**LAPORAN HASIL PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**



**Oleh:**

**DZULFIKAR MUHAMMAD AL GHIFARI**

**NIM. 2341760071**

**SIB-1F / 08**

**D-IV SISTEM INFORMASI BISNIS**

**JURUSAN TEKNOLOGI INFORMASI**

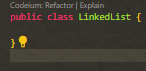
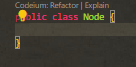
**POLITEKNIK NEGERI MALANG**

**PRAKTIKUM 1**

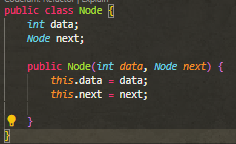
**11.2 Percobaan 1**

**11.2.1 Langkah Langkah percobaan**

1. Membuat class Node, Linkedlist, SLLMain



1. Deklarasi attr & construct class Node



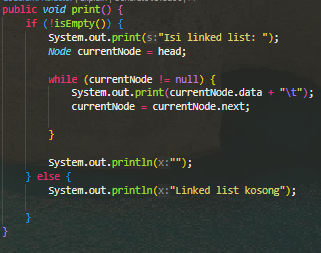
1. Deklarasi class linkedlists



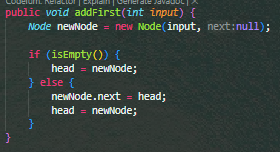
1. Menambahkan method empty pada class linkedlist



1. Menambahkan method print pada class linkedlist



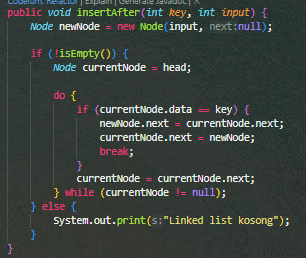
1. Menambahkan method addFirst pada class linkedlist



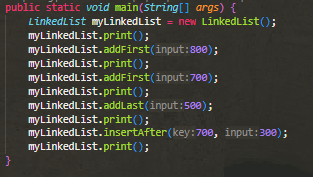
1. Menambahkan method addLast pada class linkedlist



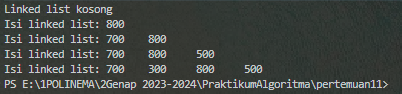
1. Menambahkan method insertAfter pada class linkedlist



9. Membuat method main pada class SSLMain



**VERIFIKASI HASIL PERCOBAAN 11.2.2**



**PERTANYAAN 12.2.3**

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

**Queue?**

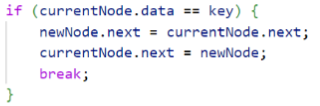
**LinkedList**, memiliki struktur data yang bersifat **linear**, yaitu elemen-elemen terhubung satu sama lain dalam urutan tertentu. LinkedList tidak memiliki batasan kapasitas, sehingga tidak memerlukan methos isFull()

**2. Mengapa class LinkedList hanya memiliki atribut head yang menyimpan informasi node**

**pertama? Bagaimana informasi node kedua dan lainnya diakses?**

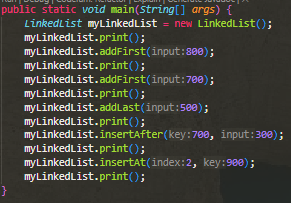
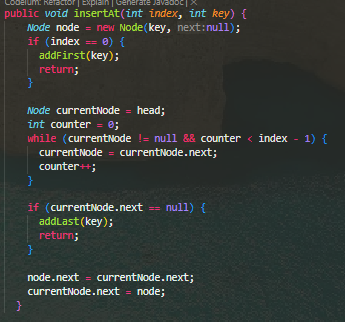
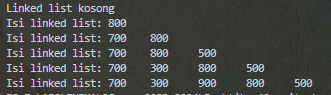
Linkedlist memulai dari head untuk menyimpan node pertama, ketika ingin mengakses node kedua, maka akan mengikuti pointer. Node pertama memiliki pointer yang menunjuk ke node kedua. Untuk mengakses ke node selanjutnya, dapat mengikuti pointer dari node saat ini. Pointer dari setiap node menunjuk ke node berikutnya.

**3. Pada langkah, jelaskan kegunaan kode berikut**

****

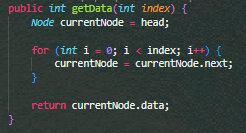
Jika data dari node saat ini cocok dengan attr key, maka node baru akan disisipkan setelah key.

