Nama : Deanissa Sherly Sabilla

Kelas / Absen : 1B SIB / 06

JOBSHEET IX

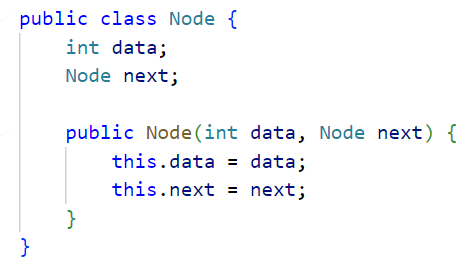
LINKED LIST

2.1 Pembuatan Linked List

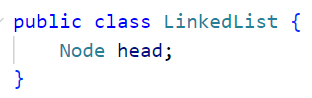
**Waktu percobaan: 50 menit**

Didalam praktikum ini, akan dilakukan implementasi pembuatan linked list menggunakan array dan penambahan node ke dalam linked list

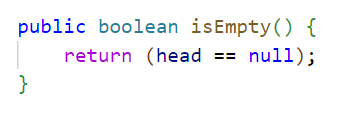
1. Buat folder baru Praktikum09
2. Tambahkan class-class berikut:
   1. Node.java
   2. LinkedList.java
   3. SLLMain.java
3. Deklarasikan class Node yang memiliki atribut data untuk menyimpan elemen dan atribut next bertipe Node untuk menyimpan node berikutnya. Tambahkan constructor berparameter untuk mempermudah inisialisasi



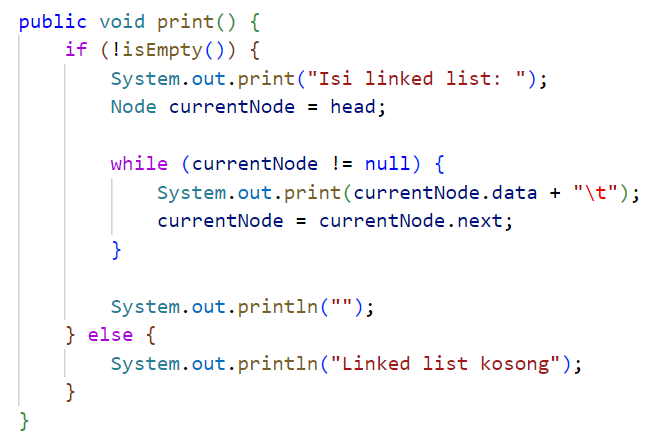
1. Deklarasikan class LinkedList yang memiliki atribut head. Atribut head menyimpan node pertama pada linked list



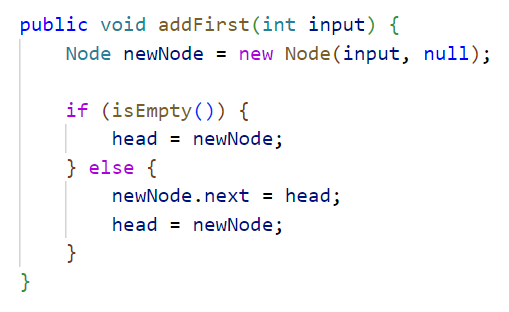
1. Sebagai langkah berikutnya, akan diimplementasikan method-method yang terdapat pada class LinkedList.
2. Tambahkan method **isEmpty()**



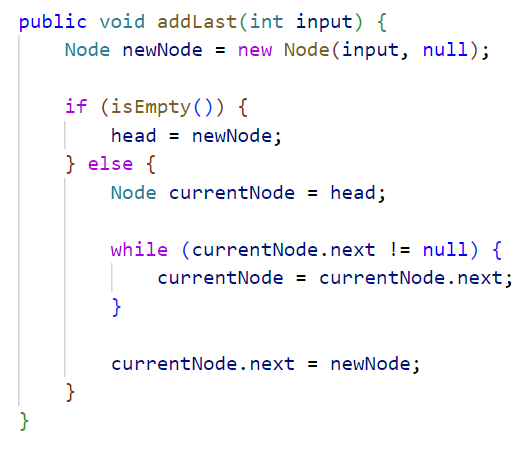
1. Implementasi method print() untuk mencetak dengan menggunakan proses traverse.



