**Laporan Praktikum ASD**

**Pertemuan 11**

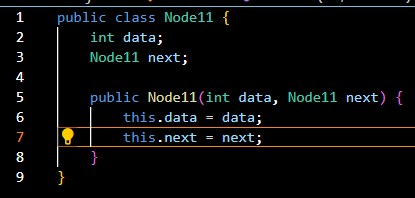
**Nama: Gegas Anugrah Derajat**

**Kelas: SIB-1F**

**NIM: 2341760140**

**Percobaan 2.1**

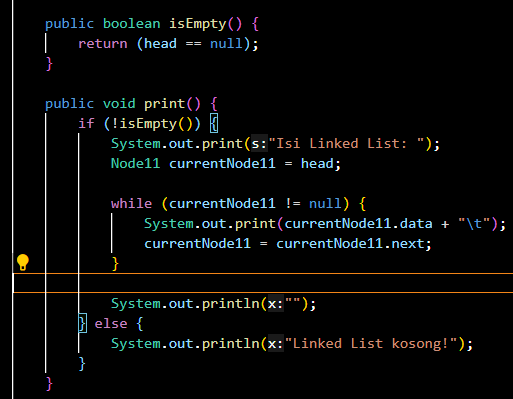
**Mendeklarasikan class node dan membuat konstruktor berparameter**

****

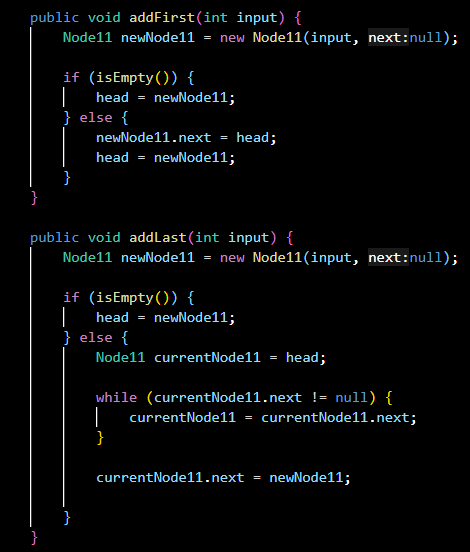
**Mendeklarasikan class LinkedList**

****

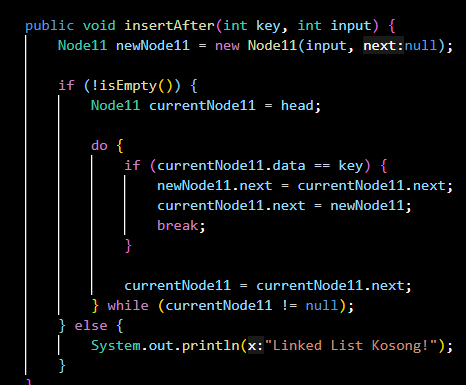
**Menambahkan method isEmpty dan print**

****

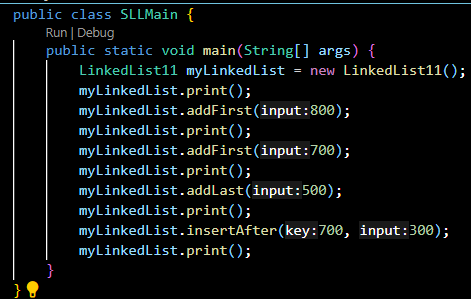
**Menambahkan method addFirst dan addLast**

