

JOBSHEET IX LINKED LIST

Nama : Siti Nikmatus Sholihah

NIM : 244107020014

Kelas : TI1B

Praktikum 1 : Pembuatan Single Linked List

1. Buat sebuah folder baru dengan nama jobsheet11 pada repository Praktikum-ASD
2. Tambahkan class:
 - a. Mahasiswa22.java
 - b. NodeMahasiswa22.java
 - c. SingleLinkedList22.java
 - d. SLLMain22.java
3. Implementasikan class Mahasiswa22.java sesuai dengan diagram yang ada pada jobsheet.

```
Jobsheet11 > Mahasiswa22.java > Mahasiswa22 > tampilInformasi()
1 public class Mahasiswa22 {
2
3     String nama, nim, kelas;
4     double ipk;
5
6     Mahasiswa22 () {
7
8     }
9
10    Mahasiswa22(String nm, String name, String kls, double ip){
11        nim = nm;
12        nama = name;
13        kelas = kls;
14        ipk = ip;
15    }
16
17    void tampilInformasi(){
18        System.out.println("NIM: " + nim + "      Nama: " + nama + "      Kelas: " + kelas + "      IPK: " + ipk);
19    }
20 }
```

4. Lalu implementasikan class NodeMahasiswa22.java seperti berikut

```
Jobsheet11 > NodeMahasiswa22.java > ...
1 public class NodeMahasiswa22 {
2
3     Mahasiswa22 data;
4     NodeMahasiswa22 next;
5
6     public NodeMahasiswa22(Mahasiswa22 data, NodeMahasiswa22 next){
7         this.data = data;
8         this.next = next;
9     }
10 }
```

5. Tambahkan atribut head dan tail pada class SingleLinkedList22.java

```
Jobsheet11 > SingleLinkedList22.java > SingleLinkedList22 > tail
1 public class SingleLinkedList22 {
2
3     NodeMahasiswa22 head;
4     NodeMahasiswa22 tail;
5 }
```

6. Setelah itu implementasikan method-method pada class SingleLinkedList22.java sebagai berikut.

- a. Tambahkan method isEmpty()

```
Jobsheet11 > SingleLinkedList22.java > ...
1 public class SingleLinkedList22 {
2
3     NodeMahasiswa22 head;
4     NodeMahasiswa22 tail;
5
6     boolean isEmpty(){
7         return (head == null);
8     }
9 }
```

- b. Implementasikan method print() untuk mencetak menggunakan proses traverse

```
Jobsheet11 > SingleLinkedList22.java > SingleLinkedList22 > print()
1 public class SingleLinkedList22 {
2
3     NodeMahasiswa22 head;
4     NodeMahasiswa22 tail;
5
6     boolean isEmpty(){
7         return (head == null);
8     }
9
10    public void print(){
11        if (!isEmpty()) {
12            NodeMahasiswa22 tmp = head;
13            System.out.println("Isi Linked List:\t");
14            while (tmp != null) {
15                tmp.data.tampilInformasi();
16                tmp = tmp.next;
17            }
18            System.out.println("");
19        } else {
20            System.out.println("linked List kosong");
21        }
22    }
23 }
```

- c. Lalu implementasikan method `addFirst()` untuk menambahkan node dari depan

```
Jobsheet11 > SingleLinkedList22.java > SingleLinkedList22 > addFirst(Mahasiswa22)
1 public class SingleLinkedList22 {
2
3     NodeMahasiswa22 head;
4     NodeMahasiswa22 tail;
5
6     boolean isEmpty(){
7         return (head == null);
8     }
9
10    public void print(){
11        if (!isEmpty()) {
12            NodeMahasiswa22 tmp = head;
13            System.out.println("Isi Linked List:\t");
14            while (tmp != null) {
15                tmp.data.tampilInformasi();
16                tmp = tmp.next;
17            }
18            System.out.println("");
19        } else {
20            System.out.println("linked List kosong");
21        }
22    }
23
24    public void addFirst(Mahasiswa22 input){
25        NodeMahasiswa22 ndInput = new NodeMahasiswa22(input, null);
26        if (isEmpty()) {
27            head = ndInput;
28            tail = ndInput;
29        } else {
30            ndInput.next = head;
31            head = ndInput;
32        }
33    }
34 }
```

