

LAPORAN 7
PRAKTIKUM STRUKTUR DATA LINIER

“Senarai Berantai Tunggal Tidak Berkepala”

Dosen Pengampu : Robertus Adi Nugroho



DIBUAT OLEH :

Nama : Yohanis Calvin D.P.U.Pati

NIM : 245314033

KELAS : CP

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

A. DIAGRAM KELAS UML

ListNode
- data: int - next: ListNode
+ ListNode(data: int) + ListNode(data: int, next: ListNode) + setData(data: int): void + getData(): int + setNext(next: ListNode): void + getNext(): ListNode

LinkedList
- firstNode: ListNode - lastNode: ListNode - nama: String
+ LinkedList() + LinkedList(nama: String) + isEmpty(): boolean + addFirst(data: int): void + addLast(data: int): void + removeFirst(): ListNode + removeLast(): ListNode + toString(): String

Main
+ main(args: String[]): void

B. LISTING PROGRAM

```

9  * @author Calvin Pati
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
29     public void addFirst(int data) {
30         ListNode newNode = new ListNode(data, firstNode);
31         if (isEmpty()) {
32             firstNode = newNode;
33             lastNode = newNode;
34         } else {
35             firstNode = newNode;
36         }
37     }
38
39     public void addLast(int data) {
40         ListNode newNode = new ListNode(data);
41         if (isEmpty()) {
42             firstNode = newNode;
43             lastNode = newNode;
44         } else {
45             lastNode.setNext(newNode);
46             lastNode = newNode;
47         }
48     }
49
50     public ListNode removeFirst() {
51         if (isEmpty()) {
52             return null;
53         } else {
54             ListNode removed = firstNode;
55             firstNode = firstNode.getNext();
56             if (firstNode == null) {
57                 lastNode = null;
58             }
59             return removed;
60         }
61     }
62
63     public ListNode removeLast() {
64         if (isEmpty()) {
65             return null;
66         } else {
67             ListNode removed = lastNode;
68             if (firstNode == lastNode) {
69                 firstNode = null;
70                 lastNode = null;
71             } else {
72                 ListNode current = firstNode;
73                 while (current.getNext() != lastNode) {
74                     current = current.getNext();
75                 }
76                 current.setNext(null);
77                 lastNode = current;
78             }
79             return removed;
80         }
81     }
82
83     public String toString() {
84         StringBuilder sb = new StringBuilder();
85         ListNode current = firstNode;
86         while (current != null) {
87             sb.append(current.getData().append(",");
88             current = current.getNext();
89         }
90         sb.append(" ");
91         return sb.toString();
92     }
93 }

```

```

8  *
9  * @author Calvin Pati
10 */
11 public class ListNode {
12     private int data;
13     private ListNode next;
14
15     public ListNode(int data) {
16         this.data = data;
17         this.next = null;
18     }
19
20     public ListNode(int data, ListNode next) {
21         this.data = data;
22         this.next = next;
23     }
24
25     public void setData(int data) {
26         this.data = data;
27     }
28
29     public int getData() {
30         return data;
31     }
32
33     public void setNext(ListNode next) {
34         this.next = next;
35     }
36
37     public ListNode getNext() {
38         return next;
39     }
40 }

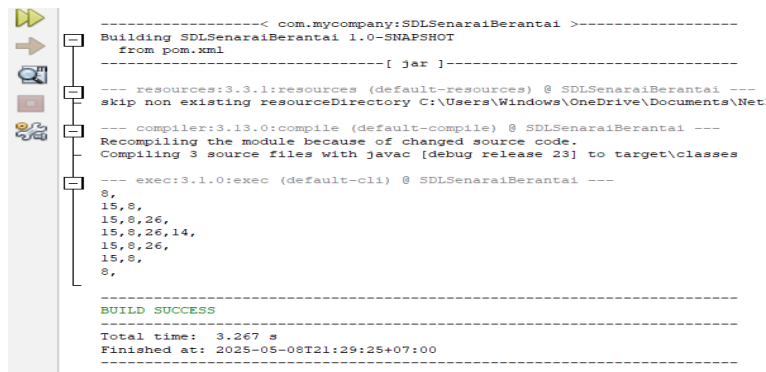
```

```

9  * @author Calvin Pati
10 */
11 public class Main {
12     public static void main(String[] args) {
13         LinkedList list = new LinkedList("SBRTRK");
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



```
-----< com.mycompany:SDLSenaraiBerantai >-----
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\Net
--- 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 ---
0,
15,8,
15,8,26,
15,8,26,14,
15,8,26,
15,8,
0,

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

- (LinkedList)

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