NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Linked List (Jobsheet 9)

1. Praktikum 1 (Pembuatan Single Linked List)

src > P11 > Node.java > ...

```java
package P11;

public class Node {
    int data;
    Node next;

    Node (int nilai, Node berikutnya) {
        data = nilai;
        next = berikutnya;
    }
}
```

src > P11 > SingleLinkedList.java > SingleLinkedList > print()

```java
package P11;

public class SingleLinkedList {
    Node head, tail;

    boolean isEmpty() {
        return head == null;
    }

    void print() {
        if (!isEmpty()) {
            Node tmp = head;
            System.out.println(x:"Isi Linked List:");
            while (tmp != null) {
                System.out.print(tmp.data + "\t");
                tmp = tmp.next;
            }
            System.out.println();
        } else {
            System.out.println(x:"Linked List Kosong");
        }
    }

    void addFirst(int input) {
        Node ndInput = new Node(input, berikutnya:null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    void addLast(int input) {
        Node ndInput = new Node(input, berikutnya:null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }
```

NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Linked List (Jobsheet 9)

```java
public class SingleLinkedList {

    void insertAfter(int key, int input) {
        Node ndInput = new Node(input, berikutnya:null);
        Node temp = head;
        while (temp != null) {
            if (temp.data == key) {
                ndInput.next = temp.next;
                temp.next = ndInput;
                if (ndInput.next == null) {
                    tail = ndInput;
                }
                break;
            }
            temp = temp.next;
        }
    }

    void insertArt(int index, int input) {
        if (index < 0) {
            System.out.println(x:"Perbaiki logikanya! Kalau indeksnya -1 bagaimana?");
        } else if (index == 0) {
            addFirst(input);
        } else {
            Node temp = head;
            for (int i = 0; i < index - 1; i++) {
                if (temp == null) {
                    System.out.println(x:"Indeks melebihi batas");
                    return;
                }
                temp = temp.next;
            }
            if (temp == null) {
                System.out.println(x:"Indeks melebihi batas");
                return;
            }
            temp.next = new Node(input, temp.next);
            if (temp.next.next == null) {
                tail = temp.next;
            }
        }
    }
}
```

src > P11 > SLLMain.java > ...

```java
package P11;

public class SLLMain {
    Run | Debug
    public static void main(String[] args) {
        SingleLinkedList singLL = new SingleLinkedList();

        singLL.print();
        singLL.addFirst(input:890);
        singLL.print();
        singLL.addLast(input:760);
        singLL.print();
        singLL.addFirst(input:700);
        singLL.print();
        singLL.insertAfter(key:700, input:999);
        singLL.print();
        singLL.insertArt(index:3, input:833);
        singLL.print();
    }
}
```

PROBLEMS 22 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01>  & 'C:\Program Files\Java\jdk
-20\bin\java.exe' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\college\semester 2\
Algoritma_dan_Struktur_Data_1G_01\bin' 'P11.SLLMain'
Linked List Kosong
Isi Linked List:
890
Isi Linked List:
890     760
Isi Linked List:
700     890     760
Isi Linked List:
700     999     890     760
Isi Linked List:
700     999     890     833     760
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01>
```

NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Linked List (Jobsheet 9)

Pertanyaan!

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?
   Jawab : Karena, pada saat kita memanggil **singLL.print()** sebelum elemen apa pun ke dalam linked list pada class SLLMain pertama kali.

```
public class SLLMain {
    public static void main(String[] args) {
        SingleLinkedList singLL = new SingleLinkedList();

        singLL.print();
        singLL.addFirst(890);
```

2. Jelaskan kegunaan variable temp secara umum pada setiap method!
   Jawab : Kegunaan variable temp secara umum pada setiap method adalah untuk bergerak dari linked list awal hingga akhir atau hingga kondisi tertentu terpenuhi.
3. Perhatikan class SingleLinkedList, pada method insertAt Jelaskan kegunaan kode berikut!

