Nama : Ahmad Franklyn Bima Aquilla

Kelas : SIB 1F – 01

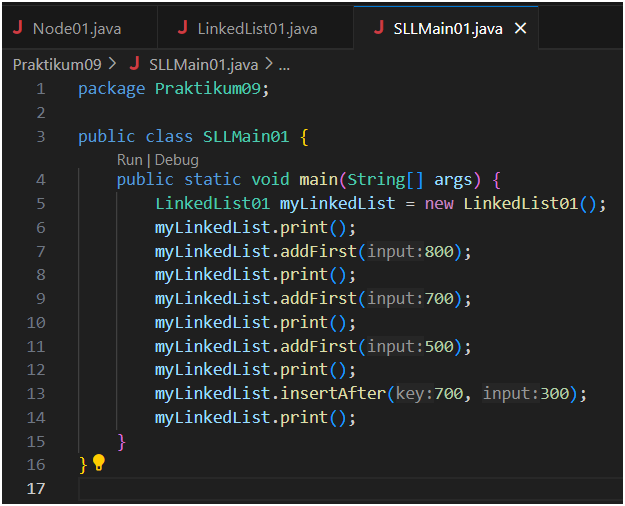
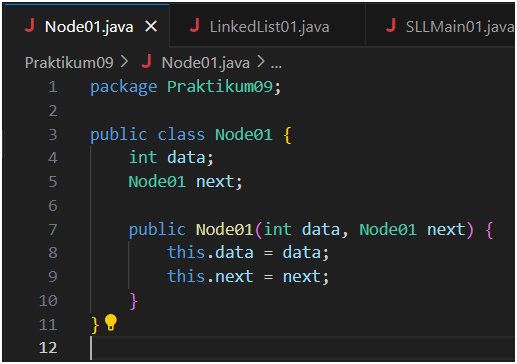
NIM : 2341760027

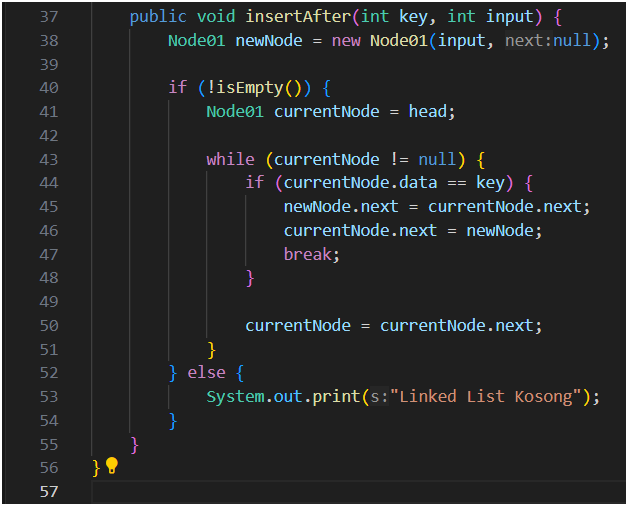
Matkul : Teori Algoritma dan Struktur Data

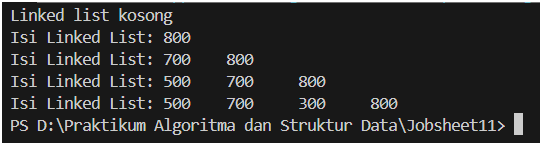
Github : <https://github.com/JustQuill25/PraktikumASD9>

**PERTEMUAN 11**

**(** ASD – *Linked List* **)**

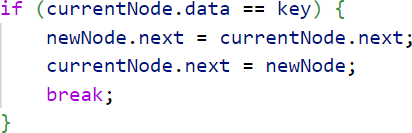
***Class Node01.java Class SLLMain.java***

