**Laporan Praktikum Algoritma & Struktur Data**

**Jobsheet 11 : Linkded List**



**Nama: Aqil Rahmat Alifiandi**

**NIM: 2341760107**

**Kelas: SIB 1E**

**PROGRAM STUDI SISTEM INFORMASI BISNIS**

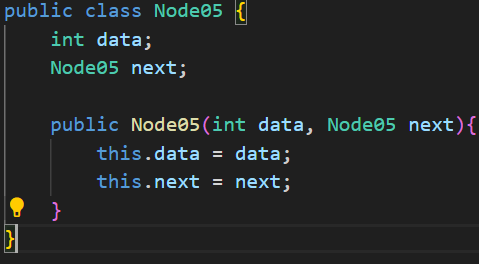
**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

**2024**

**Praktikum 1**

1. Buat folder baru Praktikum09
2. Tambahkan class-class berikut: a. Node.java b. LinkedList.java c. 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



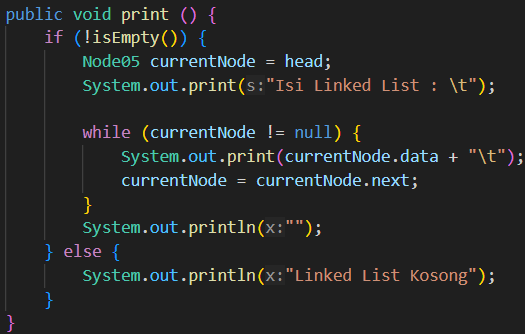
1. Deklarasikan class LinkedList yang memiliki atribut head. Atribut head menyimpan node pertama pada linked list



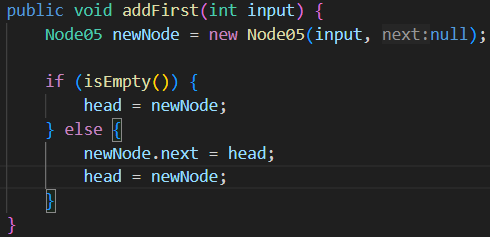
1. Sebagai langkah berikutnya, akan diimplementasikan method-method yang terdapat pada class LinkedList.
2. Tambahkan method isEmpty()



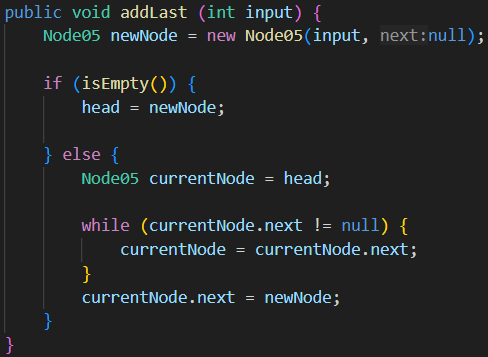
1. Implementasi method print() untuk mencetak dengan menggunakan proses traverse



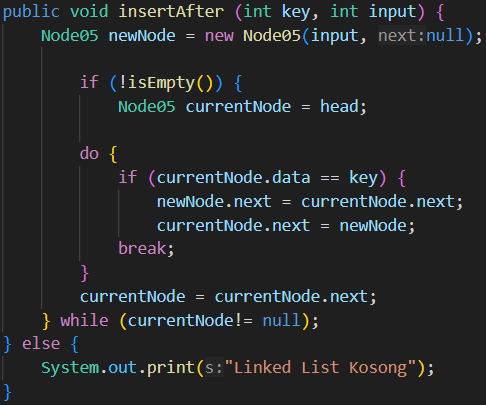
1. Implementasikan method addFirst() untuk menambahkan node baru di awal linked list



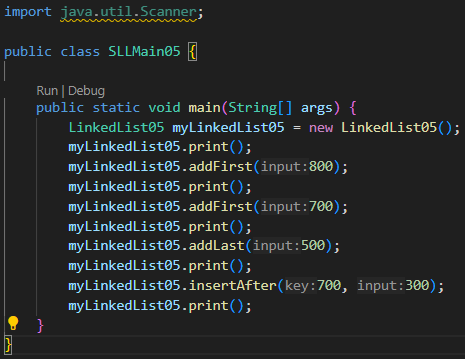
1. Implementasikan method addLast() untuk menambahkan node baru di akhir linked list



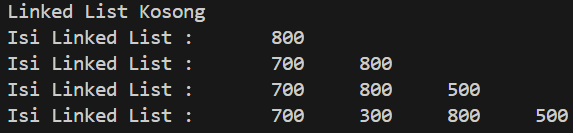
1. Implementasikan method insertAfter() menambahkan node baru pada posisi setelah node yang berisi data tertentu (key)



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()



**HASIL PERCOBAAN:**



**Pertanyaan**

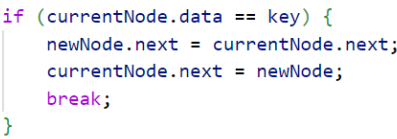
1. Mengapa class LinkedList tidak memerlukan method isFull() seperti halnya Stack dan Queue?