```
if (temp.next.next == null) {
   tail = temp.next;
}
```

   Jawab : Kegunaan kode tersebut adalah untuk memastikan bahwa referensi tail selalu menunjuk pada node terakhir dalam linked list saat operasi penyisipan sudah dilakukan.

NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Linked List (Jobsheet 9)

2. Praktikum 2 (Modifikasi Elemen pada Single Linked List)

```java
package P11;

public class Node {
    int data;
    Node next;

    Node (int nilai, Node berikutnya) {
        data = nilai;
        next = berikutnya;
    }
}
```

```java
package P11;

public class SingleLinkedList {
    Node head, tail;

    boolean isEmpty() {
        return head == null;
    }

    void print() {
        if (!isEmpty()) {
            Node tmp = head;
            System.out.println(x:"Isi Linked List:");
            while (tmp != null) {
                System.out.print(tmp.data + "\t");
                tmp = tmp.next;
            }
            System.out.println();
        } else {
            System.out.println(x:"Linked List Kosong");
        }
    }

    void addFirst(int input) {
        Node ndInput = new Node(input, berikutnya:null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            ndInput.next = head;
            head = ndInput;
        }
    }

    void addLast(int input) {
        Node ndInput = new Node(input, berikutnya:null);
        if (isEmpty()) {
            head = ndInput;
            tail = ndInput;
        } else {
            tail.next = ndInput;
            tail = ndInput;
        }
    }
```

```java
    }

    void insertAfter(int key, int input) {
        Node ndInput = new Node(input, berikutnya:null);
        Node temp = head;
        while (temp != null) {
            if (temp.data == key) {
                ndInput.next = temp.next;
                temp.next = ndInput;
                if (ndInput.next == null) {
                    tail = ndInput;
                }
                break;
            }
            temp = temp.next;
        }
    }

    void insertArt(int index, int input) {
        if (index < 0) {
            System.out.println(x:"Perbaiki logikanya! Kalau indeksnya -1 bagaimana?");
        } else if (index == 0) {
            addFirst(input);
        } else {
            Node temp = head;
            for (int i = 0; i < index - 1; i++) {
                if (temp == null) {
                    System.out.println(x:"Indeks melebihi batas");
                    return;
                }
                temp = temp.next;
            }
            if (temp == null) {
                System.out.println(x:"Indeks melebihi batas");
                return;
            }
            temp.next = new Node(input, temp.next);
            if (temp.next.next == null) {
                tail = temp.next;
            }
        }
    }
```

NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Linked List (Jobsheet 9)

src > P11 > SingleLinkedList.java > SingleLinkedList > addLast(int)

```
3   public class SingleLinkedList {
86
87      int getData(int index) {
88          Node tmp = head;
89          for (int i = 0; i < index; i++) {
90              tmp = tmp.next;
91          }
92          return tmp.data;
93      }
94
95      int indexOf(int key) {
96          Node temp = head;
97          int index = 0;
98          while (temp != null && temp.data != key) {
99              temp = temp.next;
100             index++;
101         }
102         if (temp != null) {
103             return index;
104         } else {
105             return -1; // Mengembalikan -1 jika kunci tidak ditemukan
106         }
107     }
108
109     void removeFirst() {
110         if (isEmpty()) {
111             System.out.println(x:"Linked list masih kosong, tidak dapat dihapus");
112         } else {
113             head = head.next;
114             if (head == null) {
115                 tail = null;
116             }
117         }
118     }
119
```

src > P11 > SingleLinkedList.java > SingleLinkedList > addLast(int)

