

LAPORAN 7
PRAKTIKUM STRUKTUR DATA LINIER

“Senarai Berantai Tunggal Tidak Berkepala”

Dosen Pengampu : Robertus Adi Nugroho

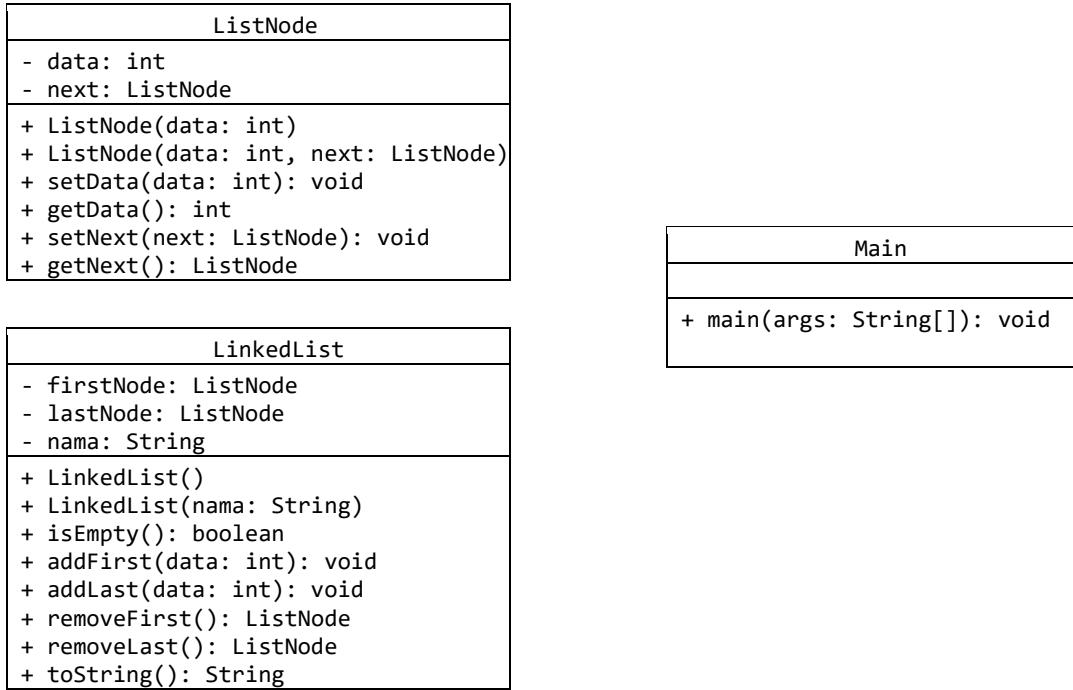


DIBUAT OLEH :
Nama : Yohanis Celvin D.P.U.Pati
NIM : 245314033

KELAS : CP

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS SANATA DHARMA
YOGYAKARTA
2025**

A. DIAGRAM KELAS UML



B. LISTING PROGRAM

```

9   /*
10  */
11 public class LinkedList {
12     private ListNode firstNode, lastNode;
13     private String nama;
14
15     public LinkedList() {
16
17         public LinkedList(string nama) {
18             this.nama = nama;
19         }
20
21         public boolean isEmpty() {
22             if (firstNode == null) {
23                 return true;
24             } else {
25                 return false;
26             }
27
28         public void addFirst(int data) {
29             ListNode newNode = new ListNode(data, firstNode);
30             if (isEmpty()) {
31                 firstNode = newNode;
32                 lastNode = newNode;
33             } else {
34                 firstNode = newNode;
35             }
36
37         public void addLast(int data) {
38             ListNode newNode = new ListNode(data);
39             if (isEmpty()) {
40                 firstNode = newNode;
41                 lastNode = newNode;
42             } else {
43                 lastNode.setNext(newNode);
44                 lastNode = newNode;
45             }
46
47         public ListNode removeFirst() {
48             if (isEmpty()) {
49                 return null;
50             } else {
51                 ListNode removed = firstNode;
52                 firstNode = firstNode.getNext();
53                 if (firstNode == null) {
54                     lastNode = null;
55                 }
56                 return removed;
57             }
58
59         public ListNode removeLast() {
60             if (isEmpty()) {
61                 return null;
62             } else {
63                 ListNode removed = lastNode;
64
65                 if (firstNode == lastNode) {
66                     firstNode = null;
67                     lastNode = null;
68                 } else {
69                     ListNode current = firstNode;
70                     while (current.getNext() != lastNode) {
71                         current = current.getNext();
72                     }
73                     current.setNext(null);
74                     lastNode = current;
75                 }
76                 return removed;
77             }
78
79         public String toString() {
80             StringBuilder sb = new StringBuilder();
81             ListNode current = firstNode;
82
83             while (current != null) {
84                 sb.append(current.getData());
85                 current = current.getNext();
86             }
87             sb.append(")");
88             return sb.toString();
89         }
90
91     }

```

```

8   /*
9  */
10 public class ListNode {
11     private int data;
12     private ListNode next;
13
14     public ListNode(int data) {
15         this.data = data;
16         this.next = null;
17     }
18     public ListNode(int data, ListNode next) {
19         this.data = data;
20         this.next = next;
21     }
22     public void setData(int data) {
23         this.data = data;
24     }
25     public int getData() {
26         return data;
27     }
28     public void setNext(ListNode next) {
29         this.next = next;
30     }
31     public ListNode getNext() {
32         return next;
33     }
34 }
35