**Jawaban**: LinkedList tidak memerlukan method isFull() karena sifatnya yang dinamis dan tidak memiliki batasan kapasitas maksimum.

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

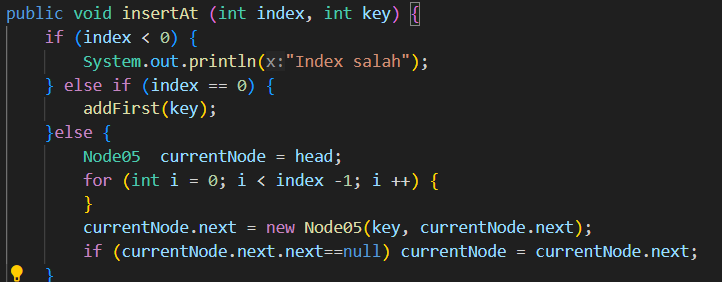
**Jawaban:** Class LinkedList hanya memiliki atribut head karena sifatnya yang dinamis dan struktur data tertautnya. Informasi node selanjutnya dapat diakses melalui pointer next yang ada di setiap node.

1. Pada langkah, jelaskan kegunaan kode berikut



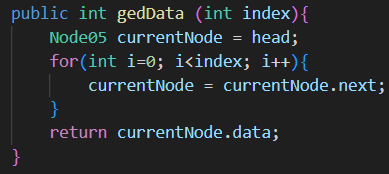
**Jawaban:** bagian dari implementasi class LinkedList yang digunakan untuk menyisipkan node baru dengan nilai yang sama dengan kunci yang dicari ke dalam LinkedList.

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

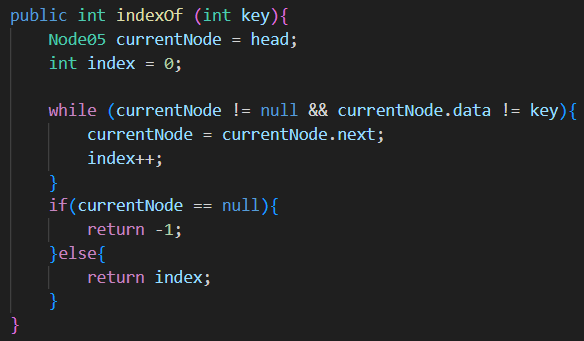
****

**Praktikum 2**

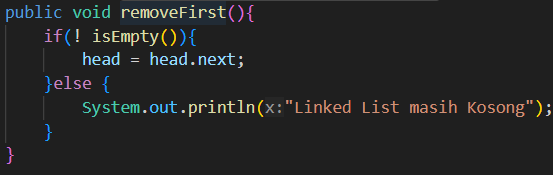
1. Tambahkan method getData() untuk mengembalikan nilai elemen di dalam node pada index tertentu



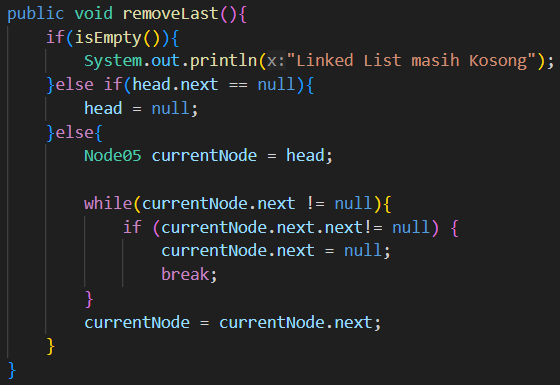
1. Tambahkan method indexOf() untuk mengetahu index dari node dengan elemen tertentu