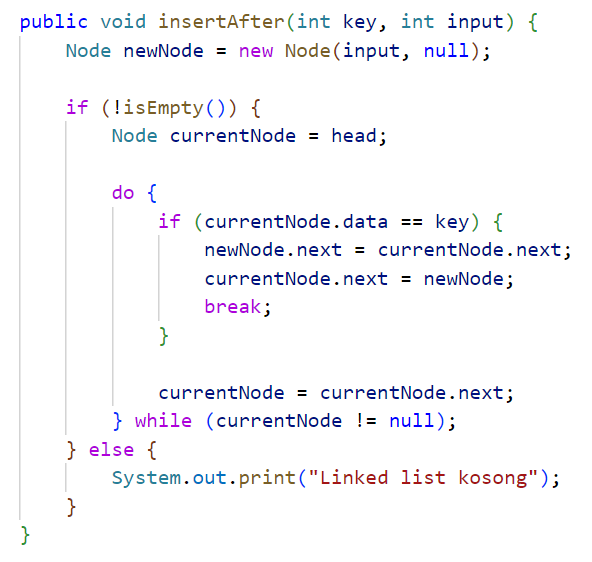
1. Implementasikan method **addFirst()** untuk menambahkan node baru di awal linked list



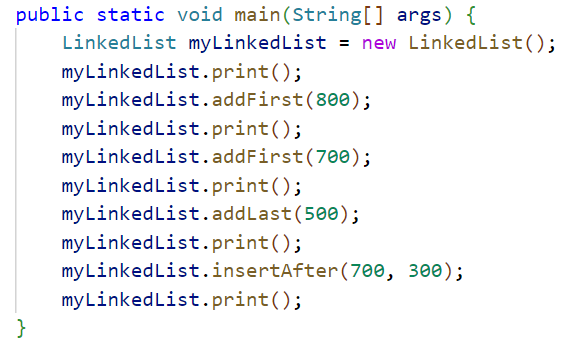
1. Implementasikan method **addLast()** untuk menambahkan node baru di akhir linked list



1. Implementasikan method **insertAfter()** menambahkan node baru pada posisi setelah node yang berisi data tertentu (key)

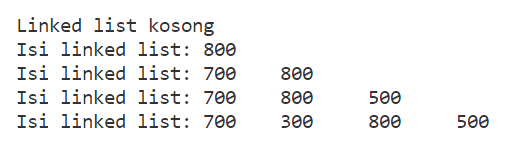


1. Pada class SLLMain, buatlah fungsi **main**, kemudian buat object myLinkedList bertipe LinkedList. Lakukan penambahan beberapa data. Untuk melihat efeknya terhadap object myLinkedList, panggil method print()



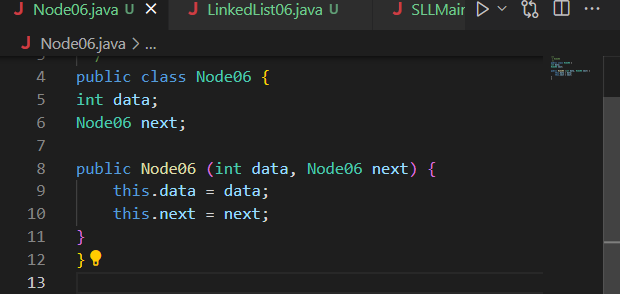
1. Verifikasi Hasil Percobaan

Cocokkan hasil run program Anda dengan output berikut ini.

