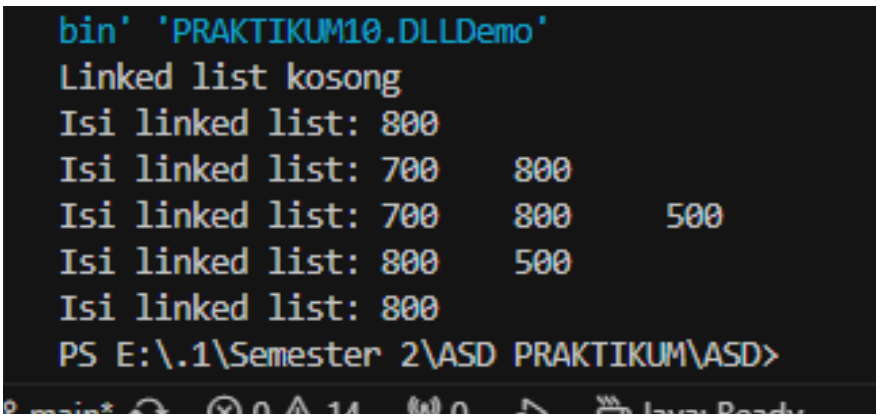
LAPORAN PRAKTIKUM
ALGORITMA DAN STRUKTUR DATA

PRAKTIKUM 10
DOUBLE LINKED LIST

Oleh:
JESSICA AMELIA
2341760185
SIB 1A/16

PROGRAM STUDI SISTEM INFORMASI BISNIS
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2023/2024

Praktikum 1



1. Perhatikan constructor class Node. Jika dimodifikasi sebagai berikut, apakah terdapat perbedaan dengan kode program semula terkait nilai awal atribut next dan prev?

```
public Node(int data)
{
    this.data = data;
}
```

   Jawab : Tidak ada perbedaan dengan kode program semula, karena dalam java atribut objek seperti next dan prev yang tidak diinisialisasi secara eksplisit akan mendapatkan nilai default null.
2. Class DoubleLinkedList tidak memiliki constructor, apakah object dari class tersebut dapat diinstansiasi? Mengapa?
   Jawab : Dapat, karena java otomatis akan menyediakan konstruktor default tanpa argumen untuk kelas yang tidak mendefinisikan konstruktor apapun.
3. Perhatikan implementasi method isEmpty(). Bagaimana jika pengecekan double linked list kosong dilakukan dengan mengevaluasi apakah atribut tail bernilai null? Perlukah kedua atribut (head dan tail) dicek?
   Jawab : Tidak perlu, pengecekan hanya tail atau head biasanya cukup untuk menentukan apakah list kosong, karena dalam implementasi yang benar, kedua atribut tersebut akan selalu konsisten yaitu keduanya null atau keduanya tidak null.
4. Perhatikan method removeFirst(). Dalam kondisi seperti apa nilai head sama dengan tail?

```
public void removeFirst() {
    if (isEmpty()) {
        System.out.println("Linked list kosong");
    } else if (head == tail) {
        head = null;
        tail = null;
    } else {
        head = head.next;
        head.prev = null;
    }
}
```

Jawab : Nilai head sama dengan tail dalam kondisi list hanya memiliki satu node atau satu elemen. Setelah node tersebut dihapus list menjadi kosong baik head maupun tail diatur ke null.

5. Pada praktikum sebelumnya, Anda telah mengimplementasikan method-method berikut pada class LinkedList:
    a. getData() untuk mengembalikan nilai elemen di dalam node pada index tertentu
    b. indexOf() untuk mengetahu index dari node dengan elemen tertentu
    Apakah algoritma yang sama dapat digunakan pada DoubleLinkedList? Jika iya, duplikasi method-method tersebut ke class DoubleLinkedList kemudian lakukan pemanggilan method getData() dan index() pada fungsi main() untuk membuktikannya

Jawab : Algoritma getData() dan indexOf() pada single linked list sama dengan algoritma dalam double linked list.

```java
public class DoubleLinkedList {
    public int getData(int index){
        Node currentNode = head;

        for (int i=0; i< index; i++){
            if(currentNode == null){
                return -1;
            }
            currentNode = currentNode.next;
        }

        return currentNode.data;
    }

    public int indexOf(int key){
        Node currentNode = head;
        int index =0;

        while(currentNode != null && currentNode.data !=key){
            currentNode = currentNode.next;
            index++;
        }

        if(currentNode==null){
            return -1;
        }else {
            return index;
        }
    }
```

```java
public class DLLDemo {
    Run | Debug
    public static void main(String[] args) {
        DoubleLinkedList myDLL = new DoubleLinkedList();
        myDLL.print();
        myDLL.addFirst(input:800);
        myDLL.print();
        myDLL.addFirst(input:700);
        myDLL.print();
        myDLL.addLast(input:500);
        myDLL.print();
        myDLL.removeFirst();
        myDLL.print();
        myDLL.removeLast();
        myDLL.print();
        System.out.println("Data pada indeks 0: "+ myDLL.getData(index:0));
        System.out.println("Data 800 berada pada index ke:  "+ myDLL.indexOf(key:800));

    }
```

```
Linked list kosong
Isi linked list: 800
Isi linked list: 700    800
Isi linked list: 700    800     500
Isi linked list: 800    500
Isi linked list: 800
Data pada indeks 0: 800
Data 800 berada pada index ke:  0
PS E:\.1\Semester 2\ASD PRAKTIKUM\ASD>
```