- d. Lalu implementasikan method `addLast()` untuk menambahkan node di akhir elemen

```
Jobsheet11 > SingleLinkedList22.java > SingleLinkedList22 > addLast(Mahasiswa22)
1 public class SingleLinkedList22 {
35
36    public void addLast(Mahasiswa22 input){
37        NodeMahasiswa22 ndInput = new NodeMahasiswa22(input, null);
38        if (isEmpty()) {
39            head = ndInput;
40            tail = ndInput;
41        } else {
42            tail.next = ndInput;
43            tail = ndInput;
44        }
45    }
46 }
```

- e. Implementasikan method `insertAfter()`, untuk memasukkan node setelah node tertentu

```
Jobsheet11 > SingleLinkedList22.java > SingleLinkedList22 > insertAfter(String, Mahasiswa22)
1 public class SingleLinkedList22 {
46
47    public void insertAfter(String key, Mahasiswa22 input){
48        NodeMahasiswa22 ndInput = new NodeMahasiswa22(input, null);
49        NodeMahasiswa22 temp = head;
50        do {
51            if (temp.data.nama.equalsIgnoreCase(key)) {
52                ndInput.next = temp.next;
53                temp.next = ndInput;
54                if (ndInput.next == null) {
55                    tail = ndInput;
56                }
57                break;
58            } while (temp != null);
59        }
60    }
61 }
```

- f. Tambahkan method penambahan node pada indeks tertentu

```
Jobsheet11 > SingleLinkedList22.java > SingleLinkedList22
1 public class SingleLinkedList22 {
2     public void insertAt(int index, Mahasiswa22 input){
3         if (index < 0) {
4             System.out.println("Index salah");
5         } else if (index == 0) {
6             addFirst(input);
7         } else {
8             NodeMahasiswa22 temp = head;
9             for (int i = 0; i < index - 1; i++) {
10                 temp = temp.next;
11             }
12             temp.next = new NodeMahasiswa22(input, temp.next);
13             if (temp.next.next == null) {
14                 tail = temp.next;
15             }
16         }
17     }
18 }
```

7. Pada class SLLMain22.java buat fungsi main lalu buat object dari class SingleLinkedList22.java

```
Jobsheet11 > SLLMain22.java > SLLMain22 > main(String[])
1 public class SLLMain22 {
2     public static void main(String[] args) {
3         SingleLinkedList22 sll = new SingleLinkedList22();
4     }
5 }
```

8. Buat empat object mahasiswa yang terdiri dari mhs1, mhs2, mhs3, dan mhs4 lalu isikan data mahasiswa tersebut lewat konstruktor

```
Jobsheet11 > SLLMain22.java > ...
1 public class SLLMain22 {
2     public static void main(String[] args) {
3         SingleLinkedList22 sll = new SingleLinkedList22();
4
5         Mahasiswa22 mhs1 = new Mahasiswa22("123", "Amel", "TI1B", 3.6);
6         Mahasiswa22 mhs2 = new Mahasiswa22("124", "Adam", "TI1B", 3.4);
7         Mahasiswa22 mhs3 = new Mahasiswa22("125", "Imel", "TI1B", 3.5);
8         Mahasiswa22 mhs4 = new Mahasiswa22("126", "Zaky", "TI1B", 3.9);
9     }
10 }
11 }
```