***Class LinkedList01.java***

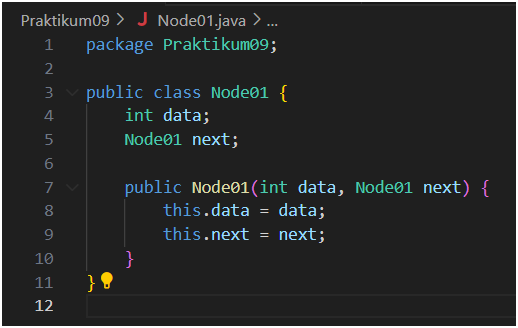
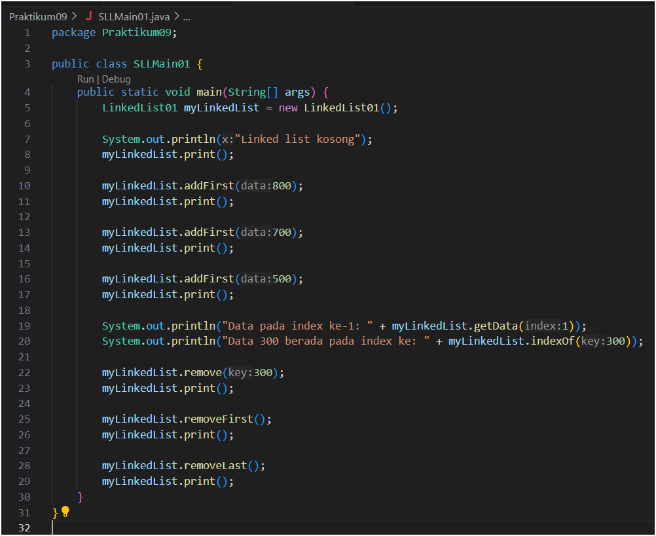
***Output***

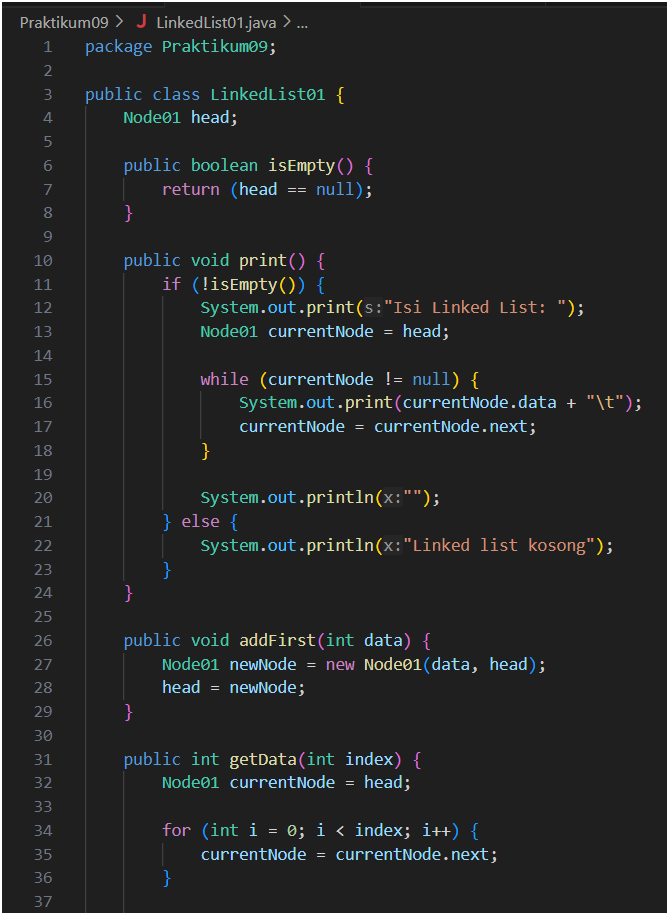
## Pertanyaan

* + - 1. Mengapa class LinkedList tidak memerlukan method isFull() seperti halnya Stack dan Queue?
* pada LinkedList, tidak diperlukan metode **isFull()** karena LinkedList menggunakan alokasi dinamis untuk elemen-elemennya. Selama masih ada memori yang tersedia pada sistem, Akan dapat terus menambahkan elemen ke LinkedList tanpa khawatir akan mencapai batas kapasitas code nya
  + - 1. Mengapa class LinkedList hanya memiliki atribut head yang menyimpan informasi node pertama? Bagaimana informasi node kedua dan lainnya diakses?
* Class **LinkedList** hanya memiliki atribut **head** yang menyimpan informasi tentang node pertama karena sifat dasar dari struktur data linked list. Atribut head sudah cukup untuk mengakses seluruh linked list karena setiap node menyimpan referensi ke node berikutnya
  + - 1. Pada langkah, jelaskan kegunaan kode berikut



