**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

**PERTEMUAN KE 11 JOBSHEET 9 Linked List**

Dosen Pengajar : Triana Fatmawati, S.T., M.T.



Muhammad Afiq Firdaus

2341760189 / 21

SIB1E

**PROGRAM STUDI D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

**Praktikum 1** **Pembuatan Linked List**

Tambahkan class-class berikut:

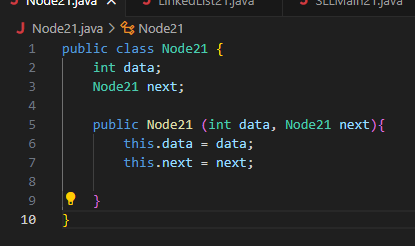
a. Node.java

b. LinkedList.java

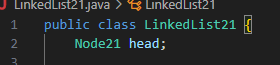
c. SLLMain.java



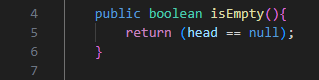
Deklarasikan class **Node** yang memiliki atribut data untuk menyimpan elemen dan atribut next bertipe Node untuk menyimpan node berikutnya. Tambahkan constructor berparameter untuk mempermudah inisialisasi



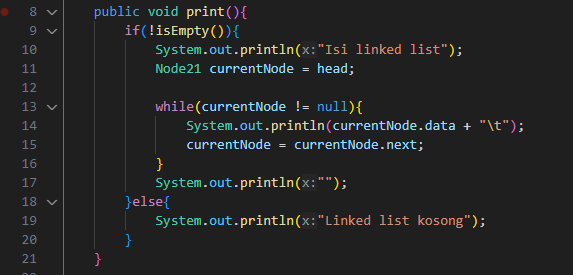
Deklarasikan class **LinkedList** yang memiliki atribut head. Atribut head menyimpan node pertama pada linked list



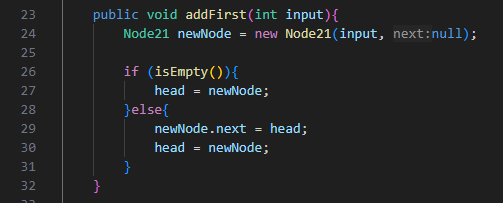
Sebagai langkah berikutnya, akan diimplementasikan method-method yang terdapat pada class LinkedList. Tambahkan method isEmpty()



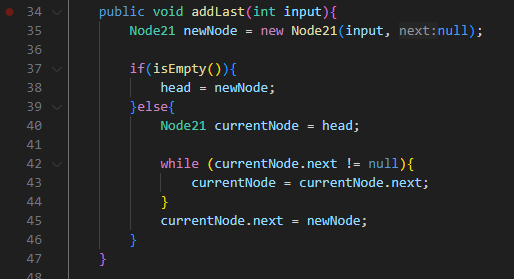
Implementasi method print() untuk mencetak dengan menggunakan proses traverse.



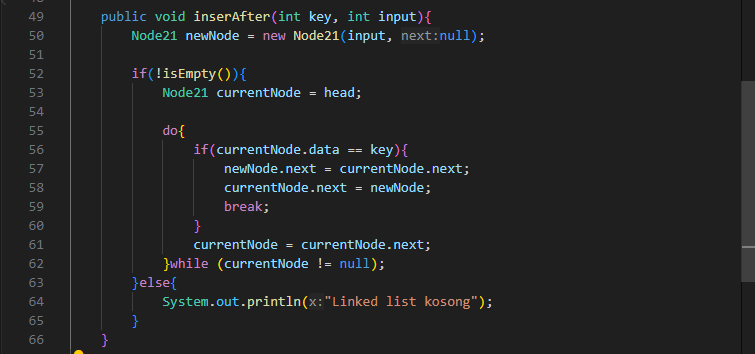
Implementasikan method addFirst() untuk menambahkan node baru di awal linked list



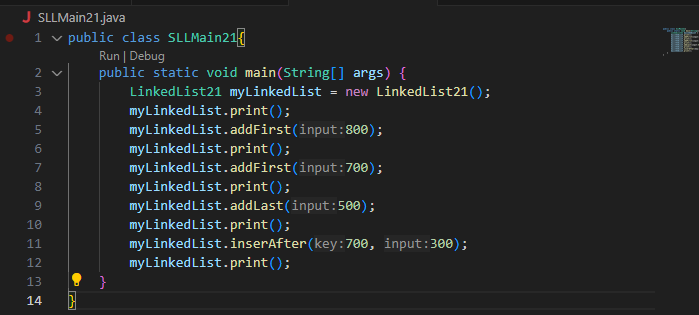
Implementasikan method addLast() untuk menambahkan node baru di akhir linked list