9. Tambahkan method penambahan data dan percetakan data di setiap penambahannya agar terlihat perubahannya.

```
Jobsheet11 > SLLMain22.java > SLLMain22
1 public class SLLMain22 {
2     public static void main(String[] args) {
3         SingleLinkedList22 sll = new SingleLinkedList22();
4
5         Mahasiswa22 mhs1 = new Mahasiswa22("123", "Amel", "TI1B", 3.6);
6         Mahasiswa22 mhs2 = new Mahasiswa22("124", "Adam", "TI1B", 3.4);
7         Mahasiswa22 mhs3 = new Mahasiswa22("125", "Imel", "TI1B", 3.5);
8         Mahasiswa22 mhs4 = new Mahasiswa22("126", "Zaky", "TI1B", 3.9);
9
10        sll.print();
11        sll.addFirst(mhs4);
12        sll.print();
13        sll.addLast(mhs1);
14        sll.print();
15        sll.insertAfter("Zaky", mhs3);
16        sll.insertAt(2, mhs2);
17        sll.print();
18    }
19 }
20 }
```

10. Lalu cocokkan hasil compile dengan output yang ada pada jobsheet

```
d:\Praktikum-ASD\Jobsheet11>cd "d:\Praktikum-ASD\Jobsheet11\" && javac SLLMain22.java && java SLLMain22
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8
Picked up JAVA_TOOL_OPTIONS: -Dstdout.encoding=UTF-8 -Dstderr.encoding=UTF-8
linked list kosong
Isi Linked List:
NIM: 126      Nama: Zaky      Kelas: TI1B      IPK: 3.9

Isi Linked List:
NIM: 126      Nama: Zaky      Kelas: TI1B      IPK: 3.9
NIM: 123      Nama: Amel      Kelas: TI1B      IPK: 3.6

Isi Linked List:
NIM: 126      Nama: Zaky      Kelas: TI1B      IPK: 3.9
NIM: 125      Nama: Imel      Kelas: TI1B      IPK: 3.5
NIM: 124      Nama: Adam      Kelas: TI1B      IPK: 3.4
NIM: 123      Nama: Amel      Kelas: TI1B      IPK: 3.6
```

Pertanyaan

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

Jawab :

karena masih belum ada data yang perlu ditampilkan (data masih kosong).

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

Jawab :

Fungsi temp secara umum pada setiap method adalah untuk menyimpan node sementara yang akan dipindahkan agar head / node lain tetap bisa dibaca datanya.

3. Lakukan modifikasi agar data dapat ditambahkan dari keyboard!

Jawab :

