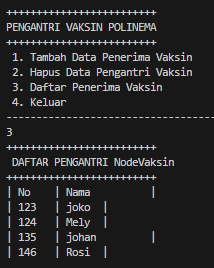
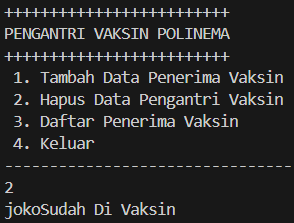
1. Hasil run:







1. Hasil Run:

