



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : LINKED LIST

Percobaan 1

```
1 package P10;  
2  
3 public class Node15 {  
4     int data;  
5     Node15 next;  
6  
7     Node15(int nilai, Node15 berikutnya) {  
8         data = nilai;  
9         next = berikutnya;  
10    }  
11  
12    Node15() {  
13  
14    }  
15 }  
16
```

```
1 package P10;  
2  
3 public class SingleLinkedList15 {  
4     Node15 head, tail;  
5  
6     boolean isEmpty() {  
7         return head == null;  
8     }  
9  
10    void print() {  
11        if (!isEmpty()) {  
12            Node15 tmp = head;  
13            System.out.print("Isi Linked List" + "\t\t");  
14            while (tmp != null) {  
15                System.out.print(tmp.data + "\t");  
16                tmp = tmp.next;  
17            }  
18            System.out.println("");  
19        } else {  
20            System.out.println("Linked list Kosong");  
21        }  
22    }  
23  
24    void addFirst(int input) {  
25        Node15 ndInput = new Node15(input, null);  
26        if (isEmpty()) {  
27            head = ndInput;  
28            tail = ndInput;  
29        } else {  
30            ndInput.next = head;  
31            head = ndInput;  
32        }  
33    }  
34
```

```
35    void addLast(int input) {  
36        Node15 ndInput = new Node15(input, null);  
37        if (isEmpty()) {  
38            tail.next = ndInput;  
39            head = tail = ndInput;  
40        } else {  
41            tail.next = ndInput;  
42            tail = ndInput;  
43        }  
44    }  
45  
46    void insertAfter(int key, int input) {  
47        Node15 ndInput = new Node15(input, null);  
48        Node15 temp = head;  
49        while (temp != null) {  
50            if (temp.data == key) {  
51                ndInput.next = temp.next;  
52                temp.next = ndInput;  
53                if (ndInput.next == null) {  
54                    tail = ndInput;  
55                }  
56                break;  
57            }  
58            temp = temp.next;  
59        }  
60    }  
61  
62    void insertAt(int index, int input) {  
63        if (index < 0) {  
64            System.out.println("Indeks tidak valid");  
65        } else if (index == 0) {  
66            addFirst(input);  
67        } else {  
68            Node15 temp = head;  
69            for (int i = 0; i < index - 1 && temp != null; i++) {  
70                temp = temp.next;  
71            }  
72            temp.next = new Node15(input, temp.next);  
73            if (temp.next.next == null) {  
74                tail = temp.next;  
75            }  
76        }  
77    }  
78 }  
79
```

```
1 package P10;  
2  
3 public class SLMain15 {  
4     public static void main(String[] args) {  
5         SingleLinkedList15 singLL = new SingleLinkedList15();  
6  
7         singLL.print();  
8         singLL.addFirst(890);  
9         singLL.print();  
10        singLL.addLast(760);  
11        singLL.print();  
12        singLL.addFirst(700);  
13        singLL.print();  
14        singLL.insertAfter(700, 999);  
15        singLL.print();  
16        singLL.insertAt(3, 833);  
17        singLL.print();  
18    }  
19 }  
20  
21
```

```
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
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : LINKED LIST

1. 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.



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
MATERI : LINKED LIST

Percobaan 2

```
1  int getData(int index) {
2      Node15 tmp = head;
3      for (int i = 0; i < index; i++) {
4          tmp = tmp.next;
5      }
6      return tmp.data;
7  }
8
9  int indexOf(int key) {
10     Node15 tmp = head;
11     int index = 0;
12     while (tmp != null && tmp.data != key) {
13         tmp = tmp.next;
14         index++;
15     }
16     if (tmp == null) {
17         return -1;
18     } else {
19         return index;
20     }
21 }
22
23 void removeFirst() {
24     if (isEmpty()) {
25         System.out.println("Linked List masih kosong, tidak dapat dihapus");
26     } else if (head == tail) {
27         head = tail = null;
28     } else {
29         head = head.next;
30     }
31 }
32
33 void removeLast() {
34     if (isEmpty()) {
35         System.out.println("Linked list masih kosong, tidak dapat dihapus");
36     } else if (head == tail) {
37         head = tail = null;
38     } else {
39         Node15 temp = head;
40         while (temp.next != tail) {
41             temp = temp.next;
42         }
43         temp.next = null;
44         tail = temp;
45     }
46 }
47
48 void remove(int key) {
49     if (isEmpty()) {
50         System.out.println("Linked list masih kosong, tidak dapat dihapus");
51     } else {
52         Node15 temp = head;
53         while (temp != null) {
54             if (temp.data == key && temp == head) {
55                 removeFirst();
56                 break;
57             } else if (temp.next.data == key) {
58                 temp.next = temp.next.next;
59                 if (temp.next == null) {
60                     tail = temp;
61                 }
62                 break;
63             }
64             temp = temp.next;
65         }
66     }
67 }
68
69 void removeAt(int index) {
70     if (index < 0) {
71         System.out.println("Indeks nya salah");
72     } else if (index == 0) {
73         removeFirst();
74     } else {
75         Node15 temp = head;
76         for (int i = 0; i < index - 1; i++) {
77             if (temp.next == null) {
78                 temp = temp.next;
79             }
80         }
81         temp.next = temp.next.next;
82         if (temp.next == null) {
83             tail = temp;
84         }
85     }
86 }
87 }
88
```



NAMA : HIZKIA ELSADANTA
NIM : 2341720253
KELAS : TI-1G
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.print();  
singLL.removeFirst();  
singLL.print();  
singLL.removeLast();  
singLL.print();  
}  
}
```

```
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 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_Structur_Data_1G_15.git