Hasil modifikasi:

```
public void insertAfter(String key, Mahasiswa22 input) {
    NodeMahasiswa22 ndInput = new NodeMahasiswa22(input, next:null);
    NodeMahasiswa22 temp = head;
    boolean found = false;

    while (temp != null) {
        if (temp.data.nama.equalsIgnoreCase(key)) {
            ndInput.next = temp.next;
            temp.next = ndInput;
            if (ndInput.next == null) {
                tail = ndInput;
            }
            found = true;
            System.out.println("Data berhasil disisipkan setelah " + key);
            break;
        }
        temp = temp.next;
    }

    if (!found) {
        System.out.println("Key " + key + " tidak ditemukan!");
    }
}
```

```

SLMain22.java > SLMain22 > main(String[])
1  import java.util.Scanner;
2
3  public class SLMain22 {
4      Run | Debug
      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          SingleLinkedList22 sll = new SingleLinkedList22();
7
8          int pilih;
9          do {
10             System.out.println(x:"\nSILAHKAN PILIH MENU");
11             System.out.println(x:"-----");
12             System.out.println(x:"1. Tambahkan data Mahasiswa di awal");
13             System.out.println(x:"2. Sisipkan data Mahasiswa menggunakan key");
14             System.out.println(x:"3. Sisipkan data Mahasiswa menggunakan index");
15             System.out.println(x:"4. Tampilkan data Mahasiswa");
16             System.out.println(x:"0. Keluar");
17             System.out.print(s:"Pilih menu: ");
18             pilih = sc.nextInt();
19             sc.nextLine();
20
21             switch (pilih) {
22                 case 1:
23                     System.out.print(s:"Masukkan NIM: ");
24                     String nim = sc.nextLine();
25                     System.out.print(s:"Masukkan Nama Mahasiswa: ");
26                     String nama = sc.nextLine();
27                     System.out.print(s:"Masukkan kelas: ");
28                     String kelas = sc.nextLine();
29                     System.out.print(s:"Masukkan IPK: ");
30                     double ipk = sc.nextDouble();
31                     sc.nextLine();
32
33                     Mahasiswa22 mhs1 = new Mahasiswa22(nim, nama, kelas, ipk);
34                     sll.addFirst(mhs1);
35                     System.out.println(x:"Data berhasil ditambahkan");
36                     break;
37
38                 case 2:
39                     System.out.print(s:"Masukkan key: ");
40                     String key = sc.nextLine();
41
42                     System.out.print(s:"Masukkan NIM: ");
43                     String newNim = sc.nextLine();
44                     System.out.print(s:"Masukkan Nama: ");
45                     String newName = sc.nextLine();
46                     System.out.print(s:"Masukkan kelas: ");
47                     String newKelas = sc.nextLine();
48                     System.out.print(s:"Masukkan IPK: ");
49                     double newIpk = sc.nextDouble();
50                     sc.nextLine();
51
52                     Mahasiswa22 mhs2 = new Mahasiswa22(newNim, newName, newKelas, newIpk);
53                     sll.insertAfter(key, mhs2);
54                     break;
55
56                 case 3:
57                     System.out.print(s:"Masukkan index seberapa anda ingin menyisipkan data: ");
58                     int index = sc.nextInt();
59
60                     System.out.print(s:"Masukkan NIM: ");
61                     String nimBaru = sc.nextLine();
62                     System.out.print(s:"Masukkan Nama: ");
63                     String namaBaru = sc.nextLine();
64                     System.out.print(s:"Masukkan kelas: ");
65                     String kelasBaru = sc.nextLine();
66                     System.out.print(s:"Masukkan IPK: ");
67                     double ipkBaru = sc.nextDouble();
68                     sc.nextLine();
69
70                     Mahasiswa22 mhs3 = new Mahasiswa22(nimBaru, namaBaru, kelasBaru, ipkBaru);
71                     sll.insertAt(index, mhs3);
72                     break;
73                 case 4:
74                     sll.print();
75                     break;
76
77                 case 0:
78                     System.out.println(x:"Keluar dari program...");
79                     break;
80
81                 default:
82                     System.out.println(x:"Pilihan tidak valid!");
83             }
84         } while (pilih != 0);
85     }
86 }

