

Percobaan 1

```
package P10;

public class Node15 {
    int data;
    Node15 next;

Node15(int niLai, Node15 berikutnya) {
    data = niLai;
    next = berikutnya;
    }

Node15() {

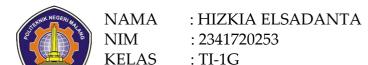
Node15() {

}
```

```
public class StiMain15 {
   public static void main(String[] args) {
        SingleLinkedList15 singll = new SingleLinkedList15();

        singll.print();
        singl
```

```
Linked list Kosong
Isi Linked List
Isi Linked List
                        890
                                760
Tsi Linked List
                        700
                                890
                                        760
Isi Linked List
                        700
                                999
                                        890
                                                760
Isi Linked List
                        700
                                999
                                        890
                                                833
                                                        760
```



MATERI : LINKED LIST

Mengapa hasil compile kode program di baris pertama menghasilkan "Linked List Kosong"?
 Jawab :

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

void print() {
    if (!isEmpty()) {
       Node15 tmp = head;
       System.out.print("Isi Linked List" + "\t\t");
       while(tmp != null) {
            System.out.print(tmp.data + "\t");
            tmp = tmp.next;
            }
            System.out.println(x:"");
            } else {
                System.out.println(x:"Linked list Kosong");
            }
}
```

Pesan "Linked List Kosong" muncul karena di method print() dalam class SingleLinkedList15 adalah if (isEmpty()), dimana isEmpty() mengembalikan true (head == null), maka pesan "Linked list Kosong" akan dicetak.

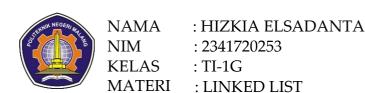
2. Jelaskan kegunaan variable temp secara umum pada setiap method!

Jawab: Pada method insertAfter(int key, int input), variabel temp digunakan untuk menemukan node dengan nilai yang sesuai dengan key, sehingga kita dapat menyisipkan node baru setelahnya. Pada method insertAt(int index, int input), variabel temp digunakan untuk mencapai node sebelum posisi yang ditentukan oleh index, sehingga kita dapat menyisipkan node baru antara temp dan temp.next

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

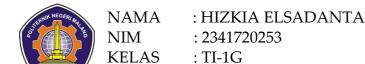
```
if(temp.next.next==null) tail=temp.next;
```

Jawab : Potongan kode tersebut bertujuan untuk memperbarui tail jika node baru disisipkan sebagai node terakhir dalam linked list.



Percobaan 2

```
Node15 tmp = head;
for (int i = 0; i < index; i++) {
    tmp = tmp.next;
                               return tmp.data;
int indexOf(int key) {
  Node15 tmp = head;
  int index = 0;
  while (tmp != null && tmp.data != key) {
      tmp = tmp.next;
      index++;
}
                   void removeFirst() {
  if (isEmpty()) {
    System.out.println("Linked List masih kodong, tidak dapat dihapus");
  } else if (head == tail) {
    head = tail = null;
}
                  void removeLast() {
   if (isEmpty()) {
      System.out.println("Linked list masih kosong, tidak dapat dihapus");
   } else if (head == tail) {
      head = tail = null;
   } else {
      Node15 temp = head;
      while (temp.next != tail) {
         temp = temp.next;
      }
}
                void removeAt(int index) {
   if (index < 0) {
      System.out.println("Indeks nya salah");
   } else if (index == 0) {
      removeFirst();
   } else {
      NodeIS temp = head;
      for (int i = 0; i < index - 1; i++) {
        if (temp.next == null) {
            temp = temp.next;
      }
}</pre>
```



MATERI : LINKED LIST

```
System.out.println("Data pada indeks ke-1=" + singLL.getData(index:1));
System.out.println("Data 3 berada pada indeks ke-" + singLL.indexOf(key:760));

singLL.remove(key:999);
singLL.print();
singLL.removeAt(index:0);
singLL.removeFirst();
singLL.removeFirst();
singLL.removeLast();
singLL.removeLast();
singLL.removeLast();
singLL.print();
}
```

```
Linked list Kosong
Isi Linked List
                         890
Isi Linked List
                                 760
                         890
Isi Linked List
                                 890
                         700
                                         760
                                         890
Isi Linked List
                                 999
                                                  760
                         700
Isi Linked List
                         700
                                 999
                                         890
                                                  833
                                                          760
Data pada indeks ke-1=999
Data 3 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
```

Pertanyaan

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

Jawab : Fungsi keyword break pada method remove adalah untuk menghentikan iterasi ketika node yang dicari telah ditemukan dan node yang ditemukan tersebut akan dihapus.

2. Jelaskan kegunaan kode dibawah pada method remove

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

Jawab: Kegunaan kode tersebut adalah untuk menghapus node dari linked list jika data pada node setelah temp sama dengan key, dengan mengubah referensi node sekarang (temp.next) untuk melompati node yang akan dihapus.

Link github:

https://github.com/Kiaakk/Algoritma_Struktur_Data_1G_15.git