

**LAPORAN HASIL PRAKTIKUM ALGORITMA DAN  
STRUKTUR DATA**

**JOBSHEET 9**



**MUHAMMAD AMMAR HAFIZH**

**(2341720074)**

**D-IV TEKNIK INFORMATIKA – 1E**

**Jurusan Teknologi Informasi**

**Politeknik Negeri Malang**

## Hasil Praktikum 1

```
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:\Kuliah\Semester-2\Algoritma Dan Struktur Data\minggu11>
```



```
Node.java  SingleLinkedList.java 1  SLLMain.java X  Jobsheet 9 - Linked List.pdf

Code > SLLMain.java > ...
1  package Code;
2
3  public class SLLMain {
4      Codeium: Refactor | Explain
5      public static void main(String[] args) {
6          SingleLinkedList singLL = new SingleLinkedList();
7          singLL.print();
8          singLL.addFirst(data:890);
9          singLL.print();
10         singLL.addLast(data:760);
11         singLL.print();
12         singLL.addFirst(data:700);
13         singLL.print();
14         singLL.insertAfter(index:700, input:999);
15         singLL.print();
16         singLL.insertAt(index:3, input:833);
17         singLL.print();
18     }
19 }
```

## Pertanyaan Praktikum 1

1. Mengapa hasil compile kode program di baris pertama menghasilkan “Linked List Kosong”?

- Karena pada SLLMain menyuruh untuk print Linked List namun belum ada isi apapun dalam Linked List

2. Jelaskan kegunaan variable temp secara umum pada setiap method!

- Temp pada setiap method difungsikan sebagai penunjuk/pembanding node yang ingin dicari

3. Perhatikan class **SingleLinkedList**, pada method **insertAt** Jelaskan kegunaan kode berikut

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

- Memindahkan tail lama ke inputan baru jika inputan berada di index terakhir

## Hasil Praktikum 2

```
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 3 berada pada index 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:\Kuliah\Semester-2\Algoritma Dan Struktur Data\minggu11>
```

```
Node.java X SingleLinkedList.java M Jobsheet 9 - Linked List.pdf SLLMain.java M X
Code > SLLMain.java > ...
1 package Code;
2
3 public class SLLMain {
4     Codeium: Refactor | Explain
5     Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X
6     public static void main(String[] args) {
7         SingleLinkedList singLL = new SingleLinkedList();
8         singLL.print();
9         singLL.addFirst(data:890);
10        singLL.print();
11        singLL.addLast(data:760);
12        singLL.print();
13        singLL.addFirst(data:700);
14        singLL.print();
15        singLL.insertAfter(index:700, input:999);
16        singLL.print();
17        singLL.insertAt(index:3, input:833);
18        singLL.print();
19        System.out.println("Data pada indeks ke-1 : " + singLL.getData(index:1));
20        System.out.println("Data 3 berada pada index 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 }
```

## Pertanyaan Praktikum 2

1. Mengapa digunakan keyword break pada fungsi remove? Jelaskan!

- Keluar dari inner loop while jika sudah ditemukan kondisi yang sesuai

2. Jelaskan kegunaan kode dibawah pada method remove

```
else if (temp.next.data == key) {  
    temp.next = temp.next.next;
```

- Menghubungkan node sebelum key ke node setelah key artinya node key dilewatkan untuk dihapus