****

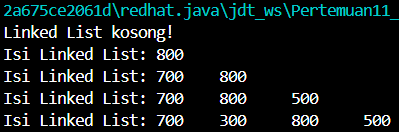
**Menambahkan method insertAfter()**

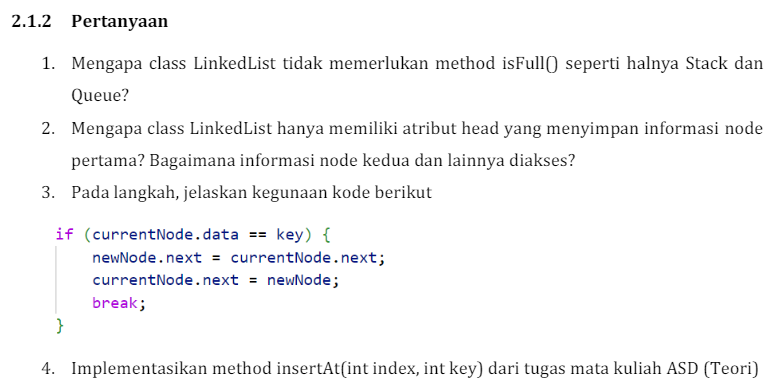
****

**Membuat fungsi SLLMain dan membuat objek mylinkedlist dan memanggil method print**

****

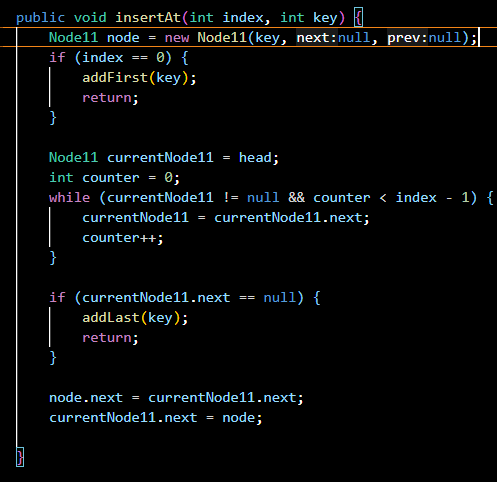
**Hasil**



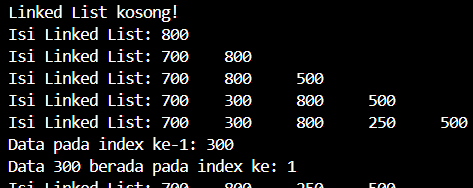


**Jawaban:**

1. **Karena pada Linked List tidak terdapat batasan kapasitas.**
2. **Linked list memulai dari head untuk menyimpan node pertama, untuk mengakses node kedua, maka akan mengikuti pointer. Untuk mengakses node berikutnya dapat mengikuti pointer yang ada pada node saat ini.**
3. **Jika data node saat ini sama dengan key, maka node baru akan disisipkan setelah key.**

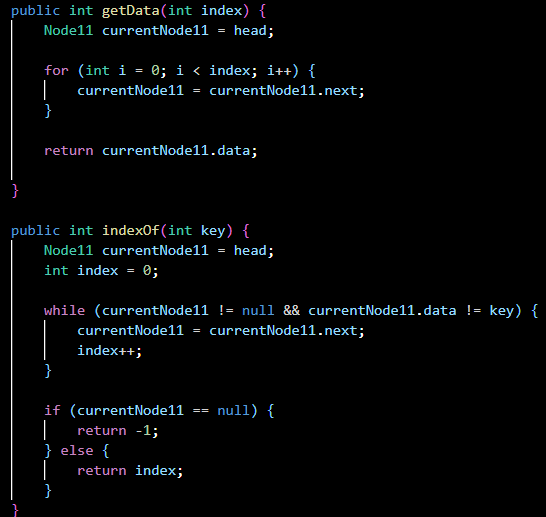




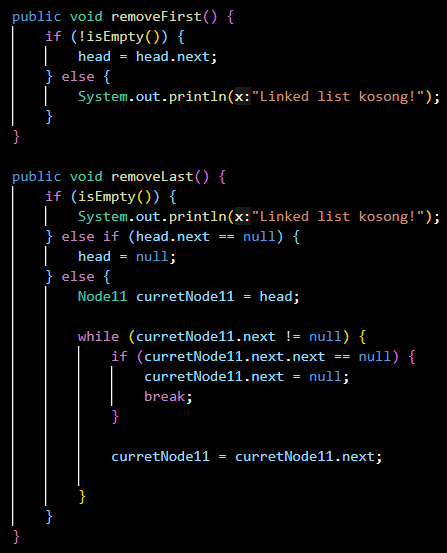


**Percobaan 2.2**

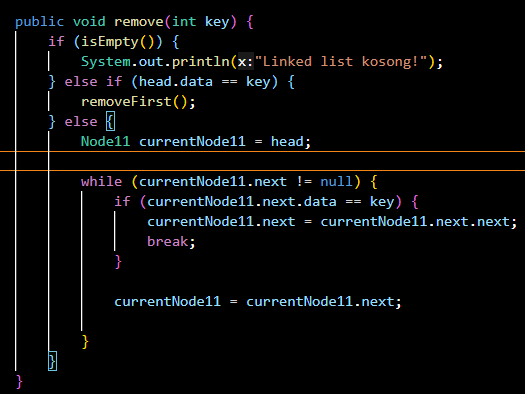
**Menambahkan method getData dan indexOf pada class LinkedList**

****

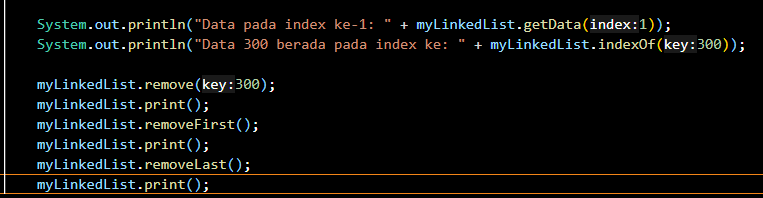
**Menambahkan method removeFirst dan removeLast pada class LinkedList**

