**Pertanyaan praktikum 1**

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

Karena struktur data linked list tidak memiliki batasan pada jumlah elemen yang dapat disimpan. Berbeda dengan struktur data queue dan stack yang dibatasi dengan ukuran array

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

Kelas LinkedList hanya memiliki atribut head yang menyimpan informasi

tentang node pertama karena struktur data linked list dirancang agar dapat

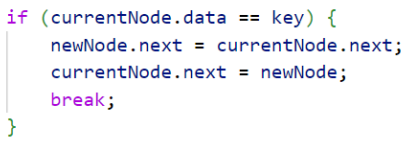
mengakses node lainnya melalui node pertama (head) dan koneksi antar node yang

disebut pointer. Ketika ingin mengakses node kedua, ketiga, atau node lainnya dalam

linked list, Anda dapat melakukannya dengan cara mengikuti pointer dari node

pertama (head).

1. Pada langkah, jelaskan kegunaan kode berikut



* **if (currentNode.data == key) :** Memeriksa apakah nilai data dari node saat ini sama

dengan nilai kunci yang dicari. Jika kondisi tersebut terpenuhi, maka langkah

selanjutnya newNode.next = currentNode.next;

• **newNode.next = currentNode.next;** : Mengatur node berikutnya dari

node baru (newNode) untuk menunjuk ke node yang saat ini ditunjuk

oleh currentNode. Dengan langkah newNode akan terhubung dengan

node yang sebelumnya ditunjuk oleh currentNode

• **currentNode.next = newNode;** : Mengatur node berikutnya dari

currentNode untuk menunjuk ke newNode. Dengan Langkah

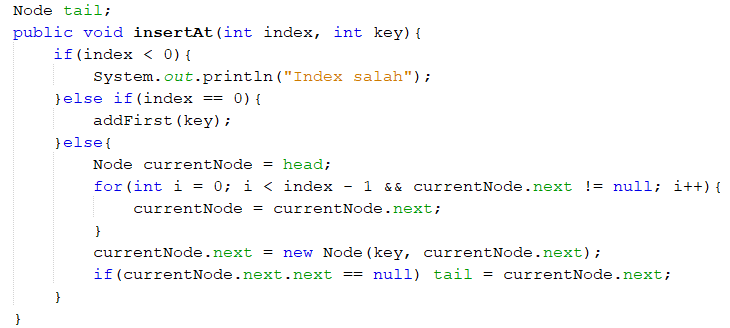
currentNode akan terhubung dengan newNode

• break; : Menghentikan atau keluar dari iterasi loop karena penyisipan

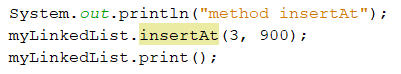
node baru telah selesai dilakukan

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

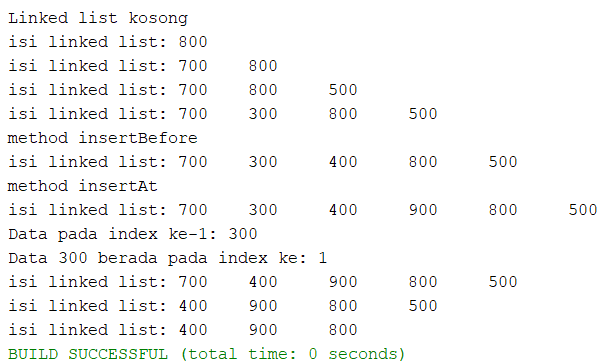
**class LinkedList**

****

**Class SLLMain**

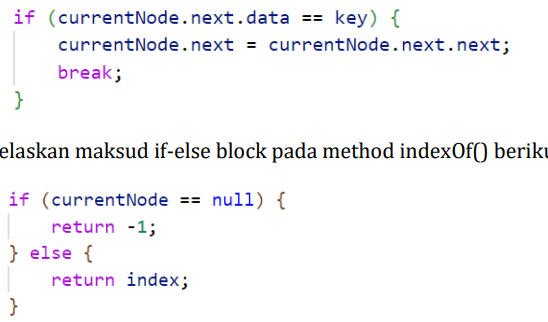
****

**Output**

****

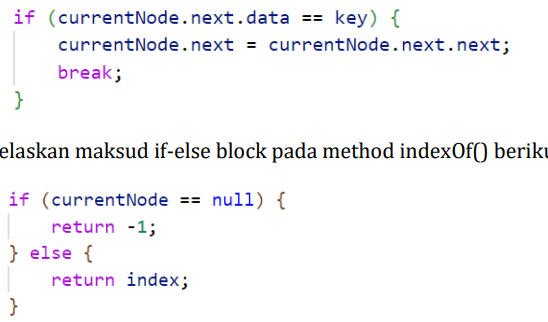
**Pertanyaan praktikum 2**

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

****

Potongan kode tersebut guna untuk menghapus node yang memiliki data sama dengan nilai key

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

****

* **if (currentNode == null)** : merupakan kondisi dimana iterasi telah

mencapai akhir linked list atau currentNode sudah menjadi null tanpa

menemukan node dengan nilai data yang sama dengan key. Key tidak

ditemukan dalam linked list, dan method mengembalikan nilai -1 untuk

menunjukkan bahwa key tidak ada dalam linked list

• **return -1;** : Jika currentNode sama dengan null, maka method akan

langsung mengembalikan -1 karena, key tidak ditemukan dalam linked

list

• **else** : bagian yang dijalankan jika kondisi if sebelumnya tidak

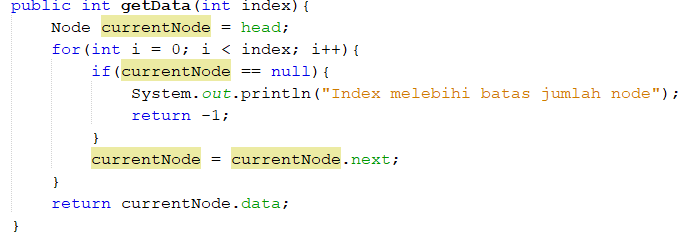
terpenuhi, artinya key ditemukan dalam linked list.

• **return index;** : pada saat key ditemukan dalam linked list, method

akan mengembalikan nilai index, yang merupakan indeks dari node

yang memiliki nilai data yang sama dengan key

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

****

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

Keyword break pada method remove( ) digunakan untuk menghentikan iterasi ketika

node yang akan dihapus telah ditemukan dan dihapus. Tanpa penggunaan ‘break’,

iterasi akan terus berlanjut bahkan setelah operasi penghapusan telah dilakukan, yang

tidak efisien dan mungkin dapat menyebabkan kesalahan dalam operasi penghapusan,

seperti menghapus node yang salah atau bahkan terjadi NullPointerException jika

mencoba mengakses node yang telah dihapus. Dengan kata lain, break digunakan

untuk keluar dari loop setelah operasi penghapusan selesai dilakukan, sehingga

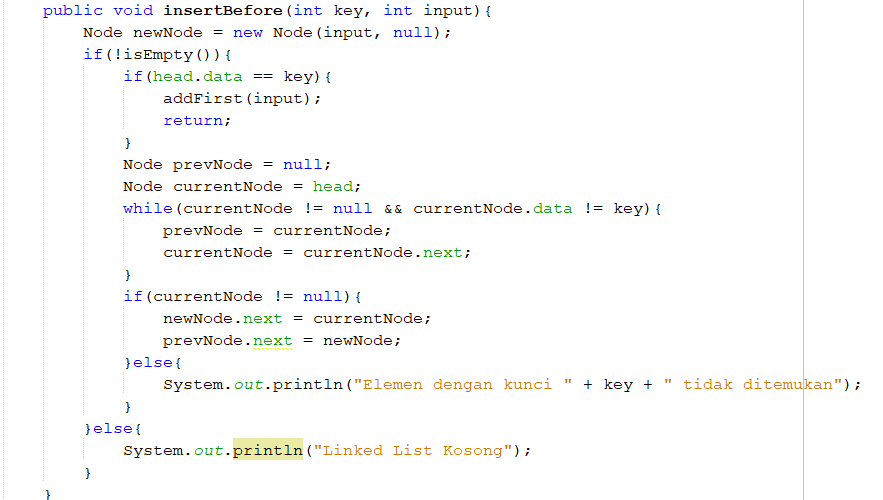
mencegah iterasi lebih lanjut yang tidak perlu.

**Tugas**

1. Implementasikan method-method berikut pada class LinkedList: a. insertBefore() untuk menambahkan node sebelum keyword yang diinginkan b. insertAt(int index, int key) untuk menambahkan node pada index tertentu c. removeAt(int index) untuk menghapus node pada index tertentu

* **insertBefore()**

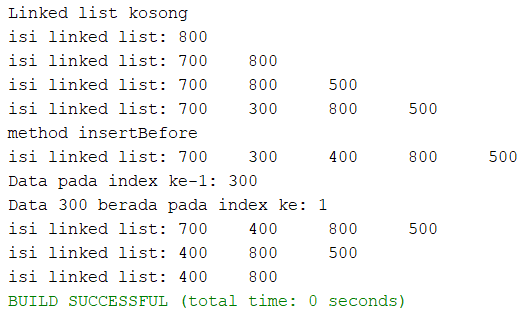
**class LinkedList**

****

**Class SLLMain**

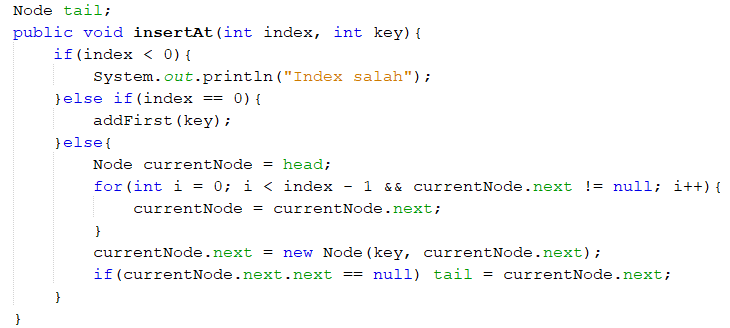
****

**Output**

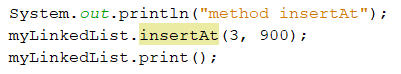
****

* **insertAt(int index, int key)**

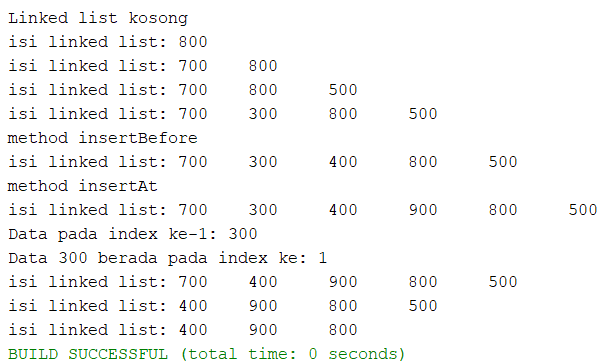
**class LinkedList**

****

**Class SLLMain**

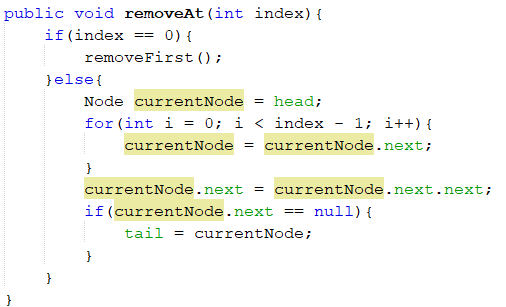
****

**Output**

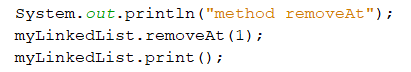
****

* **removeAt(int index)**

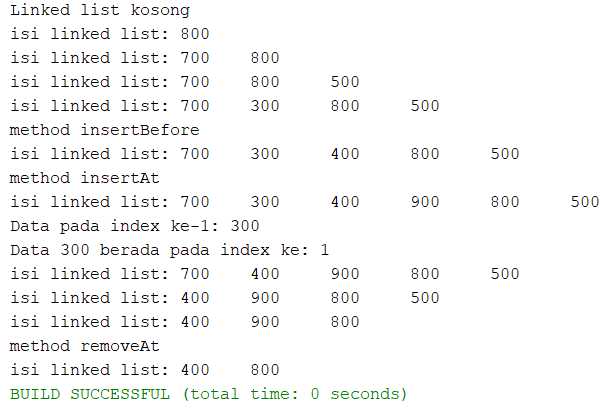
**class LinkedList**

****

**Class SLLMain**

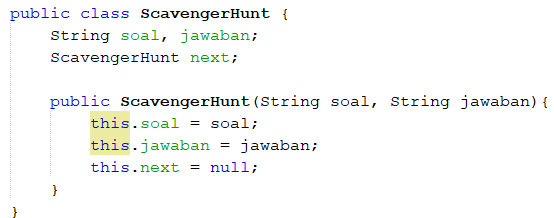
****

**Output**

****

1. 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

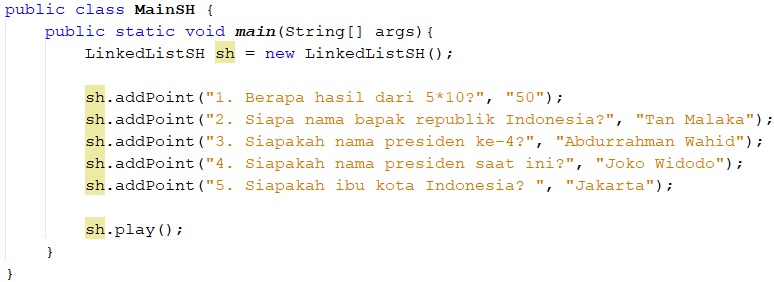
**Node class**

****

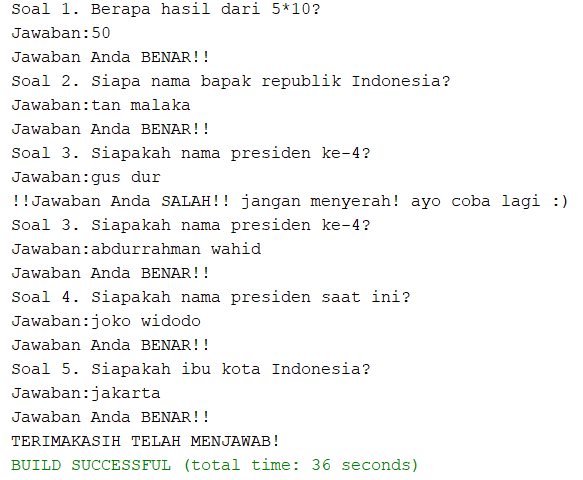
**Linked list class**

****

**Main class**

****

**Output**

****