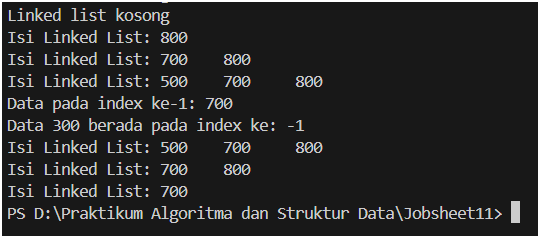
* Kode di atas digunakan dalam metode **insertAfter** pada kelas **LinkedList01**. Memiliki fungsi spesifik untuk menyisipkan node baru setelah node dengan nilai kunci tertentu (**key**). Jadi, keseluruhan kode ini bertujuan untuk menyisipkan node baru setelah node dengan nilai tertentu (**key**) dalam linked list.
  + - 1. Implementasikan method insertAt(int index, int key) dari tugas mata kuliah ASD (Teori)

***Class Node01.java Class Main01.java***

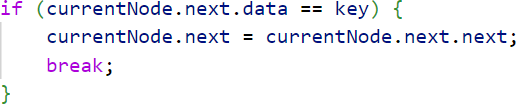
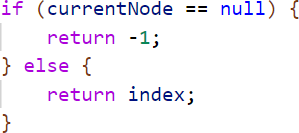
***Class LinkedList01.java***



***Output :***

******

## Pertanyaan

* + - 1. Jelaskan maksud potongan kode di bawah pada method remove()
* Berfungsi dalam method **remove()** untuk menghapus node yang memiliki elemen tertentu (**key**). Dengan demikian, potongan kode tersebut berfungsi untuk menghapus node yang memiliki elemen tertentu (**key**) dari linked list.
  + - 1. Jelaskan maksud if-else block pada method indexOf() berikut
* Blok **if-else** dalam method **indexOf()** digunakan untuk menentukan apakah elemen yang dicari (**key**) ditemukan dalam linked list atau tidak. Blok **if-else** tersebut digunakan untuk mengatur nilai yang akan dikembalikan dari method **indexOf()** berdasarkan apakah elemen yang dicari ditemukan atau tidak dalam linked list.
  + - 1. Error apa yang muncul jika argumen method getData() lebih besar dari jumlah node pada linked list? Modifikasi kode program untuk menghandle hal tersebut.
* Akan terjadi error yang disebut **NullPointerException**. Hal ini terjadi karena method mencoba mengakses node yang tidak ada (melewati node terakhir), yang menyebabkan **currentNode** menjadi **null**.

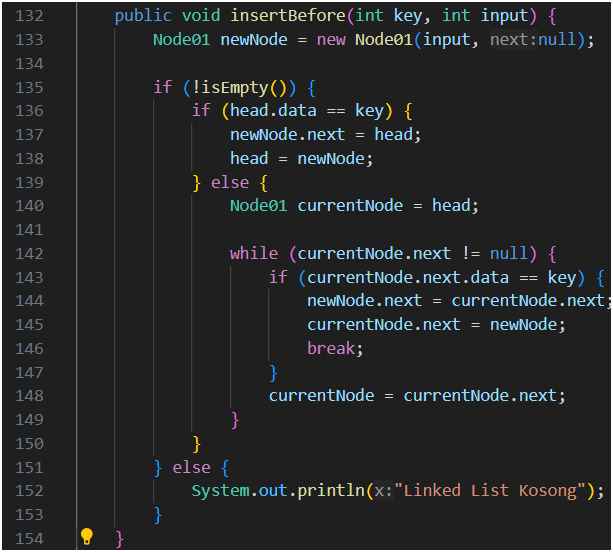
Untuk mengatasi hal ini, kita perlu menambahkan pengecekan apakah indeks yang diminta melebihi jumlah node yang ada dalam linked list.



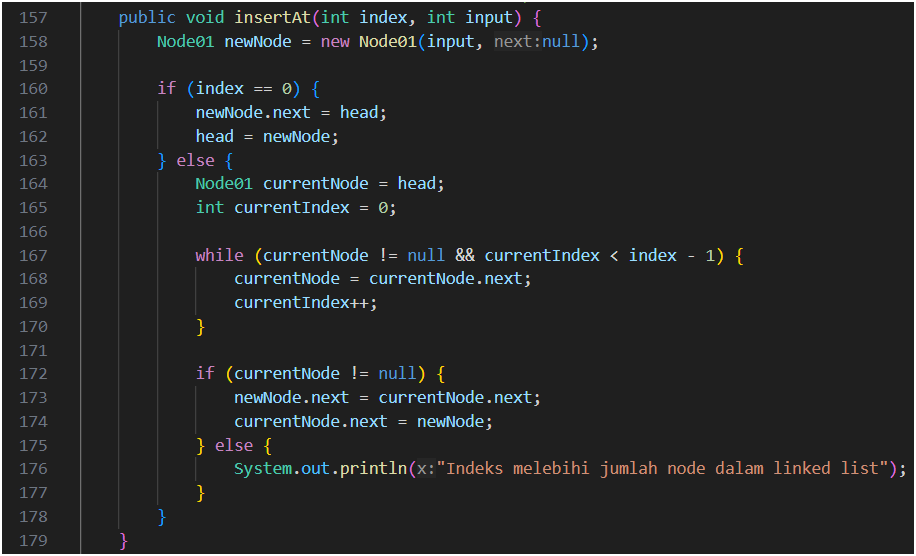
* + - 1. Apa fungsi keyword break pada method remove()? Bagaimana efeknya jika baris tersebut dihapus?
* Keyword **break** digunakan dalam method **remove()** untuk menghentikan perulangan saat node yang sesuai dengan nilai **key** telah ditemukan dan dihapus. Jadi, efek dari menghapus baris **break** adalah perulangan akan berlanjut bahkan setelah operasi penghapusan dilakukan, yang dapat menyebabkan masalah dalam program seperti mengakses node yang tidak ada atau melakukan operasi yang tidak diperlukan secara berulang-ulang