****

**Menambahkan method remove() untuk menghapus elemen tertentu**

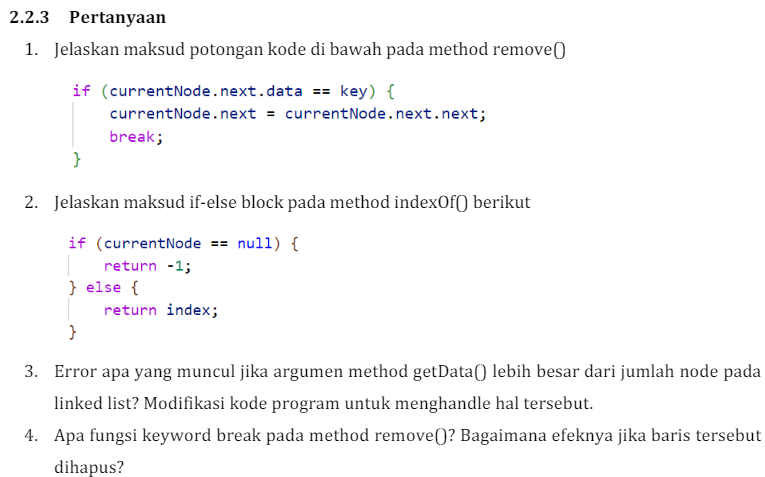
****

**Megakses dan menghapus data pada class SLLMain**

****

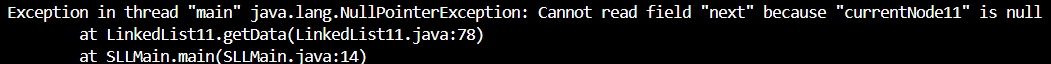
**Hasill**

****

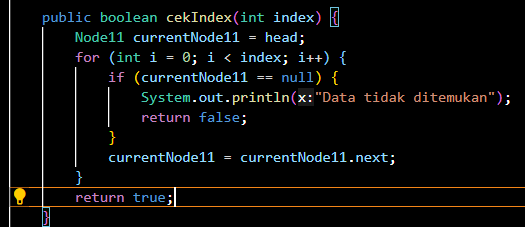


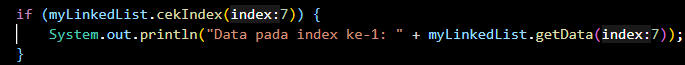
**Jawaban:**

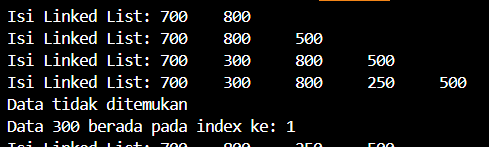
1. **Memeriksa apakah data dari node selanjutnya sama dengan key. Jika sama, maka node yang ditemukan akan dihapus.**
2. **Jika node saat ini bernilai null, maka akan mengmbalikan nilai -1. Jika tidak, maka akan mengembalikan index untuk ditmpilkan.**
3. **Error yang muncul adalah**

****

**Kode untuk menghandle**







1. **Berfungsi untuk menghentikan perulangan while jika sudah menemukan node yang ingin dihapus. jika dihapus maka akan terus berulang.**

**Tugas**

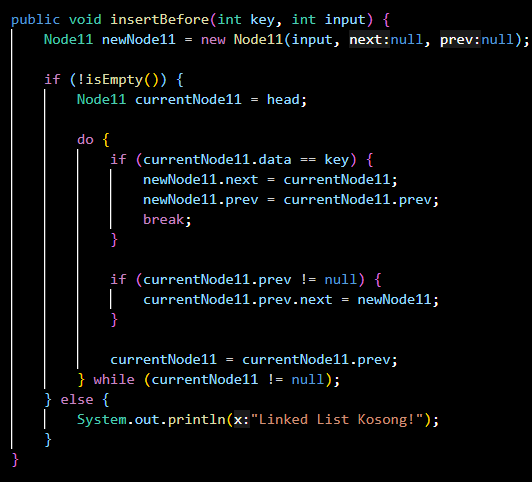
1. Implementasikan method-method berikut pada class LinkedList:
2. insertBefore() untuk menambahkan node sebelum keyword yang diinginkan
3. insertAt(int index, int key) untuk menambahkan node pada index tertentu

c. removeAt(int index) untuk menghapus node pada index tertentu

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.