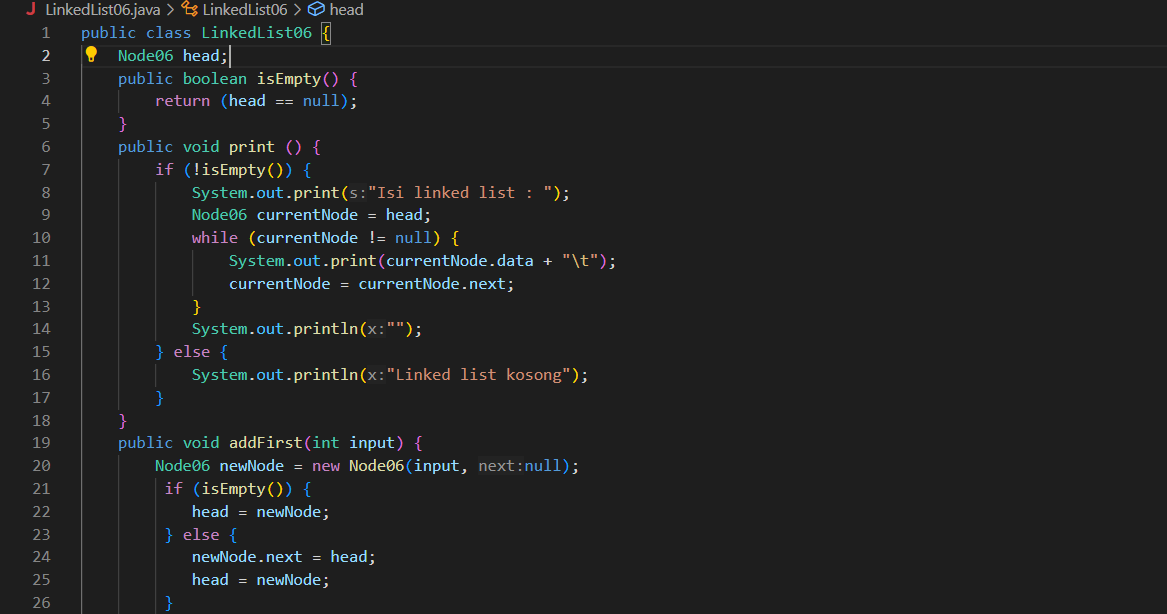


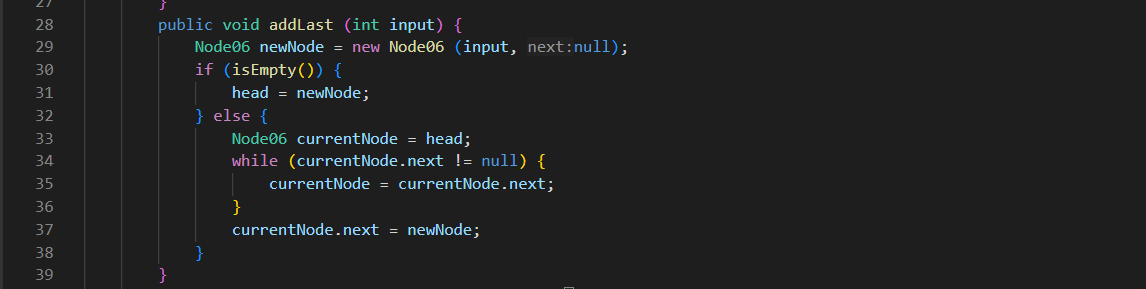
* **INPUT :**

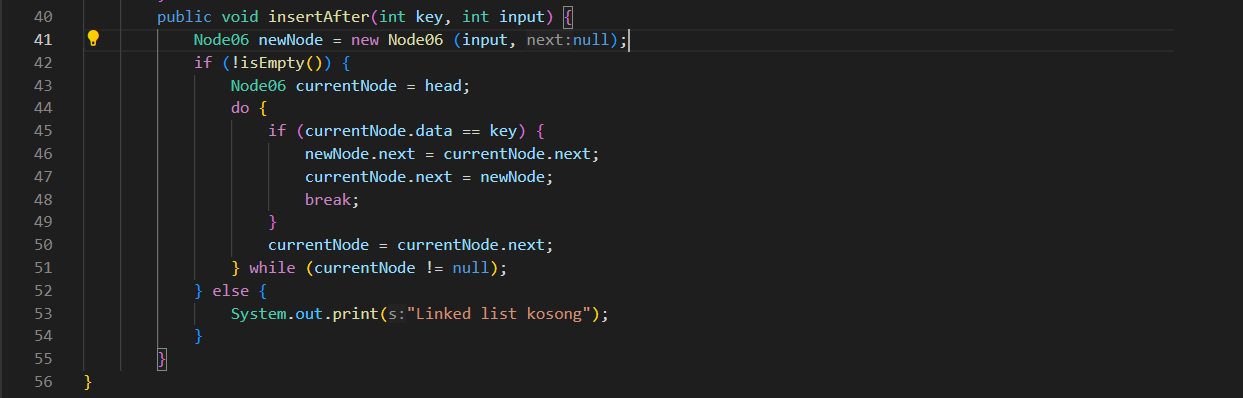
**Class Node**



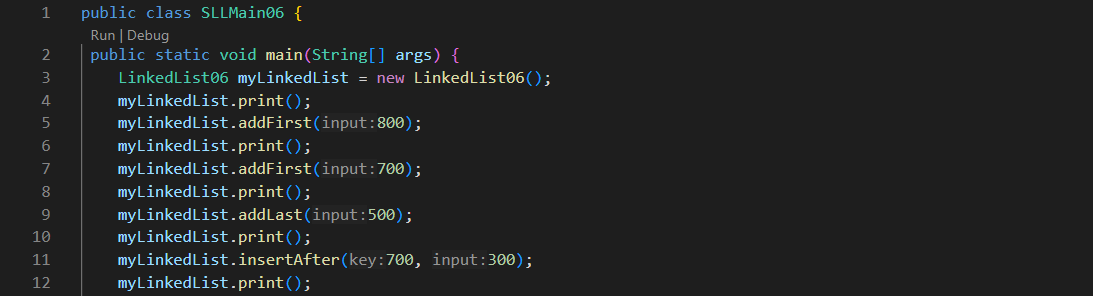
**Class LinkedList**



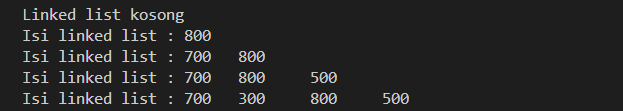




**Class Main**

****

* **OUTPUT :**



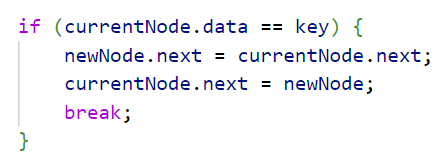
1. Pertanyaan
2. Mengapa class LinkedList tidak memerlukan method isFull() seperti halnya Stack dan Queue?

* class LinkedList tidak memerlukan method isFull() karena LinkedList biasanya tidak memiliki batasan maksimum pada jumlah elemen yang dapat disimpan di dalamnya, seperti pada Stack atau Queue yang memiliki ukuran terbatas (terbatas oleh ukuran array yang digunakan dalam implementasinya).

1. Mengapa class LinkedList hanya memiliki atribut head yang menyimpan informasi node pertama? Bagaimana informasi node kedua dan lainnya diakses?