**TUGAS**

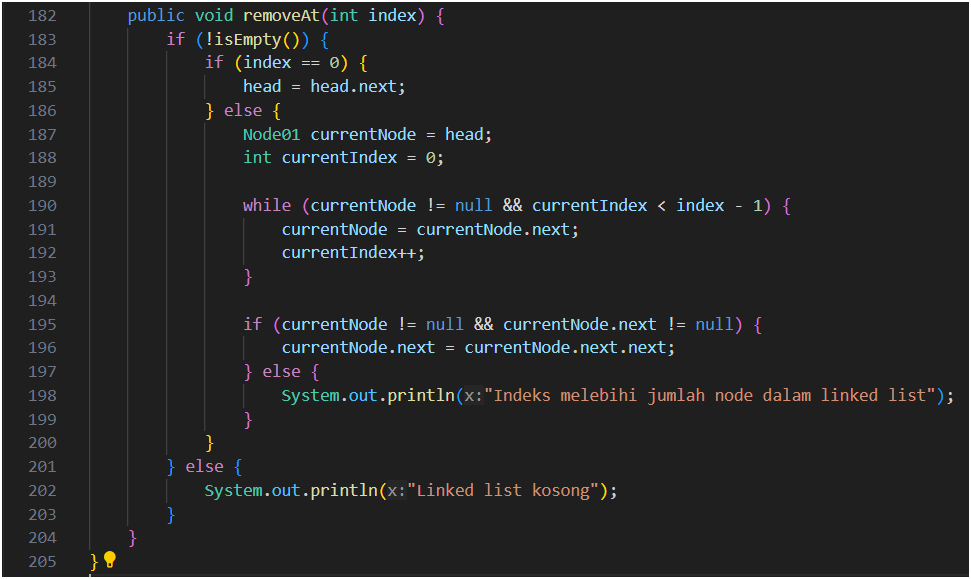
1. Implementasikan method-method berikut pada class LinkedList:
   1. insertBefore() untuk menambahkan node sebelum keyword yang diinginkan