**4. Implementasikan method insertAt(int index, int key) dari tugas mata kuliah ASD (Teori 12.3 Percobaan 2**

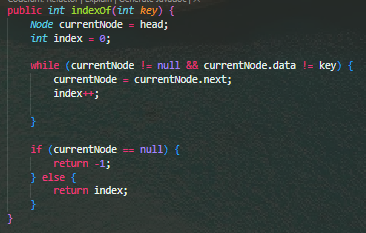
** **

**11.3.1 Langkah-langkah Percobaan**

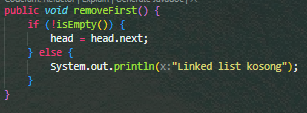
1. Menambahkan method getData pada class LinkedList



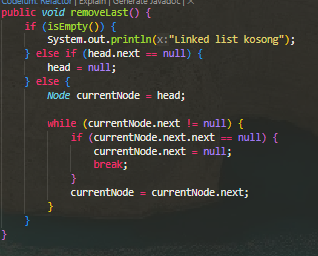
1. Menambahkan method indexOf pada class LinkedList



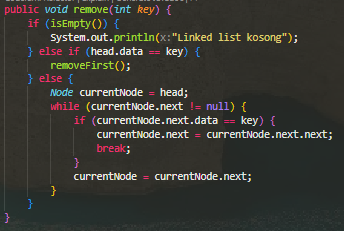
1. Menambahkan method removeFirst pada class LinkedList



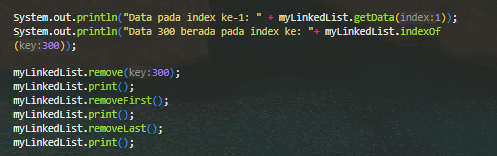
1. Menambahkan method removeLast pada class LinkedList



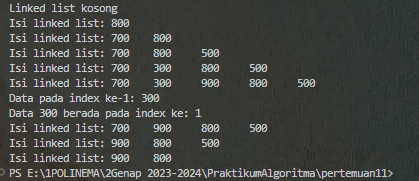
1. Menambahkan method remove pada class LinkedList



1. Menambahkan kodebari pada SLLMain

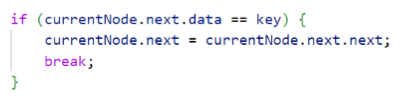


**VERIFIKASI HASIL PERCOBAAN 11.3.2**



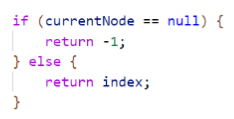
**PERTANYAAN 11.3.3**

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

****

memeriksa apakah data dari node selanjutnya sama dengan key . Jika sama, maka node yang ingin dihapus telah ditemukan.

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

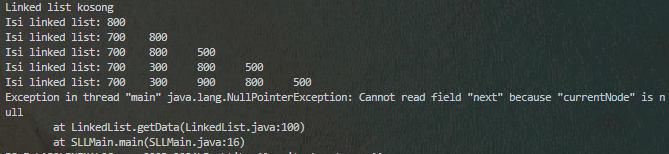
****

Memeriksa apakah code saat ini bernilai null, jika iya maka akan mengembalikan -1, jika tidak maka akan mengembalikan index saat ini untuk kemudian ditampilkan.

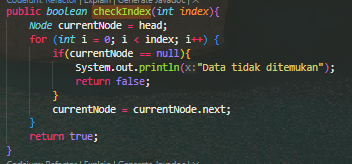
**3. Error apa yang muncul jika argumen method getData() lebih besar dari jumlah node pada**

**linked list? Modifikasi kode program untuk menghandle hal tersebut.**

Akan terjadi error seperti dibawah ini

****

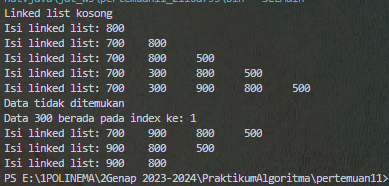
Menambahkan method untuk mengecek apakah data dengan index tersebut ada

****

Mengubah pada main



Hasil compile



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

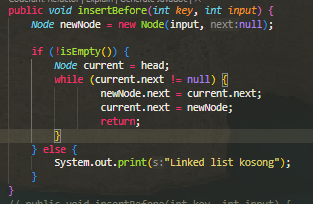
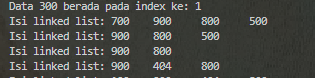
berfungsi untuk menghentikan iterasi loop while lebih awal jika sudah menemukan node yang ingin dihapus

**11.4 TUGAS**

1. Implementasikan method-method berikut pada class LinkedList:

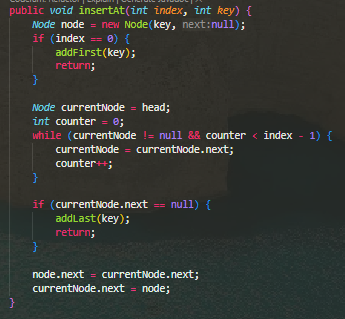
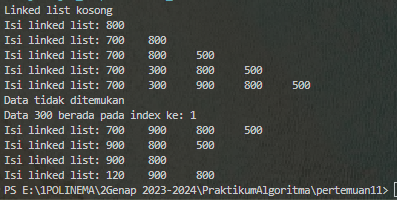
a. insertBefore() untuk menambahkan node sebelum keyword yang diinginkan



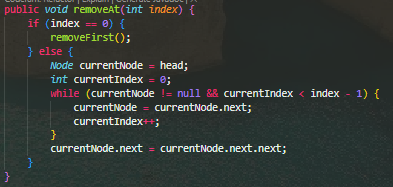
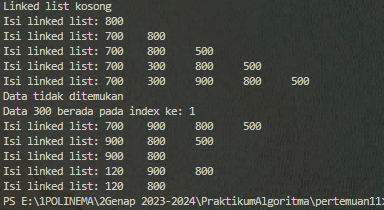
b. insertAt(int index, int key) untuk menambahkan node pada index tertentu



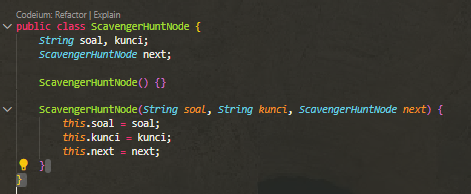
c. removeAt(int index) untuk menghapus node pada index tertentu



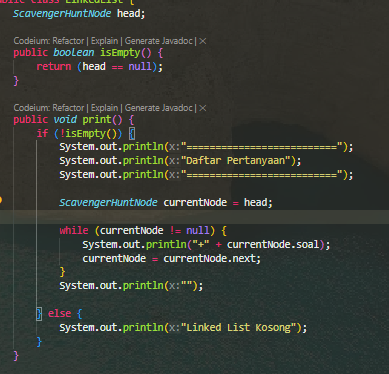
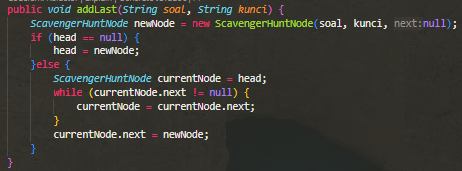
 

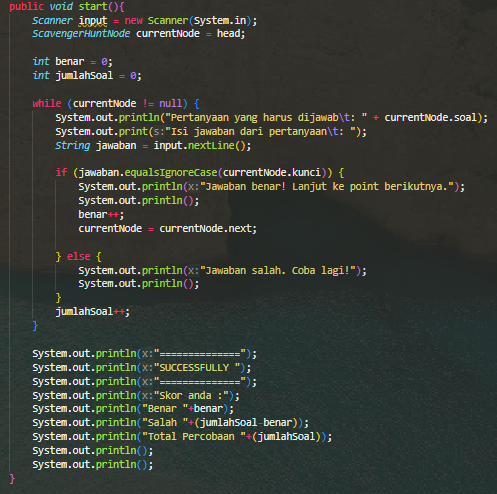
2. Dalam suatu game scavenger hunt, terdapat beberapa point yang harus dilalui peserta untuk menemukan harta karun. Setiap point memiliki soal yang harus dijawab, kunci jawaban, dan pointer ke point selanjutnya. Buatlah implementasi game tersebut dengan linked list.

Class node

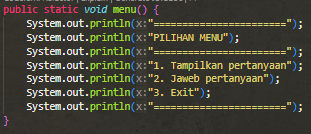


Class linkedlist



Class main





Hasil compile