```

```

9   /*
10  */
11 public class Main {
12     public static void main(String[] args) {
13         LinkedList list = new LinkedList("SBBTK");
14
15         list.addFirst(8);
16         System.out.println(list);
17
18         list.addFirst(15);
19         System.out.println(list);
20
21         list.addLast(26);
22         System.out.println(list);
23
24         list.addLast(14);
25         System.out.println(list);
26
27         list.removeLast();
28         System.out.println(list);
29
30         list.removeLast();
31         System.out.println(list);
32
33         list.removeFirst();
34         System.out.println(list);
35
36         list.removeFirst();
37         System.out.println(list);
38
39     }

```

C. OUTPUT

```
Building SDLSenaraiBerantai 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----
resources:3.3.1:resources (default-resources) @ SDLSenaraiBerantai --
skip non existing resourceDirectory C:\Users\Windows\OneDrive\Documents\SDLSenaraiBerantai\src\main\resources
compiler:3.13.0:compile (default-compile) @ SDLSenaraiBerantai --
Recompiling the module because of changed source code.
Compiling 3 source files with javac [debug release 23] to target\classes
exec:3.1.0:exec (default-cli) @ SDLSenaraiBerantai --
8,
15,8,
15,8,26,
15,8,26,14,
15,8,26,
15,8,
8,
-----BUILD SUCCESS-----
Total time: 3.267 s
Finished at: 2025-05-08T21:29:25+07:00
-----
```

D. ANALISA

- Ini adalah Analisa hasil atau penjelasan (Base Code) dari program diatas:
(ListNode)

No	Syntax	penjelasan
1	public class ListNode {	//Deklarasi class ListNode sebagai representasi satu elemen (node) dalam linked list.
2	private int data;	//Atribut untuk menyimpan data bertipe integer.
3	private ListNode next;	//Atribut untuk menyimpan referensi ke node berikutnya.
4	public ListNode(int data) { this.data = data; this.next = null; }	//Konstruktor untuk membuat node dengan data tertentu dan next diset ke null.
5	public ListNode(int data, ListNode next) { this.data = data; this.next = next; }	//Konstruktor untuk membuat node dengan data dan referensi ke node selanjutnya.
6	public void setData(int data) { this.data = data; }	//Method setter untuk mengubah nilai data.
7	public int getData() { return data; }	//Method getter untuk mendapatkan nilai data.
8	public void setNext(ListNode next) { this.next = next; }	//Method setter untuk mengatur referensi node berikutnya.

9	<pre>public ListNode getNext() { return next; }</pre>	//Method getter untuk mengambil node berikutnya.
---	---	--

- (LinkedList)

No	Syntax	penjelasan
1	<pre>public class LinkedList {</pre>	//Deklarasi class LinkedList sebagai representasi struktur data linked list
2	<pre>private ListNode firstNode, lastNode;</pre>	//Menyimpan referensi ke node pertama dan terakhir dari list.
3	<pre>private String nama</pre>	//Menyimpan nama dari linked list.
4	<pre>public LinkedList() {}</pre>	//Konstruktor default, tidak memberi nama pada list.
5	<pre>public LinkedList(String nama) { this.nama = nama; }</pre>	//Konstruktor untuk memberi nama pada list saat dibuat.
6	<pre>public boolean isEmpty() { return firstNode == null; }</pre>	//Mengecek apakah list kosong.
7	<pre>public void addFirst(int data) { ListNode newNode = new ListNode(data, firstNode); if (isEmpty()) { firstNode = lastNode = newNode; } else { firstNode = newNode; }}</pre>	//Menambahkan elemen baru di awal list. Jika kosong, first dan last menunjuk ke node baru.
8	<pre>public void addLast(int data) { ListNode newNode = new ListNode(data); if (isEmpty()) { firstNode = lastNode = newNode; } else { lastNode.setNext(newNode); lastNode = newNode; } }</pre>	//Menambahkan elemen baru di akhir list
9	<pre>public ListNode removeFirst() { if (isEmpty()) return null; ListNode removed = firstNode; firstNode = firstNode.getNext(); if (firstNode == null) lastNode = null; return removed; }</pre>	//Menghapus node pertama dan mengembalikannya

10	<pre>public ListNode removeLast() { if (isEmpty()) return null; ListNode removed = lastNode; if (firstNode == lastNode) { firstNode = lastNode = null; } else { ListNode current = firstNode; while (current.getNext() != lastNode) { current = current.getNext(); } current.setNext(null); lastNode = current; } return removed; }</pre>	//Menghapus node terakhir dan mengembalikannya. Jika hanya satu elemen, set keduanya ke null.
11	<pre>public String toString() { StringBuilder sb = new StringBuilder(); ListNode current = firstNode; while (current != null) { sb.append(current.getData()).append(","); current = current.getNext(); } sb.append(" "); return sb.toString(); }</pre>	//Mengembalikan string berisi semua data node dalam list, dipisah koma.