**Jawaban:**

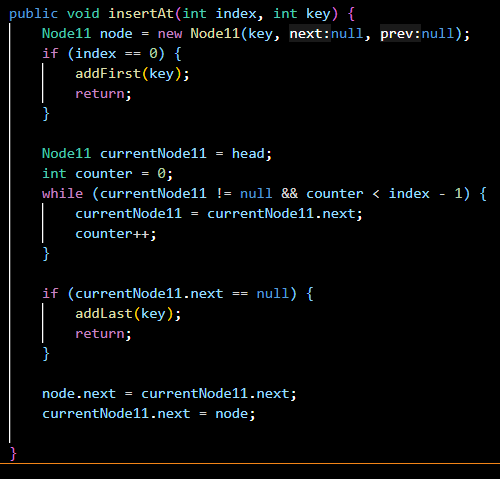
2. **insertBefore() untuk menambahkan node sebelum keyword yang diinginkan**

****

****

****

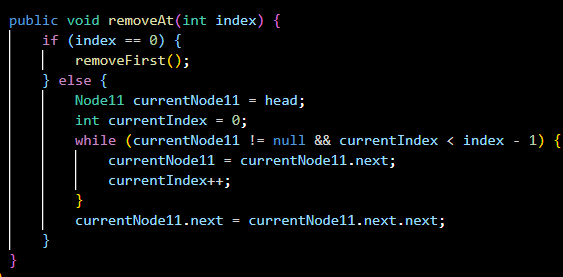
1. **insertAt(int index, int key) untuk menambahkan node pada index tertentu**

****

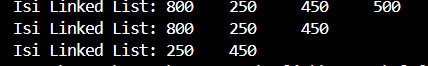
****

****

1. **removeAt(int index) untuk menghapus node pada index tertentu**

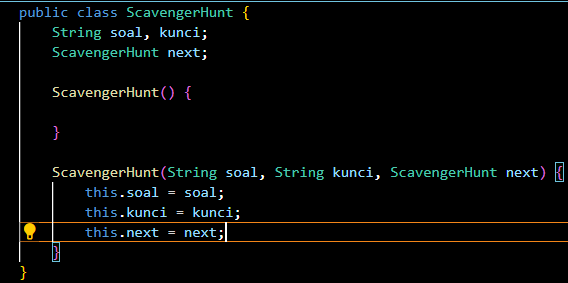
****

****

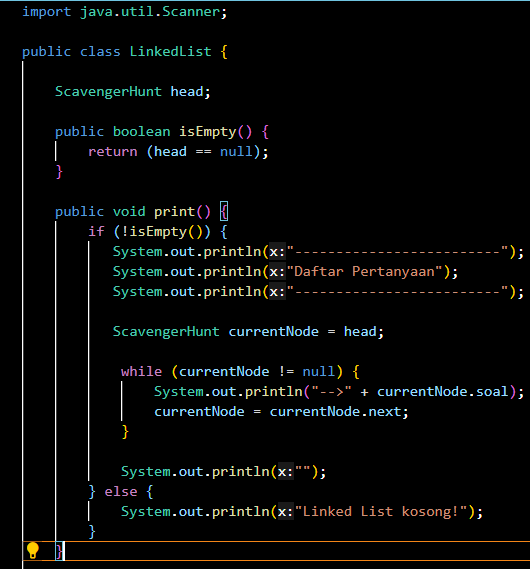
****

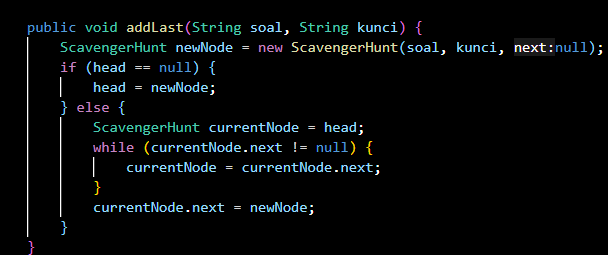


**Class Node**

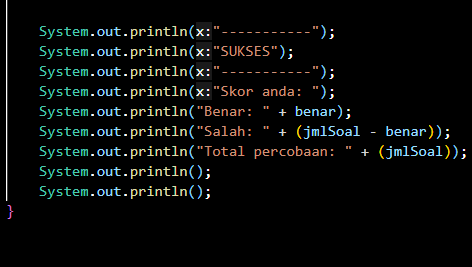
****

**Class linkedList**

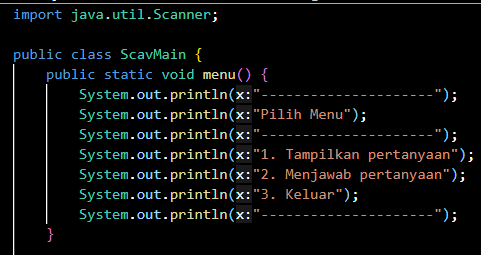
****

****

****

****

**Class Main**

****

****

**Hasil**