* karena node selanjutnya dapat diakses secara berurutan dari node pertama menggunakan referensi next yang disimpan dalam setiap node.

1. Pada langkah, jelaskan kegunaan kode berikut



* if (currentNode.data == key) kondisi yang mengecek apakah data dari node saat ini (currentNode.data) sama dengan nilai kunci (key).
* newNode.next = currentNode.next; mengatur referensi next dari newNode agar menunjuk ke node yang sebelumnya ditunjuk oleh currentNode.next.
* currentNode.next = newNode; menyisipkan newNode setelah currentNode dengan mengatur referensi next dari currentNode untuk menunjuk ke newNode.
* break; setelah menemukan tempat di mana newNode harus disisipkan, keluar dari loop dengan menggunakan pernyataan break;

1. Implementasikan method insertAt(int index, int key) dari tugas mata kuliah ASD (Teori)

* Input dari implementai method insertAt (int index, int key)

-

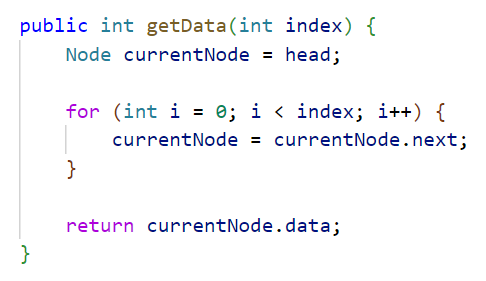
2.2 Mengakses dan menghapus node pada Linked List