Tugas

1. Implementasikan method reversePrint() dari tugas ASD Teori

   Jawab :

```java
public class DoubleLinkedList {
    public void reservePrint() {
            System.out.println(x:"Linked List Kosong");
        } else {
            Node currentNode = tail; // Mulai dari node terakhir (tail)
            while (currentNode != null) {
                System.out.print(currentNode.data + " "); // Cetak data dari currentNode
                currentNode = currentNode.prev; // Pindah ke node sebelumnya (prev)
            }
            System.out.println();
        }
    }
```

```java
    Run | Debug
    public static void main(String[] args) {
        DoubleLinkedList myDLL = new DoubleLinkedList();
        myDLL.print();
        myDLL.addFirst(input:800);
        myDLL.print();
        myDLL.addFirst(input:700);
        myDLL.print();
        myDLL.addLast(input:500);
        myDLL.print();
        myDLL.removeFirst();
        myDLL.print();
        myDLL.removeLast();
        myDLL.print();
        System.out.println("Data pada indeks 0: "+ myDLL.getData(index:0));
        System.out.println("Data 800 berada pada index ke:  "+ myDLL.indexOf(key:800));
        myDLL.addFirst(input:400);
        myDLL.print();
        myDLL.addLast(input:900);
        myDLL.print();
        System.out.print(s:"Menampilkan Linked List secara mundur: ");
        myDLL.reservePrint();

    }
```

```
Linked list kosong
Isi linked list: 800
Isi linked list: 700    800
Isi linked list: 700    800     500
Isi linked list: 800    500
Isi linked list: 800
Data pada indeks 0: 800
Data 800 berada pada index ke:  0
Isi linked list: 400    800
Isi linked list: 400    800     900
Menampilkan Linked List secara mundur: 900 800 400
PS E:\.1\Semester 2\ASD PRAKTIKUM\ASD>
```

2. Pada game scavenger hunt pada praktikum sebelumnya, ternyata peserta dalam kembali ke post sebelumnya untuk mengubah jawaban. Modifikasi kode programnya menggunakan struktur data double linked list

   Jawab :

```java
package PRAKTIKUM10.Tugas;
//Modifikasi
class Point {
    String question;
    String answer;
    Point next;
    Point prev;

    Point(String question, String answer) {
        this.question = question;
        this.answer = answer;
        this.next = null;
        this.prev = null;
    }
}
```

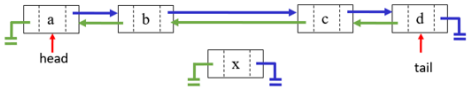
```java
package PRAKTIKUM10.Tugas;
import java.util.Scanner;
//Modifikasi
class ScavengerHunt {
    Point head;
    Point tail;
    Point currentPoint;
    Point lastWrongPoint;
    // Metode untuk menambahkan titik baru dengan pertanyaan dan jawaban
    public void addPoint(String question, String answer) {
        Point newPoint = new Point(question, answer);
        if (head == null) {
            head = newPoint;
            tail = newPoint; //Modifikasi
            currentPoint = head;
        } else {//Modifikasi
            tail.next = newPoint;
            newPoint.prev = tail;
            tail = newPoint;
        }
    }
```

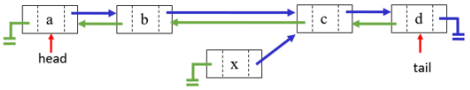
```java
//Modifikasi
// Metode untuk memainkan permainan scavenger hunt
public void play() {
    Scanner scanner = new Scanner(System.in);
    System.out.println(x:"================================");
    System.out.println(x:"Selamat datang di Scavenger Hunt!");
    System.out.println(x:"================================");

    while (currentPoint != null) {
        System.out.println("Pertanyaan: " + currentPoint.question);
        System.out.print(s:"Jawaban Anda: ");
        String userAnswer = scanner.nextLine();

        if (userAnswer.equalsIgnoreCase(currentPoint.answer)) {
            System.out.println(x:"Jawaban Anda benar!");
            currentPoint = currentPoint.next;
            lastWrongPoint = null; // Reset lastWrongPoint setelah jawaban benar
        } else {
            System.out.println(x:"Jawaban Anda salah! Coba lagi.");
            lastWrongPoint = currentPoint; // Set lastWrongPoint ke currentPoint yang salah

            String response;
            do {
                System.out.print(s:"Apakah Anda ingin kembali ke pertanyaan terakhir yang salah? (y/t): ");
                response = scanner.nextLine();
            } while (!response.equalsIgnoreCase(anotherString:"y") && !response.equalsIgnoreCase(anotherString:"t"));

            if (response.equalsIgnoreCase(anotherString:"y") && lastWrongPoint != null) {
                currentPoint = lastWrongPoint;
            } else {
                System.out.println(x:"Melanjutkan ke pertanyaan berikutnya.");
                currentPoint = currentPoint.next;
            }
        }
    }
    scanner.close();
    System.out.println(x:"============================================");
    System.out.println(x:"Selamat! Anda telah menyelesaikan Scavenger Hunt\ndan menemukan harta karun!");
    System.out.println(x:"============================================");
}
```