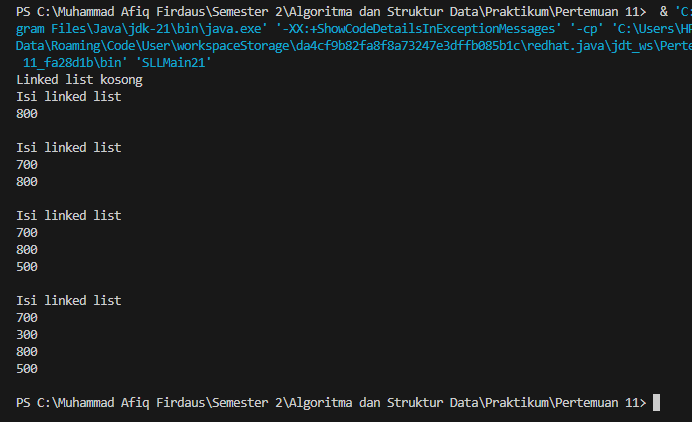
Implementasikan method insertAfter() menambahkan node baru pada posisi setelah node yang berisi data tertentu (key)



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()

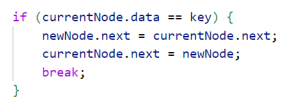


**Verifikasi Hasil Percobaan**



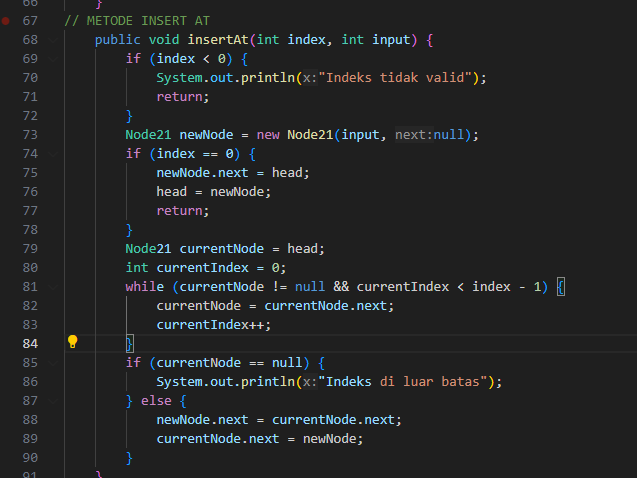
**Pertanyaan**

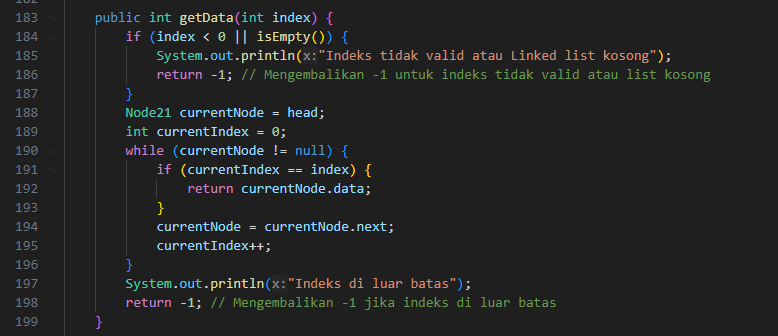
1. Mengapa class LinkedList tidak memerlukan method isFull() seperti halnya Stack dan Queue?
2. Mengapa class LinkedList hanya memiliki atribut head yang menyimpan informasi node pertama? Bagaimana informasi node kedua dan lainnya diakses?
3. Pada langkah, jelaskan kegunaan kode berikut