```

Output:

```
SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 5
linked list kosong

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 1

Masukkan NIM: 123
Masukkan Nama Mahasiswa: Nikma
Masukkan kelas: TE18
Masukkan IPK: 3,7
Data berhasil ditambahkan

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 5
Isi linked List:
NIM: 123      Nama: Nikma      Kelas: TE18      IPK: 3,7

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 2

Masukkan NIM: 124
Masukkan Nama Mahasiswa: Haydar
Masukkan kelas: TE18
Masukkan IPK: 3,8
Data berhasil ditambahkan

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 5
Isi linked List:
NIM: 123      Nama: Nikma      Kelas: TE18      IPK: 3,7
NIM: 124      Nama: Haydar      Kelas: TE18      IPK: 3,8

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 3

Masukkan key: Nikma

Masukkan NIM: 125
Masukkan Nama: Zaky
Masukkan kelas: TE18
Masukkan IPK: 3,7
Data berhasil disisipkan setelah Nikma

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 5
Isi linked List:
NIM: 123      Nama: Nikma      Kelas: TE18      IPK: 3,7
NIM: 125      Nama: Zaky      Kelas: TE18      IPK: 3,7
NIM: 124      Nama: Haydar      Kelas: TE18      IPK: 3,8

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 4

Masukkan index beberapa anda ingin menyisipkan data: 2

Masukkan NIM: 126
Masukkan Nama: Tin
Masukkan kelas: TE18
Masukkan IPK: 3,7

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 5
Isi linked list:
NIM: 123      Nama: Nikma      Kelas: TE18      IPK: 3,7
NIM: 125      Nama: Zaky      Kelas: TE18      IPK: 3,7
NIM: 126      Nama: Tin      Kelas: TE18      IPK: 3,7
NIM: 124      Nama: Haydar      Kelas: TE18      IPK: 3,8

SELARAH PILIH MENU
-----
1. Tambahkan data Mahasiswa di awal
2. Tambahkan data Mahasiswa di akhir
3. Sisipkan data Mahasiswa menggunakan key
4. Sisipkan data Mahasiswa menggunakan index
5. Tampilkan data Mahasiswa
0. Keluar
Pilih menu: 0
Keluar dari program...
```

Praktikum 2: Modifikasi Elemen pada Single Linked List

1. Implementasikan method untuk mengakses data dan index pada linked list
2. Tambahkan method `getData()` yang fungsinya untuk mendapatkan data pada index tertentu

```
1 public class SingleLinkedList22 {  
81     public void getData(int index){  
82         NodeMahasiswa22 tmp = head;  
83         for (int i = 0; i < index; i++) {  
84             tmp = tmp.next;  
85         }  
86         tmp.data.tampilInformasi();  
87     }
```

3. Implementasikan method `indexOf()` yang fungsinya untuk mengecek sebuah data ada pada index ke berapa menggunakan key

```
1 public class SingleLinkedList22 {  
89     public int indexOf(String key){  
90         NodeMahasiswa22 tmp = head;  
91         int index = 0;  
92         while (tmp != null && !tmp.data.nama.equalsIgnoreCase(key)) {  
93             tmp = tmp.next;  
94             index ++;  
95         }  
96  
97         if (tmp == null) {  
98             return -1;  
99         } else {  
100             return index;  
101         }  
102     }
```

4. Tambahkan method `removeFirst()` yang fungsinya untuk menghapus data yang ada di di depan

```
1 public class SingleLinkedList22 {  
104     public void removeFirst(){  
105         if (isEmpty()) {  
106             System.out.println(x:"Linked List masih kosong, tidak dapat dihapus!");  
107         } else if (head == tail) {  
108             head = tail = null;  
109         } else {  
110             head = head.next;  
111         }  
112     }
```

5. Lalu tambahkan method `removeLast()` yang fungsinya untuk menghapus data yang ada di belakang

```
SingleLinkedList22.java > SingleLinkedList22 > insertAfter(String, Mahasiswa22)  
1 public class SingleLinkedList22 {  
114     public void removeLast(){  
115         if (isEmpty()) {  
116             System.out.println(x:"Linked List masih kosong, tidak dapat dihapus!");  
117         } else if (head == tail) {  
118             head = tail = null;  
119         } else {  
120             NodeMahasiswa22 temp = head;  
121             while (temp.next != tail) {  
122                 temp = temp.next;  
123             }  
124             temp.next = null;  
125             tail = temp;  
126         }  
127     }
```


6. Tambahkan method `remove()` yang fungsinya untuk menghapus sebuah node sesuai dengan key

```
SingleLinkedList22.java > SingleLinkedList22 > insertAfter(String, Mahasiswa22)
1 public class SingleLinkedList22 {
129     public void remove(String key){
130         if (isEmpty()) {
131             System.out.println(x:"Linked list masih kosong, tidak dapat dihapus!");
132         } else {
133             NodeMahasiswa22 temp = head;
134             while (temp != null) {
135                 if ((temp.data.nama.equalsIgnoreCase(key)) && (temp == head)) {
136                     this.removeFirst();
137                     break;
138                 } else if (temp.data.nama.equalsIgnoreCase(key)) {
139                     temp.next = temp.next.next;
140                     if (temp.next == null) {
141                         tail = temp;
142                     }
143                     break;
144                 }
145                 temp = temp.next;
146             }
147         }
148     }
}
```

