

## Tugas 4 – Praktikum Pemrograman Dasar

nama : M Irfan S

NRP : 1711111051

-Script

1.

```
1. public class linkedlist {
2.     LinkedListNode head;
3.     LinkedListNode tail;
4.
5.     linkedlist () {
6.         this.head = null;
7.         this.tail = null;
8.     }
9.     void print() {
10.        LinkedListNode current = this.head;
11.        int i = 1;
12.        while (current != null) {
13.            System.out.println(i+ "." + current.data);
14.            current = current.next;
15.            i = i+1;
16.        }
17.        if (this.head == null) {
18.            System.out.println("Kosong");
19.        } else {
20.            System.out.println("");
21.        }
22.    }
23. }
24. public int size() {
25.     int r = 0;
26.     LinkedListNode current = this.head;
27.     while (current != null) {
28.         current = current.next;
29.         r = r+1;
30.     }
31.     return r;
32. }
33. void push(LinkedListNode new_node) {
34.     if (this.head == null && this.tail == null) {
35.         head = new_node;
36.         tail = new_node;
37.     } else {
38.         tail.next = new_node;
39.         new_node.prev = tail;
40.         tail = new_node;
41.     }
42. }
43. LinkedListNode qpop() {
44.     LinkedListNode taken = null;
45.     if (this.head == null && this.tail == null) {
46.         taken = null;
```

```

47. } else if (this.head == this.tail) {
48.     taken = head;
49.     head = null;
50.     tail = null;
51. }else{
52.     taken = head;
53.     head = head.next;
54. }
55.     return taken;
56.}
57. LinkedListNode spop(){
58.     LinkedListNode taken = null;
59.     if (this.head == null && this.tail == null) {
60.         taken = null;
61.     }else if (this.head == this.tail){
62.         taken = tail;
63.         head = null;
64.         tail = null;
65.     }else{
66.         taken = null;
67.         tail.prev.next = null;
68.         tail = tail.prev;
69.     }
70.     return taken;
71.}
72.}

```

2.

```

1. public class LinkedListNode{
2.     LinkedListNode next;
3.     LinkedListNode prev;
4.     String data;
5.
6.     LinkedListNode(String new_data) {
7.         this.data = new_data;
8.         this.prev = null;
9.         this.next = null;
10.    }
11.
12.    void set_prev(LinkedListNode other){
13.        this.prev = other;
14.        if (other != null) {
15.            other.prev = this;
16.        }
17.    }
18.}

```

3.

```

1. import java.util.Scanner;

```

```

2.
3. public class maint {
4.     public static void main(String[] args) {
5.         Scanner sc = new Scanner(System.in);
6.         Scanner pl = new Scanner(System.in);
7.         linkedlist z = new linkedlist();
8.         System.out.println();
9.         int x = 0 ;
10.        do{
11.            System.out.println("=====");
12.            System.out.println("        Antrian PS");
13.            System.out.println("=====");
14.            System.out.println("mau main? pilih dulu boss");
15.            System.out.println("=====");
16.            System.out.println("Menu \n0. lihat daftar antrian \n1. tambah nama dal
am antrian \n2.hapus nama dalam antrian \n3.keluar");
17.            System.out.println("Masukkan Pilihan: ");
18.            String masuk = sc.next();
19.            if (masuk.equals("0")){
20.                System.out.println("nama dalam antrian: ");
21.                z.print();
22.                System.out.println("");
23.                System.out.print("banyak yang mengantri: ");
24.                System.out.print(z.size());
25.                System.out.print("\n\n");
26.                x = x + 1;
27.            }else if(masuk.equals("1")){
28.                System.out.println("Nama baru ");
29.                String nama = pl.next();
30.                z.push(new LinkedListNode(nama));
31.                x = x + 1;
32.            }else if (masuk.equals("2")){
33.                if(z.size() >= 1){
34.                    z.print();
35.                    System.out.println("nama dalam antrian yang dihapus: ");
36.                    System.out.println(z.qpop().data);
37.                } else if (z.size() == 0){
38.                    System.out.println("Data kosong, tidak ada yang dihapus");
39.                }
40.                x = x + 1;
41.            }else if (masuk.equals("3")){
42.                x = 0;
43.            }else{
44.                System.out.println("Error");
45.                System.out.println("inputan anda" + masuk);
46.                x = 0;
47.            }
48.        }while (x != 0);
49.        System.out.println("Kau keluar dari program, makasih !!!11!!1!");
50.    }
51. }

```

## - Screenshot

The screenshot shows an IDE with three tabs: `linkedlist.java`, `linkedlistcode.java - Modul4-Tugas`, and `main.java`. The `main.java` tab is active, displaying a Java program that interacts with a linked list queue. The program starts with a menu and a loop where the user can perform operations like adding, deleting, or viewing the queue. The `linkedlist.java` tab shows the implementation of the `LinkedList` class, which includes methods for adding, deleting, and viewing the queue. The `linkedlistcode.java` tab is also visible, showing the implementation of the `LinkedList` class, which includes methods for adding, deleting, and viewing the queue.

```
File Edit View Terminal Tabs Help
=====ter
mau main? pilih dulu boss
=====
Menu
0. lihat daftar antrian
1. tambah nama dalam antrian
2.hapus nama dalam antrian
3.keluar
Masukkan Pilihan:
1
Nama baru
pler
=====
Antrian PS
=====
mau main? pilih dulu boss
=====
Menu
0. lihat daftar antrian
1. tambah nama dalam antrian
2.hapus nama dalam antrian
3.keluar
Masukkan Pilihan:
0
nama dalam antrian:
1.pler
=====
banyak yang mengantri: 1
=====
Antrian PS
=====
mau main? pilih dulu boss
=====
Menu
0. lihat daftar antrian
1. tambah nama dalam antrian
2.hapus nama dalam antrian
3.keluar
Masukkan Pilihan:
3
Kau keluar dari program, makasih !!!!!!!
+ Tugas clear
=====
Praktikum Informatika 2024/2025/Modul4-Tugas/linkedlist.java 78/100 (72,144%)
+ Tugas jav main.java
```

```
linkedlist.java
linkedlistcode.java - Modul4-Tugas
main.java

class LinkedList {
    private Node head;
    private Node tail;

    public void add(String name) {
        if (head == null) {
            head = new Node(name);
            tail = head;
        } else {
            tail.next = new Node(name);
            tail = tail.next;
        }
    }

    public void delete(String name) {
        if (head == null) {
            return;
        }
        if (head.name.equals(name)) {
            head = head.next;
        } else {
            Node current = head;
            while (current.next != null) {
                if (current.next.name.equals(name)) {
                    current.next = current.next.next;
                }
                current = current.next;
            }
        }
    }

    public void view() {
        if (head == null) {
            System.out.println("Antrian PS");
        } else {
            Node current = head;
            while (current != null) {
                System.out.print(current.name + " ");
                current = current.next;
            }
        }
    }
}

class Node {
    String name;
    Node next;

    Node(String name) {
        this.name = name;
        this.next = null;
    }
}
```