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

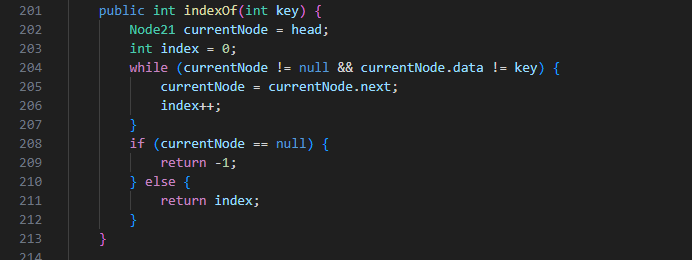
**Jawaban**

1. karena struktur data LinkedList pada dasarnya tidak memiliki batasan ukuran yang tetap seperti Stack atau Queue yang diimplementasikan dengan array
2. struktur dasar dari LinkedList didesain untuk bekerja secara dinamis dan fleksibel dengan memori yang dialokasikan untuk setiap elemen (node) secara individu.
3. untuk menyisipkan sebuah node baru ke dalam LinkedList setelah node yang memiliki nilai data tertentu (key).
4. 

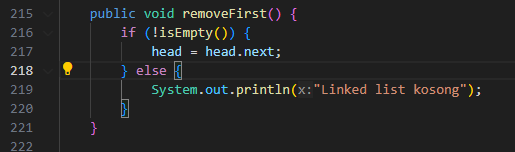
**Praktikum 2 Mengakses dan menghapus node pada Linked List**

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

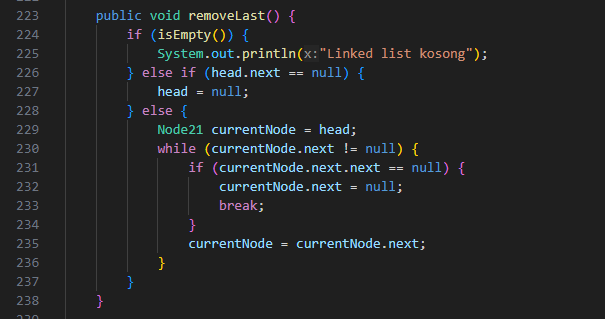
Tambahkan method indexOf() untuk mengetahu index dari node dengan elemen tertentu



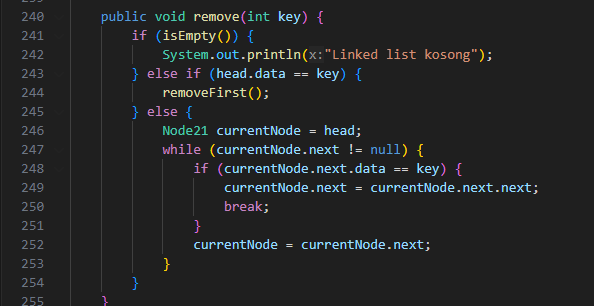
Tambahkan method removeFirst() untuk menghapus node pertama pada linked list



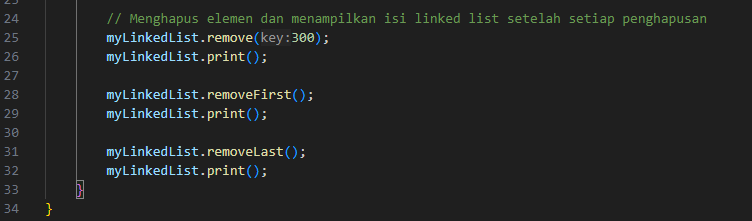
Tambahkan method removeLast() untuk menghapus node terakhir pada linked list



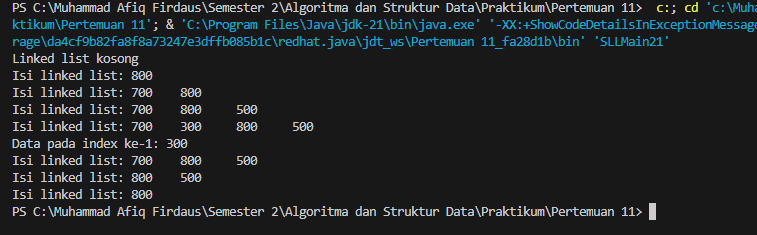
Method remove() digunakan untuk mengapus node yang berisi elemen tertentu