7. Tambahkan method `removeAt()` yang fungsinya untuk menghapus sebuah node menggunakan index

```
SingleLinkedList22.java > SingleLinkedList22 > insertAfter(String, Mahasiswa22)
1 public class SingleLinkedList22 {
150     public void removeAt(int index){
151         if (index == 0) {
152             removeFirst();
153         } else {
154             NodeMahasiswa22 temp = head;
155             for (int i = 0; i < index - 1; i++) {
156                 temp = temp.next;
157             }
158             temp.next = temp.next.next;
159             if (temp.next == null) {
160                 tail = temp;
161             }
162         }
163     }
164 }
```

8. Lalu coba lakukan pengaksesan dan penghapusan data pada method main di class `SLLMain22.java`

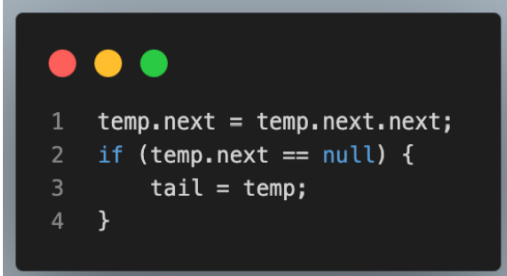
```
SLLMain22.java > SLLMain22 > main(String[])
1 public class SLLMain22 {
2     public static void main(String[] args) {
3         SingleLinkedList22 sll = new SingleLinkedList22();
4
5         Mahasiswa22 mhs1 = new Mahasiswa22(nm:"22212024", name:"Cintia", kls:"3C", ip:3.5);
6         Mahasiswa22 mhs2 = new Mahasiswa22(nm:"23212201", name:"Bimon", kls:"2B", ip:3.8);
7         Mahasiswa22 mhs3 = new Mahasiswa22(nm:"22213062", name:"Putri", kls:"4C", ip:3.5);
8         Mahasiswa22 mhs4 = new Mahasiswa22(nm:"23212209", name:"Satrio", kls:"1D", ip:3.7);
9
10        sll.addFirst(mhs4);
11        sll.addLast(mhs3);
12        sll.insertAfter(key:"Satrio", mhs1);
13        sll.insertAt(index:2, mhs2);
14
15        System.out.println(x:"Data index 1:");
16        sll.getData(index:1);
17
18        System.out.println("Data Mahasiswa AN Bimon ada pada index: " + sll.indexOf(key:"Bimon"));
19        System.out.println();
20
21        sll.removeFirst();
22        sll.removeLast();
23        sll.print();
24        sll.removeAt(index:0);
25        sll.print();
26    }
27 }
```

Pertanyaan

1. Mengapa digunakan keyword break pada fungsi remove?

Jawab : Karena agar menghentikan perulangan while setelah elemen berhasil yang dicari berhasil ditemukan dan dihapus.

2. Jelaskan kegunaan kode dibawah pada method remove()



```
1 temp.next = temp.next.next;
2 if (temp.next == null) {
3     tail = temp;
4 }
```

Jawab : Kode program di atas berfungsi untuk menghapus node setelah node temp dalam linked list. Baris `temp.next = temp.next.next;` akan melewati node setelah temp, sehingga node tersebut tidak lagi terhubung dalam list (terhapus). Kemudian, jika setelah penghapusan node tersebut adalah node terakhir (ditandai dengan `temp.next == null`), maka pointer tail diperbarui agar menunjuk ke node temp sebagai node terakhir yang baru.