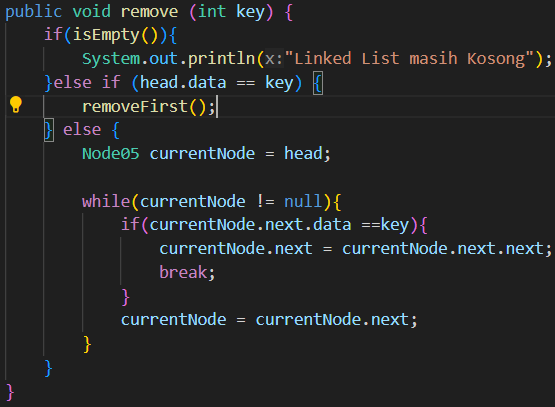
1. Tambahkan method removeFirst() untuk menghapus node pertama pada linked list



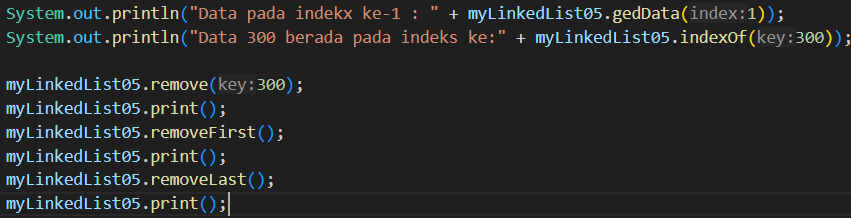
1. Tambahkan method removeLast() untuk menghapus node terakhir pada linked list



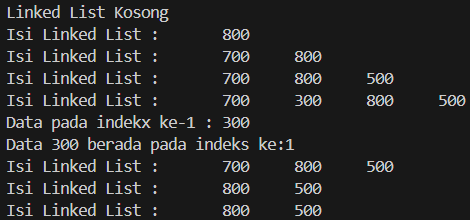
1. Method remove() digunakan untuk mengapus node yang berisi elemen tertentu



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

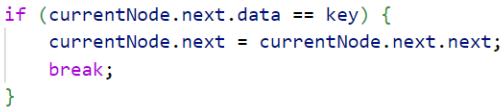


Hasil percobaan:



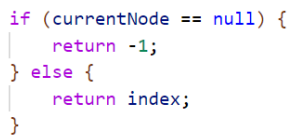
**Pertanyaan:**

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

****

**Jawaban:** Memeriksa apakah data pada node saat ini (currentNode) cocok dengan kunci yang ditentukan (key) dan menghapus node yang sesuai jika cocok.

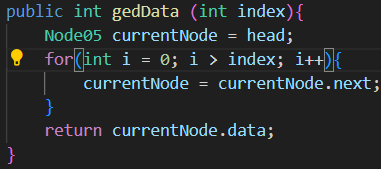
1. Jelaskan maksud if-else block pada method indexOf() berikut

****

**Jawaban:** If-else block pada method indexOf() secara efektif mengimplementasikan algoritma pencarian linier untuk menemukan indeks elemen yang diberikan dalam daftar tertaut.

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

**Jawaban:**

****

Error yang terjadi muncul jika argumen method getData() lebih besar dari jumlah node pada linked list yaitu Exception in thread "main" java.lang.NullPointerException: Cannot read field "next" because "currentNode" is null, berikut modifikasi kode program untuk menghandle hal tersebut

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

**Jawaban:** Pada metode remove(int key), kata kunci break digunakan untuk menghentikan iterasi setelah node yang memiliki nilai key ditemukan dan dihapus. Setelah node tersebut dihapus, tidak perlu lagi melanjutkan iterasi, karena node yang ingin dihapus telah ditemukan dan dihapus dengan sukses

