Nama : Monica Tifani Zahara

NRP : 171 111 077 / TI

Praktikum Pemrograman Dasar 2

1. Source code

```
public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        LinkedList a = new LinkedList();
        int data = 0, bdata = 0, pilih;
        char ulang, ulangpr;
        ulang = 'y';
        ulangpr = 'y';

        do {
            System.out.println("=================== MENU ===================");
            System.out.println("1. Masukkan Antrian ");
            System.out.println("2. Tampilkan Antrian ");
            System.out.println("3. Keluarkan Antrian dengan Stack ");
            System.out.println("4. Keluarkan Antrian dengan Queue");
            System.out.println("============================================");
            do {
                System.out.print("Pilih proses\t: ");
                pilih = sc.nextInt();
            } while (pilih < 1 || pilih > 4);
            switch (pilih) {
                case 1:
                    do {
                        System.out.print("Masukkan banyak data\t: ");
                        bdata = sc.nextInt();
                        sc.nextLine();
                        System.out.println("-------------------------------");
                        for (int i = 1; i <= bdata; i++) {
                            System.out.print("Masukkan Data ke " + i + "\t: ");
                            data = sc.nextInt();
                            a.push(new LinkedListNode(data));
                        }
                        do {
                            System.out.print("Tambah antrian lagi? (Y / T)\t");
                            ulangpr = sc.next().charAt(0);
                        } while (ulangpr != 't' && ulangpr != 'y');
                    } while (ulangpr == 'y');
                    do {
                        System.out.print("Kembali ke menu awal ? (Y / T)\t");
                        ulang = sc.next().charAt(0);
                    } while (ulang != 'y' && ulang != 't');

                    break;

```

```
                case 2:
                    System.out.println("============= Data Antrian =============");
                    a.print();
                    do {
                        System.out.print("Kembali ke menu awal ? (Y / T)");
                        ulang = sc.next().charAt(0);
                    } while (ulang != 'y' && ulang != 't');

                    break;

                case 3:
                    if (a.head != null && a.tail != null) {
                        System.out.println("Antrian yang dihapus adalah : " + a.spop().data);
                    } else {
                        System.out.println("Tidak dapat menghapus data. Antrian Kosong !!");
                    }
                    a.print();
                    do {
                        System.out.print("Kembali ke menu awal ? (Y / T)");
                        ulang = sc.next().charAt(0);
                    } while (ulang != 'y' && ulang != 't');

                    break;

                case 4:
                    if (a.head != null && a.tail != null) {
                        System.out.println("Antrian yang dihapus adalah : " + a.qpop().data);
                    } else {
                        System.out.println("Tidak dapat menghapus data. Antrian Kosong !!");
                    }
                    a.print();
                    do {
                        System.out.print("Kembali ke menu awal ? (Y / T)");
                        ulang = sc.next().charAt(0);
                    } while (ulang != 'y' && ulang != 't');
                    break;
            }
        } while (ulang == 'y' || ulang == 'Y');
```

```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package pertemuan3;

import java.util.Scanner;

public class LinkedList {

    LinkedListNode head;
    LinkedListNode tail;

    LinkedList() {
        this.head = null;
        this.tail = null;
    }

    void print() {
        LinkedListNode current = this.head;
        while (current != null) {
            System.out.print(current.data + " ");
            current = current.next;
        }
        System.out.println("");
    }

    void push(LinkedListNode new_node) {
        if (this.head == null && this.tail == null) {
            this.head = new_node;
            this.tail = new_node;
        } else {
            tail.next = new_node;
            new_node.prev = tail;
            this.tail = new_node;

        }
    }
    void insert(LinkedListNode new_node) {
        if (this.head == null) {
            this.head = new_node;
            this.tail = new_node;
        } else if (new_node.data <= this.head.data) {
            this.head.set_prev(new_node);
            this.head = new_node;

        } else if (new_node.data >= this.tail.data) {
            this.tail.set_next(new_node);
            this.tail = new_node;

        } else {
            LinkedListNode position = head;
            while (position.data < new_node.data) {
                position = position.next;

            }
            LinkedListNode previous_position = position.prev;
            new_node.set_prev(previous_position);
            new_node.set_next(position);

        }
    }
```

```java

    LinkedListNode find_node_by_data(int data) {
        LinkedListNode current = this.head;
        while (current != null) {
            if (current.data == data) {
                return current;
            }
            current = current.next;
        }
        return null;
    }

    LinkedListNode lastNode(LinkedListNode node) {
        while (node.next != null) {
            node = node.next;
        }
//        System.out.println("last : " + node.data);
        return node;
    }

    void delete(LinkedListNode deleted) {
        if (deleted != null && this.head != null) {
            if (this.head == this.tail && deleted == this.head) {
                this.head = null;
                this.tail = null;
            } else if (deleted == this.head) {
                LinkedListNode new_head = this.head.next;
                this.head.set_next(null);
                new_head.set_prev(null);
                this.head = new_head;
            } else if (deleted == this.tail) {
                LinkedListNode new_tail = this.tail.prev;
                this.tail.set_prev(null);
                new_tail.set_next(null);
                this.tail = new_tail;
            } else {
                LinkedListNode deleted_prev = deleted.prev;
                LinkedListNode deleted_next = deleted.next;
                deleted.set_prev(null);
                deleted.set_next(null);
                deleted_prev.set_next(deleted_next);
            }
        }

    }
```

```java

    LinkedListNode qpop() {
        LinkedListNode taken = null;
        if (this.head == null && this.tail == null) {
            taken = null;
        } else if (this.head == this.tail) {
            taken = head;
            this.head = null;
            this.tail = null;
        } else {
            taken = head;
            head = head.next;

        }
        return taken;
    }

    LinkedListNode spop() {
        LinkedListNode taken = null;
        if (this.head == null && this.tail == null) {
            taken = null;
        } else if (this.head == this.tail) {
            taken = tail;
            this.head = null;
            this.tail = null;
        } else {
            taken = tail;
            this.tail.prev.set_next(null);
            //tail.prev.next = null;
            this.tail = tail.prev;
        }
        return taken;
    }
```