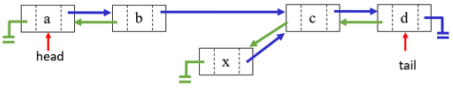
Output :

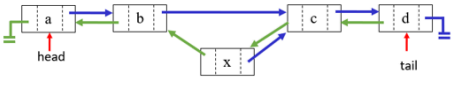
```
================================
Selamat datang di Scavenger Hunt!
================================
Pertanyaan: Siswa newSiswa = new Siswa(); , Tunjukkan yang disebut tipe data?
Jawaban Anda: Siswa
Jawaban Anda benar!
Pertanyaan: Apa tipe data untuk kumpulan karakter?
Jawaban Anda: Integer
Jawaban Anda salah! Coba lagi.
Apakah Anda ingin kembali ke pertanyaan terakhir yang salah? (y/t): y
Pertanyaan: Apa tipe data untuk kumpulan karakter?
Jawaban Anda: String
Jawaban Anda benar!
Pertanyaan: Construktor juga dimaksud dengan method apa ?
Jawaban Anda: Default
Jawaban Anda salah! Coba lagi.
Apakah Anda ingin kembali ke pertanyaan terakhir yang salah? (y/t): t
Melanjutkan ke pertanyaan berikutnya.
Pertanyaan: Class digunakan untuk membentuk apa?
Jawaban Anda: atribut
Jawaban Anda salah! Coba lagi.
Apakah Anda ingin kembali ke pertanyaan terakhir yang salah? (y/t): y
Pertanyaan: Class digunakan untuk membentuk apa?
Jawaban Anda: objek
Jawaban Anda benar!
Pertanyaan: Kata kunci new digunakan untuk ?
Jawaban Anda: instansiasi
Jawaban Anda benar!
============================================
Selamat! Anda telah menyelesaikan Scavenger Hunt
dan menemukan harta karun!
============================================
```

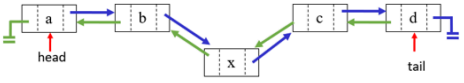
3. Perhatikan langkah-langkah untuk menyisipkan input setelah key tertentu:



1


2


3


4


Implementasikan langkah di atas dengan melengkapi kode program berikut pada kotak merah:

```
public void insertAfter(int key, int input) {
    Node newNode = new Node(input);

    if (!isEmpty()) {
        Node currentNode = head;

        do {
            if (currentNode.data == key) {
                if (currentNode == tail) {
                    addLast(input);
                } else {




                }

                break;
            }

            currentNode = currentNode.next;
        } while (currentNode != null);
    } else {
        System.out.print("Linked list kosong");
    }
}
```

Jawab :

```
//Tugas
public void insertAfter(int key, int input){
    Node newNode = new Node(input);

    if(!isEmpty()){
        Node currentNode = head;

        do{
            if (currentNode.data == key){
                if (currentNode == tail){
                    addLast(input);
                }else {
                    newNode.next = currentNode.next;
                    currentNode.next.prev = newNode;
                    currentNode.next = newNode;
                    newNode.prev = currentNode;
                }

                break;
            }

            currentNode = currentNode.next;
        }while (currentNode != null);
    } else {
        System.out.print(s:"Linked list kosong");
    }
}
```

```java
public class DLLDemo {
    Run | Debug
    public static void main(String[] args) {
        DoubleLinkedList myDLL = new DoubleLinkedList();
        myDLL.print();
        myDLL.addFirst(input:800);
        myDLL.print();
        myDLL.addFirst(input:700);
        myDLL.print();
        myDLL.addLast(input:500);
        myDLL.print();
        myDLL.removeFirst();
        myDLL.print();
        myDLL.removeLast();
        myDLL.print();
        System.out.println("Data pada indeks 0: "+ myDLL.getData(index:0));
        System.out.println("Data 800 berada pada index ke:  "+ myDLL.indexOf(key:800));
        myDLL.addFirst(input:400);
        myDLL.print();
        myDLL.addLast(input:900);
        myDLL.print();
        System.out.print(s:"Menampilkan Linked List secara mundur: ");
        myDLL.reservePrint();
        myDLL.insertAfter(key:400, input:500);
        myDLL.print();
```

Output

```
Linked list kosong
Isi linked list: 800
Isi linked list: 700    800
Isi linked list: 700    800     500
Isi linked list: 800    500
Isi linked list: 800
Data pada indeks 0: 800
Data 800 berada pada index ke:  0
Isi linked list: 400    800
Isi linked list: 400    800     900
Menampilkan Linked List secara mundur: 900 800 400
Isi linked list: 400    500     800     900
PS E:\.1\Semester 2\ASD PRAKTIKUM\ASD>
```