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

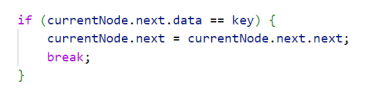


**Verifikasi Hasil Percobaan**

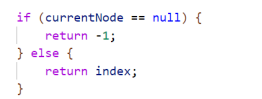


**Pertanyaan**

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



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

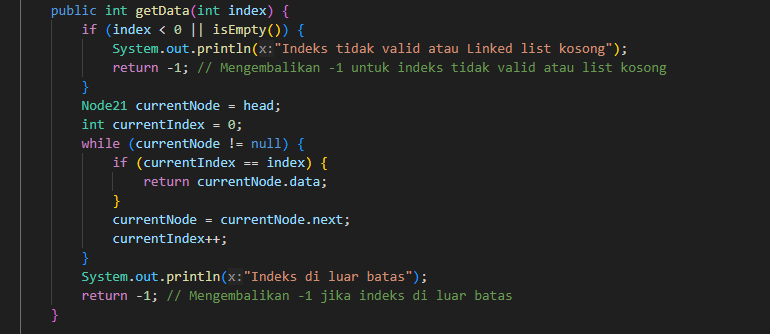


1. Error apa yang muncul jika argumen method getData() lebih besar dari jumlah node pada linked list? Modifikasi kode program untuk menghandle hal tersebut.
2. Apa fungsi keyword break pada method remove()? Bagaimana efeknya jika baris tersebut dihapus?

**Jawaban**

1. untuk menghapus node dengan nilai tertentu dari linked list
2. untuk mencari indeks dari node yang memiliki nilai tertentu (key) dalam linked list
3. maka akan terjadi **NullPointerException**. Hal ini terjadi karena currentNode akan menjadi null saat iterasi mencapai akhir linked list, dan kemudian mencoba mengakses currentNode.data, yang tidak ada (karena currentNode adalah null).

Modifikasi

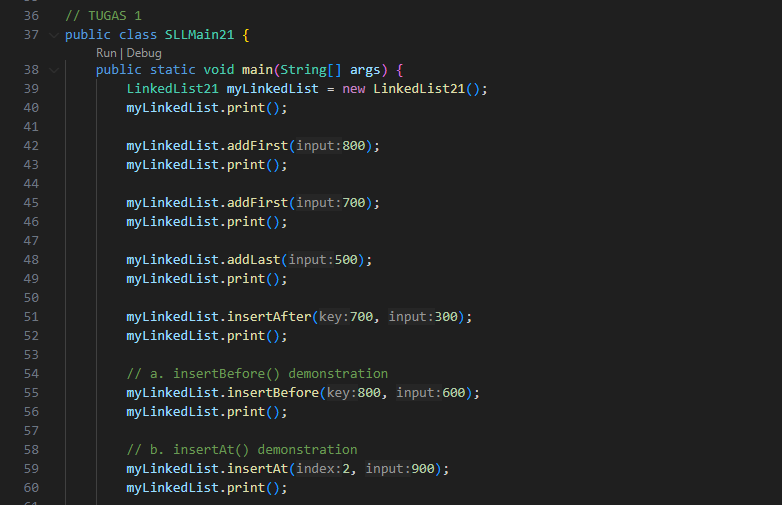


1. Keyword break pada metode remove(int key) memiliki fungsi penting untuk menghentikan eksekusi loop setelah node yang memiliki nilai key ditemukan dan dihapus dari linked list. Jika baris break dihapus, maka loop akan terus berjalan sampai mencapai akhir linked list, meskipun node dengan nilai key sudah ditemukan dan dihapus

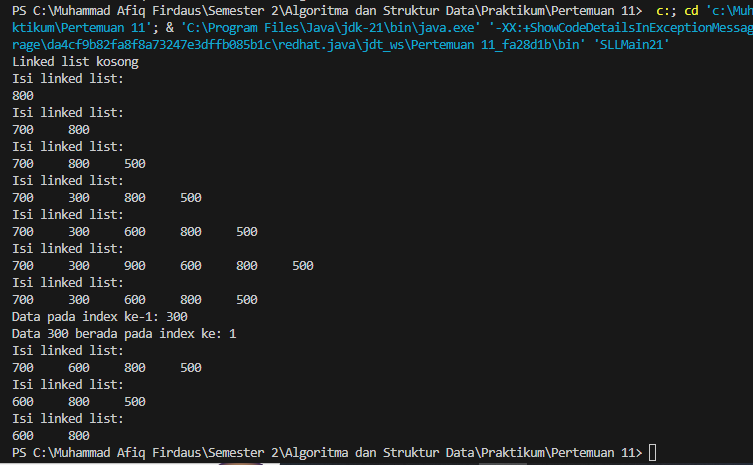
**Tugas**

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
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.

**Jawaban**

1. 





1. Berikut adalah hasil dari code yang telah dibuat