**TUGAS**

1. Implementasikan method-method berikut pada class LinkedList:

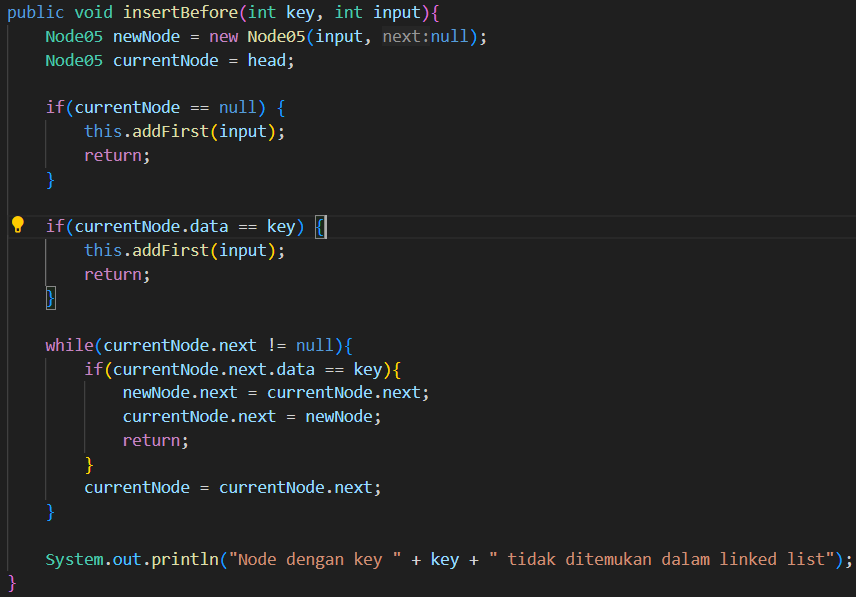
a. insertBefore() untuk menambahkan node sebelum keyword yang diinginkan

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

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

**jawaban:**

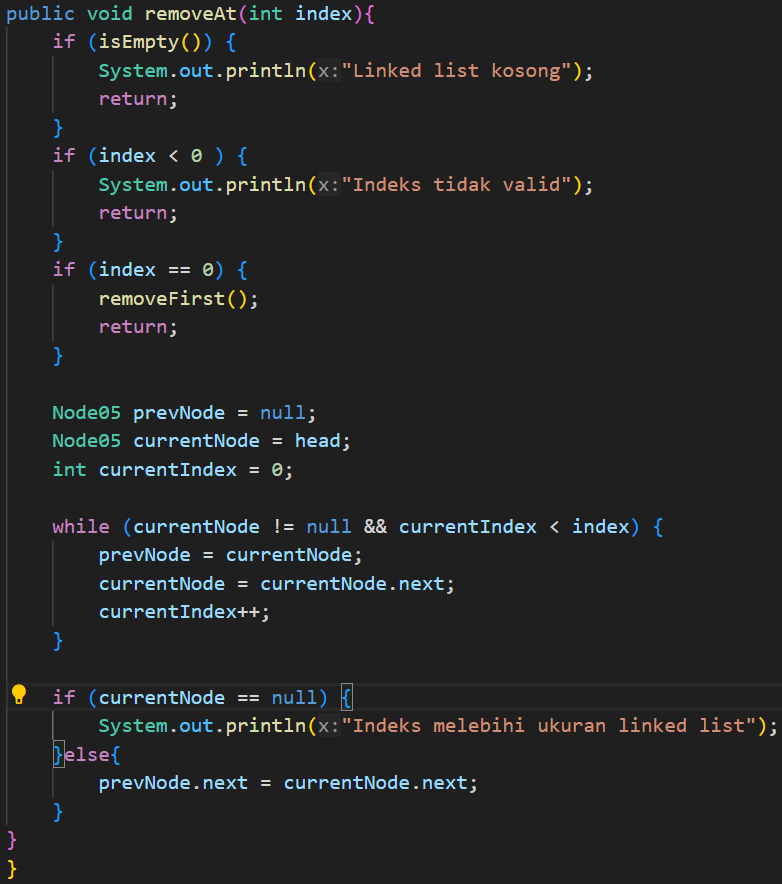
1. InsertBefore()

****

1. insertAt(int index, int key)

****

1. removeAt(int index)

****

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.

Kode Program pada class PointGame05.java dengan memberi beberapa atribut antara

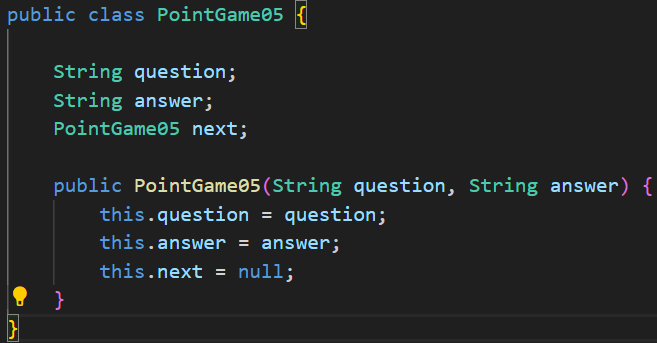
lain:

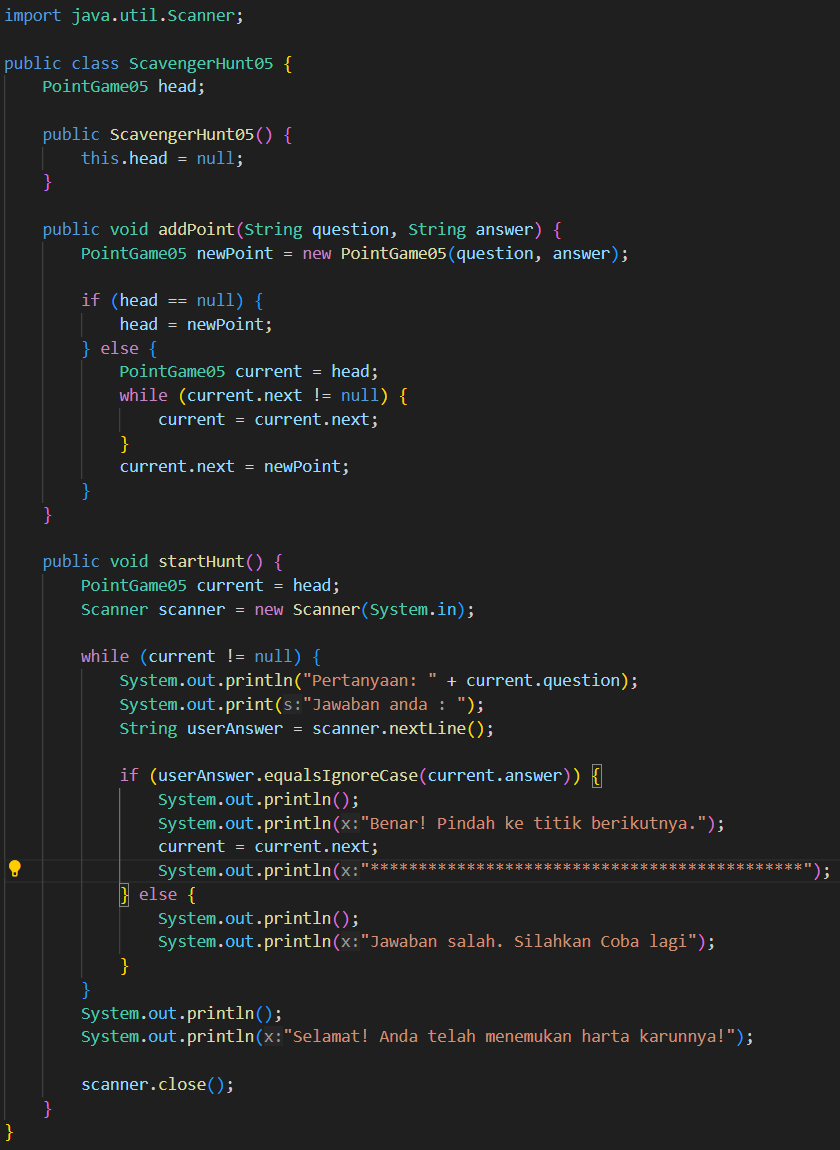
- Question (int)

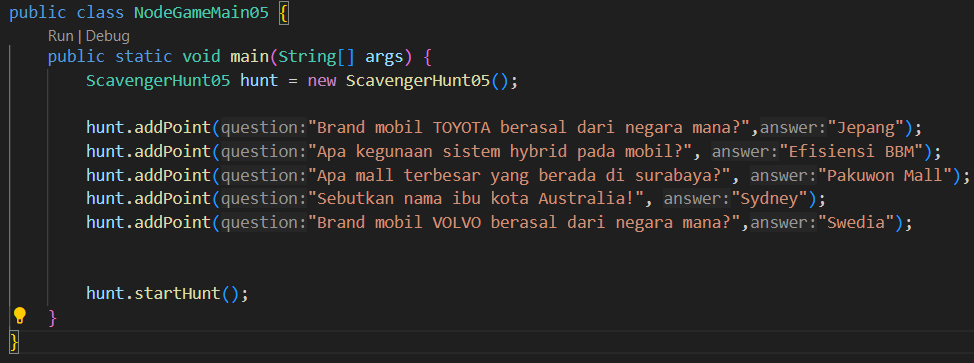
- Answer (String)

- Next (PointGame01)

Serta menuliskan konstruktor berparameternya.







**Hasil Run Program:**