**Waktu percobaan: 50 menit**

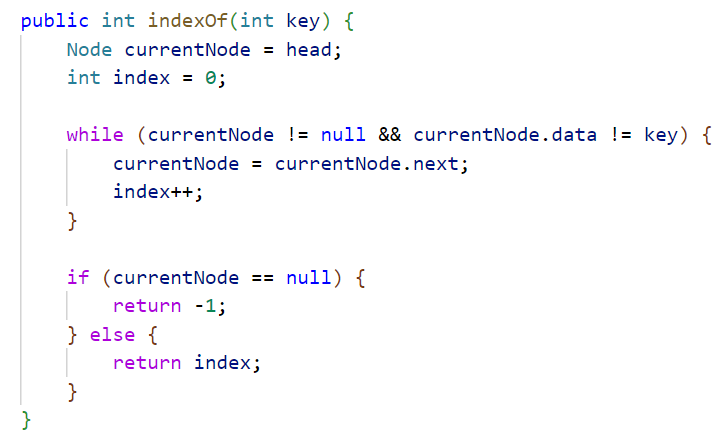
Didalam praktikum ini, kita akan mengimplementasikan method untuk melakukan pengaksesan dan penghapusan data pada linked list

2.2.1 Langkah-langkah Percobaan

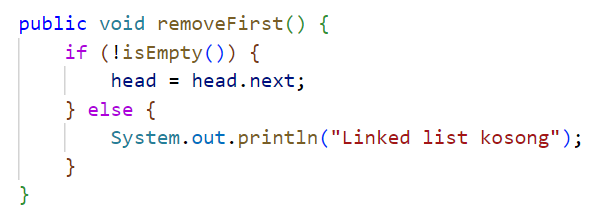
1. Tambahkan method getData() untuk mengembalikan nilai elemen di dalam node pada index tertentu



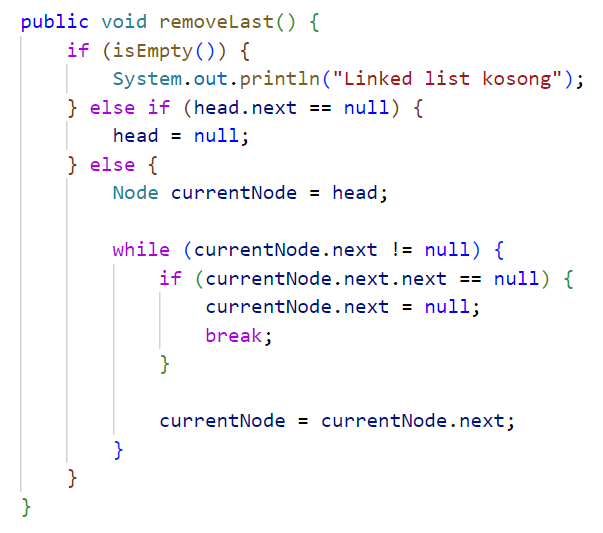
1. Tambahkan method indexOf() untuk mengetahu index dari node dengan elemen tertentu



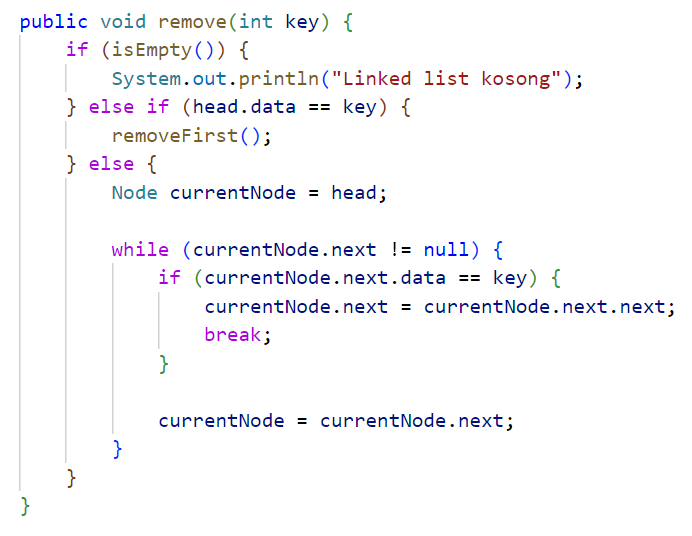
1. Tambahkan method removeFirst() untuk menghapus node pertama pada linked list