```
3   public class SingleLinkedList {
120     void removeLast() {
121         if (isEmpty()) {
122             System.out.println(x:"Linked list masih kosong, tidak dapat dihapus");
123         } else if (head == tail) {
124             head = tail = null;
125         } else {
126             Node temp = head;
127             while (temp.next != tail) {
128                 temp = temp.next;
129             }
130             temp.next = null;
131             tail = temp;
132         }
133     }
134
135     void remove(int key) {
136         if (isEmpty()) {
137             System.out.println(x:"Linked list masih kosong, tidak dapat dihapus");
138         } else {
139             if (head.data == key) {
140                 removeFirst();
141                 return;
142             }
143             Node temp = head;
144             while (temp.next != null && temp.next.data != key) {
145                 temp = temp.next;
146             }
147             if (temp.next != null) {
148                 temp.next = temp.next.next;
149                 if (temp.next == null) {
150                     tail = temp;
151                 }
152             } else {
153                 System.out.println(x:"Elemen tidak ditemukan");
154             }
155         }
156     }
157
```

NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Linked List (Jobsheet 9)

src > P11 > SingleLinkedList.java > SingleLinkedList > addLast(int)

```
3   public class SingleLinkedList {
158     void removeAt(int index) {
159         if (index < 0) {
160             System.out.println(x:"Indeks tidak valid");
161             return;
162         }
163         if (index == 0) {
164             removeFirst();
165             return;
166         }
167         Node temp = head;
168         for (int i = 0; temp != null && i < index - 1; i++) {
169             temp = temp.next;
170         }
171         if (temp == null || temp.next == null) {
172             System.out.println(x:"Indeks melebihi batas");
173             return;
174         }
175         temp.next = temp.next.next;
176         if (temp.next == null) {
177             tail = temp;
178         }
179     }
180 }
181
```

src > P11 > SLLMain.java > SLLMain > main(String[])

```
1   package P11;
2
3   public class SLLMain {
        Run | Debug
4       public static void main(String[] args) {
5           SingleLinkedList singLL = new SingleLinkedList();
6
7           singLL.print();
8           singLL.addFirst(input:890);
9           singLL.print();
10          singLL.addLast(input:760);
11          singLL.print();
12          singLL.addFirst(input:700);
13          singLL.print();
14          singLL.insertAfter(key:700, input:999);
15          singLL.print();
16          singLL.insertArt(index:3, input:833);
17          singLL.print();
18
19          System.out.println("Data pada indeks ke-1 = " + singLL.getData(index:1));
20          System.out.println("Data 760 berada pada indeks ke- " + singLL.indexOf(key:760));
21
22          singLL.remove(key:999);
23          singLL.print();
24          singLL.removeAt(index:0);
25          singLL.print();
26          singLL.removeFirst();
27          singLL.print();
28          singLL.removeLast();
29          singLL.print();
30      }
31  }
32
```

```
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01>  d:; cd 'd:\college\semester
2\Algoritma_dan_Struktur_Data_1G_01'; & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:
+ShowCodeDetailsInExceptionMessages' '-cp' 'D:\college\semester 2\Algoritma_dan_Struktur_
Data_1G_01\bin' 'P11.SLLMain' ta_1G_01\x5cbin' 'P11.SLLMain' ;4297611c-24c8-4e2f-9d58-43b
b6a3370e3
Linked List Kosong
Isi Linked List:
890
Isi Linked List:
890     760
Isi Linked List:
700     890     760
Isi Linked List:
700     999     890     760
Isi Linked List:
700     999     890     833     760
Data pada indeks ke-1 = 999
Data 760 berada pada indeks ke- 4
Isi Linked List:
700     890     833     760
Isi Linked List:
890     833     760
Isi Linked List:
833     760
Isi Linked List:
833
PS D:\college\semester 2\Algoritma_dan_Struktur_Data_1G_01>
```

NAMA : Adani Salsabila
NIM : 2341720123
KELAS : 1-G
MATERI : Linked List (Jobsheet 9)

Pertanyaan!

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!
   Jawab : Digunakannya keyword break pada fungsi remove adalah untuk menghentikan operasi iterasi setelah operasi penghapusan dilakukan.
2. Jelaskan kegunaan kode dibawah pada method remove!
   ```
   } else if (temp.next.data == key) {
   temp.next = temp.next.next;
   ```
   Jawab : Kegunaan kode di atas yang terletak pada method remove adalah untuk menghapus node yang tepat setelah node yang sekarang diperiksa (temp).