2. Running program

```
Output - PraktikumProdas2 (run)
run:
=================== MENU ===================
1. Masukkan Antrian
2. Tampilkan Antrian
3. Keluarkan Antrian dengan Stack
4. Keluarkan Antrian dengan Queue
============================================
Pilih proses    : 1
Masukkan banyak data    : 5
-------------------------------
Masukkan Data ke 1      : 7
Masukkan Data ke 2      : 8
Masukkan Data ke 3      : 99
Masukkan Data ke 4      : 4
Masukkan Data ke 5      : 2
Tambah antrian lagi? (Y / T)    y
Masukkan banyak data    : 5
-------------------------------
Masukkan Data ke 1      : 1
Masukkan Data ke 2      : 3
Masukkan Data ke 3      : 23
Masukkan Data ke 4      : 57
Masukkan Data ke 5      : 10
Tambah antrian lagi? (Y / T)    t
Kembali ke menu awal ? (Y / T)  y
=================== MENU ===================
1. Masukkan Antrian
2. Tampilkan Antrian
3. Keluarkan Antrian dengan Stack
4. Keluarkan Antrian dengan Queue
============================================
Pilih proses    : 2
============= Data Antrian =============
7 8 99 4 2 1 3 23 57 10
Kembali ke menu awal ? (Y / T)y
=================== MENU ===================
1. Masukkan Antrian
2. Tampilkan Antrian
3. Keluarkan Antrian dengan Stack
4. Keluarkan Antrian dengan Queue
============================================
Pilih proses    : 3
Antrian yang dihapus adalah : 10
7 8 99 4 2 1 3 23 57
Kembali ke menu awal ? (Y / T)y
=================== MENU ===================
1. Masukkan Antrian
2. Tampilkan Antrian
3. Keluarkan Antrian dengan Stack
4. Keluarkan Antrian dengan Queue
============================================
Pilih proses    : 4
Antrian yang dihapus adalah : 7
8 99 4 2 1 3 23 57
Kembali ke menu awal ? (Y / T)t
BUILD SUCCESSFUL (total time: 1 minute 16 seconds)
```