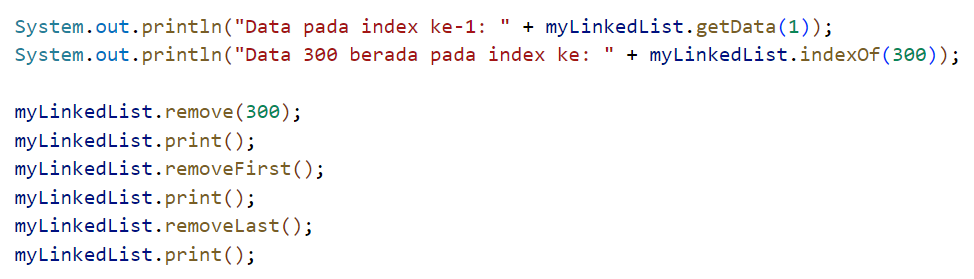
1. Tambahkan method removeLast() untuk menghapus node terakhir pada linked list



1. Method remove() digunakan untuk mengapus node yang berisi elemen tertentu

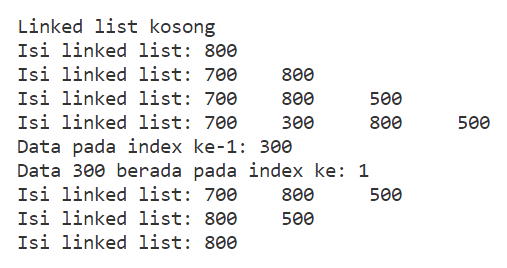


1. Kemudian, coba lakukan pengaksesan dan penghapusan data di method main pada class SLLMain dengan menambahkan kode berikut



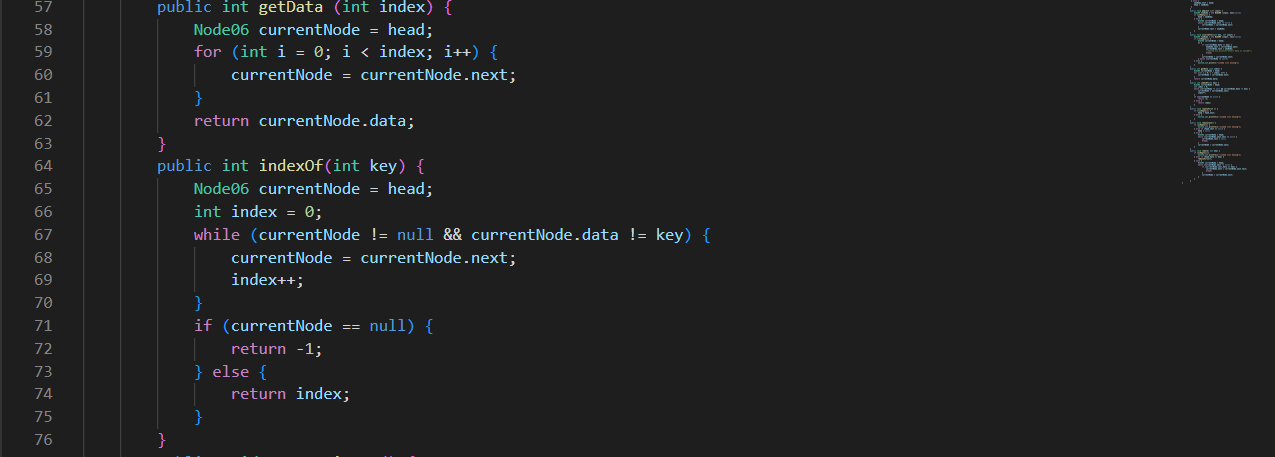
1. Compile dan run program kemudian amati hasilnya
   * 1. Verifikasi Hasil Percobaan