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

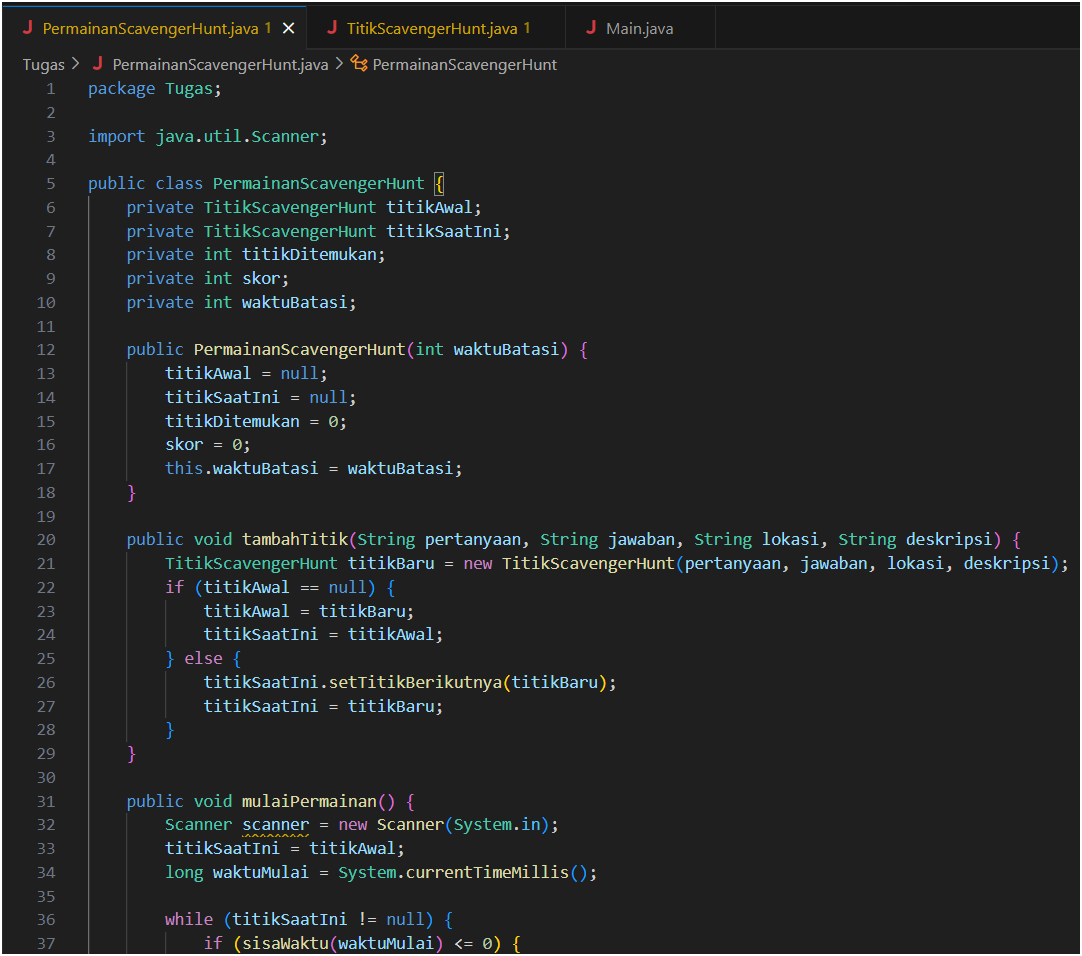


* 1. removeAt(int index) untuk menghapus node pada index tertentu

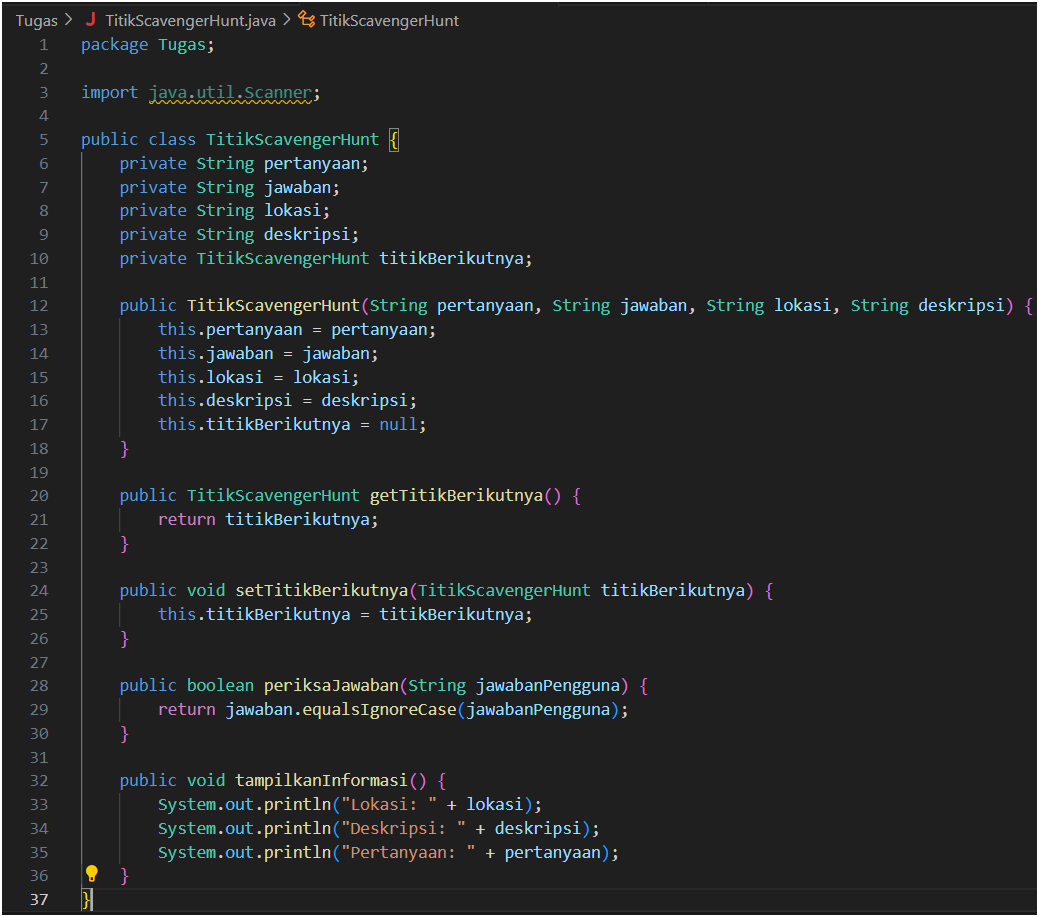


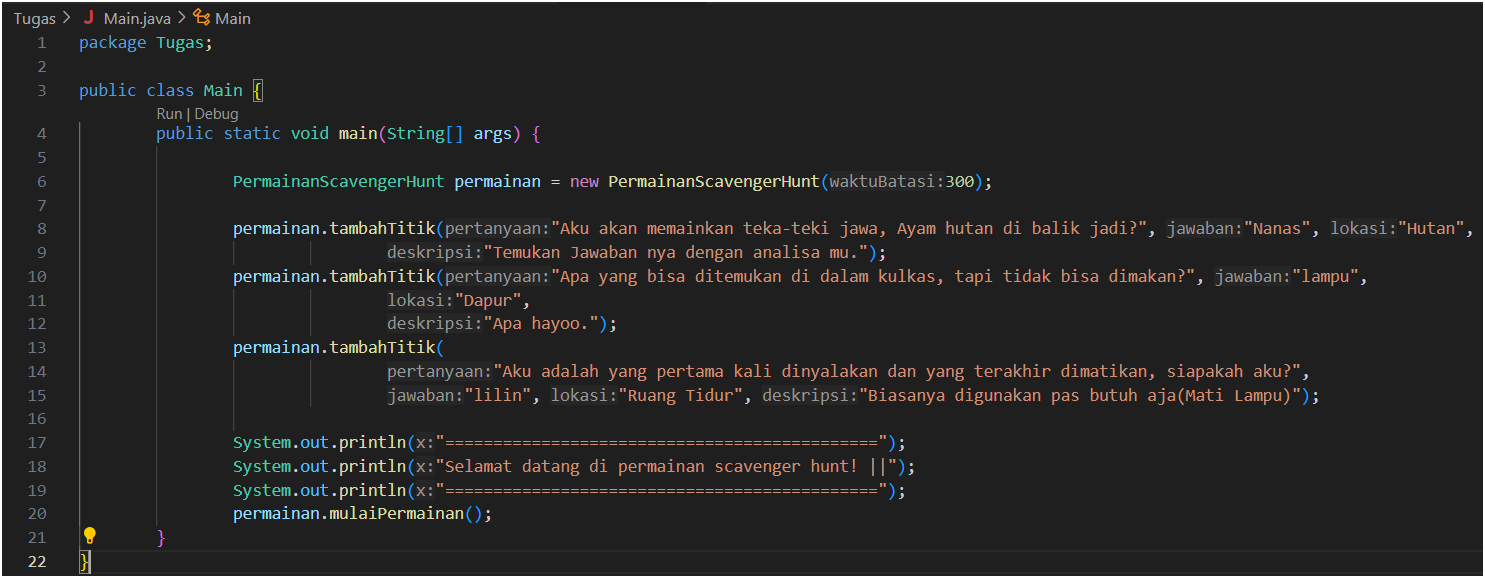
1. Dalam suatu game scavenger hunt, terdapat beberapa point yang harus dilalui peserta untuk menemukan harta karun. Setiap point memiliki soal yang harus dijawab, kunci jawaban, dan pointer ke point selanjutnya. Buatlah implementasi game tersebut dengan linked list.

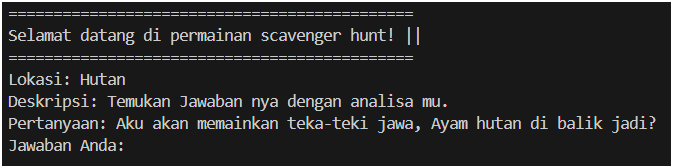
***Class PermainanScavengerHunt.java***



****

***Class TitikScavengerHunt.java***

***Class Main.java***

***Output***