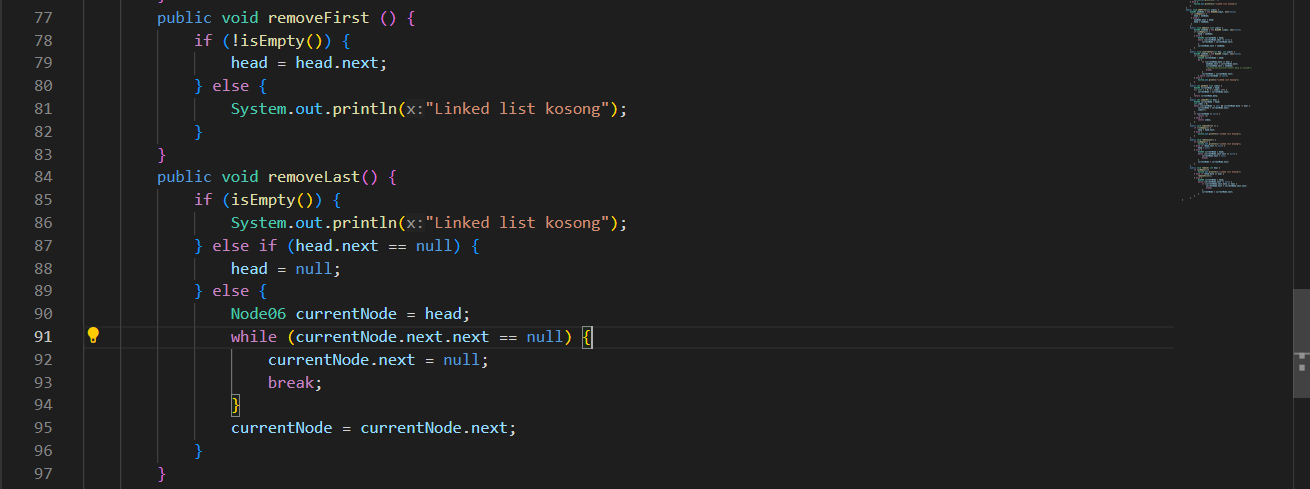
Cocokkan hasil run program dengan output berikut ini.



* **INPUT**

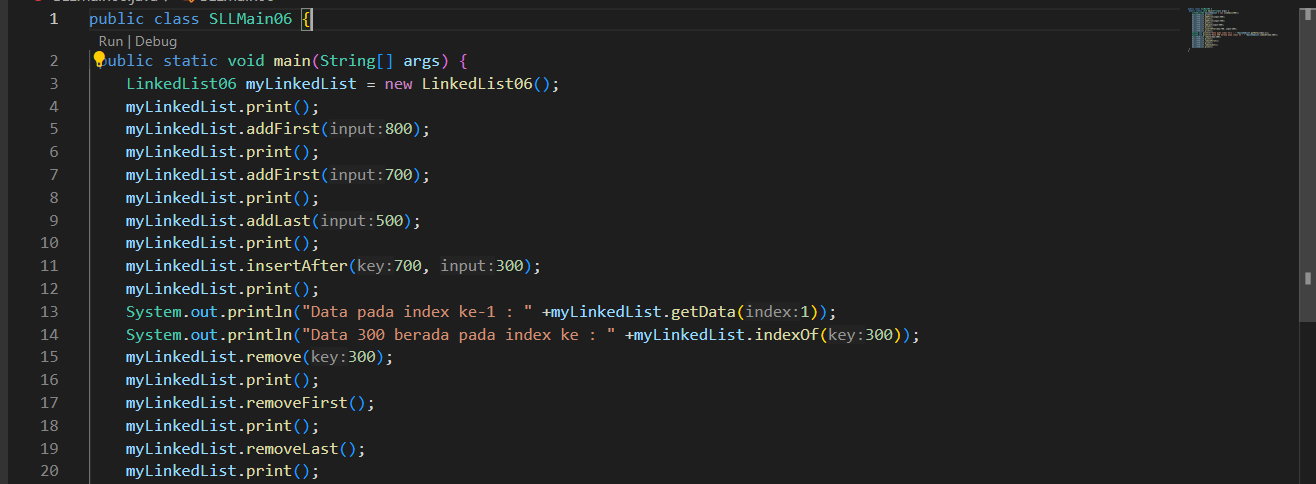
**Class LinkedList (kode tambahan)**

****

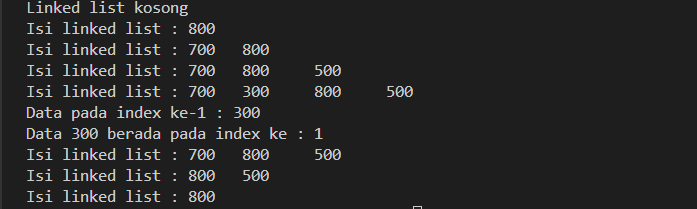
****

****

**Class Main (kode tambahan)**

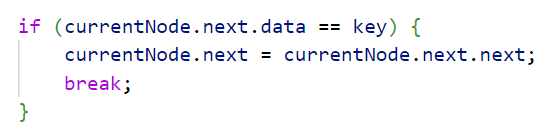
****

* **OUTPUT**

****

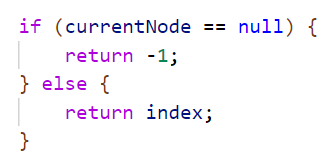
* + 1. Pertanyaan

1. Jelaskan maksud potongan kode di bawah pada method remove()



* Potongan kode tersebut untuk menghapus node pertama dari LinkedList jika nilainya sama dengan nilai kunci (key) yang diberikan.

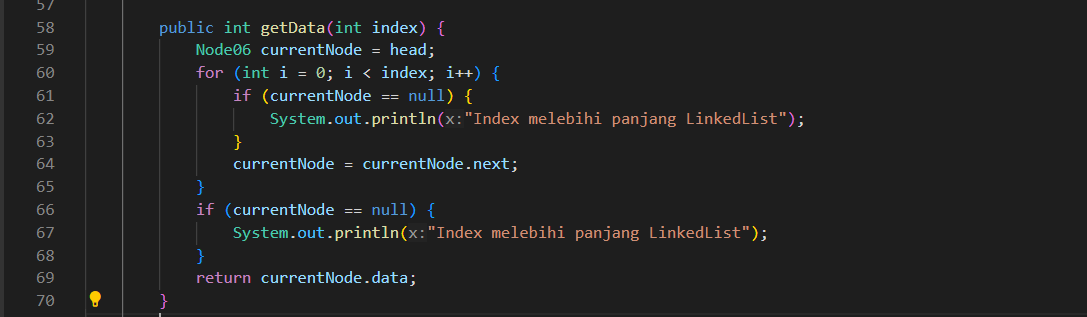
1. Jelaskan maksud if-else block pada method indexOf() berikut



* Potongan kode tersebut untuk untuk menentukan nilai yang akan dikembalikan berdasarkan apakah elemen yang dicari ditemukan atau tidak dalam LinkedList.

1. Error apa yang muncul jika argumen method getData() lebih besar dari jumlah node pada linked list? Modifikasi kode program untuk menghandle hal tersebut.

* Jika argumen yang diberikan lebih besar dari jumlah node pada LinkedList, maka akan muncul error NullPointerException.
* Modifikasi :



1. Apa fungsi keyword break pada method remove()? Bagaimana efeknya jika baris tersebut dihapus?

* untuk menghentikan iterasi melalui linked list setelah operasi penghapusan dilakukan.

3. Tugas

**Waktu pengerjaan: 50 menit**

1. Implementasikan method-method berikut pada class LinkedList:
2. insertBefore() untuk menambahkan node sebelum keyword yang diinginkan
3. insertAt(int index, int key) untuk menambahkan node pada index tertentu
4. removeAt(int index) untuk menghapus node pada index tertentu
5. Dalam suatu game scavenger hunt, terdapat beberapa point yang harus dilalui peserta untuk menemukan harta karun. Setiap point memiliki soal yang harus dijawab, kunci jawaban, dan pointer ke point selanjutnya. Buatlah implementasi game tersebut dengan linked list.