



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET IX - LINKED LIST

PERTEMUAN MINGGU 11 - LINKED LIST

Percobaan 1 : Pembuatan Single Linked List

Waktu percobaan : 30 menit

```
1 package Jobsheet_9;  
2  
3 public class Node07 {  
4     int data;  
5     Node07 next;  
6  
7     public Node07(int nilai, Node07 berikutnya) {  
8         data = nilai;  
9         next = berikutnya;  
10    }  
11 }
```

```
1 package Jobsheet_9;  
2  
3 public class SingleLinkedList07 {  
4     Node07 head, tail;  
5  
6     boolean isEmpty() {  
7         return head == null;  
8     }  
9     void print() {  
10        if (!isEmpty()) {  
11            Node07 tmp = head;  
12            System.out.print("Isi Linked List :\t");  
13            while (tmp != null) {  
14                System.out.print(tmp.data + "\t");  
15                tmp = tmp.next;  
16            } System.out.println(" ");  
17        } else {  
18            System.out.println("Linked List kosong");  
19        }  
20    }  
21    void addFirst(int input) {  
22        Node07 ndInput = new Node07(input, null);  
23        if (isEmpty()) {  
24            head = tail = ndInput;  
25        } else {  
26            ndInput.next = head;  
27            head = ndInput;  
28        }  
29    }  
30    void addLast(int input) {  
31        Node07 ndInput = new Node07(input, null);  
32        if (isEmpty()) {  
33            tail.next = ndInput;  
34            head = tail = ndInput;  
35        } else {  
36            tail.next = ndInput;  
37            tail = ndInput;  
38        }  
39    }  
40 }
```

```
38    }  
39    }  
40    void insertAfter(int key, int input) {  
41        Node07 ndInput = new Node07(input, null);  
42  
43        if (!isEmpty()) {  
44            Node07 temp = head;  
45            do {  
46                if (temp.data == key) {  
47                    ndInput.next = temp.next;  
48                    temp.next = ndInput;  
49                    if (ndInput.next == null) {  
50                        tail = ndInput;  
51                    } break;  
52                }  
53                temp = temp.next;  
54            } while (temp != null);  
55        } else {  
56            System.out.println("Linked List kosong");  
57        }  
58    }  
59    void insertAt(int index, int input) {  
60        if (index < 0) {  
61            System.out.println("Indeks tidak valid");  
62        } else if (index == 0) {  
63            addFirst(input);  
64        } else {  
65            Node07 temp = head;  
66            for (int i = 0; i < index - 1; i++) {  
67                temp = temp.next;  
68            }  
69            temp.next = new Node07(input, temp.next);  
70            if (temp.next.next == null) {  
71                tail = temp.next;  
72            }  
73        }  
74    }  
75 }
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET IX - LINKED LIST

```
1 package Jobsheet_9;  
2  
3 public class SLLMain07 {  
4     public static void main(String[] args) {  
5         SingleLinkedList07 singLL = new SingleLinkedList07();  
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 }
```

Verifikasi Hasil Percobaan

```
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 C:\Users\Admin\OneDrive - ypt.or.id\Documents\Kuliah\Semester  
a_07>
```

Pertanyaan

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

Jawab : Hasil compile kode program akan menghasilkan “Linked List Kosong” karena method print memeriksa apakah linked list dalam keadaan kosong atau tidak. Saat linked list pertama kali dibuat, maka belum ada node yang ditambahkan sehingga linked list masih dalam keadaan kosong. Sehingga saat pertama kali linked list di print, linked list dalam keadaan kosong.

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

Jawab : Variable temp digunakan sebagai variabel sementara untuk menampung node sementara saat melakukan iterasi atau pencarian dalam linked list. yang akan ditambahkan pada elemen linked list



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET IX - LINKED LIST

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

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

Jawab : Kode tersebut pada method insertAt digunakan untuk memperbarui nilai tail jika node baru yang dimasukkan menjadi node terakhir dalam linked list. Maka temp.next nya adalah null, maka tail harus diperbarui menjadi node baru tersebut. Sehingga kode tersebut memastikan agar tail selalu menunjuk ke node terakhir dalam linked list.

Percobaan 2 : Modifikasi Elemen pada Single Linked List

Waktu percobaan : 30 menit

```
1 int getData(int index) {
2     Node07 tmp = head;
3     for (int i = 0; i < index - 1; i++) {
4         tmp = tmp.next;
5     }
6     return tmp.next.data;
7 }
8 int indexOf(int key) {
9     Node07 tmp = head;
10    int index = 0;
11    while (tmp != null && tmp.data != key) {
12        tmp = tmp.next;
13        index++;
14    }
15    if (tmp != null) {
16        return index;
17    } else {
18        return -1;
19    }
20 }
21 void removeFirst() {
22     if (isEmpty()) {
23         System.out.println("Linked list masih kosong, tidak dapat dihapus");
24     } else if (head == tail) {
25         head = tail = null;
26     } else {
27         head = head.next;
28     }
29 }
30 void removeLast() {
31     if (isEmpty()) {
32         System.out.println("Linked list masih kosong, tidak dapat dihapus");
33     } else if (head == tail) {
34         head = tail = null;
35     } else {
36         Node07 tmp = head;
37         while (tmp.next != tail) {
38             tmp = tmp.next;
39         }
40         tmp.next = null;
41         tail = tmp;
```

```
41         tail = tmp;
42     }
43 }
44 void remove(int key) {
45     if (isEmpty()) {
46         System.out.println("Linked list masih kosong, tidak dapat dihapus");
47     } else {
48         Node07 tmp = head;
49         while (tmp != null) {
50             if (tmp.data == key && tmp == head) {
51                 removeFirst();
52                 break;
53             } else if (tmp.next.data == key) {
54                 tmp.next = tmp.next.next;
55                 if (tmp.next == null) {
56                     tail = tmp;
57                 }
58                 break;
59             }
60             tmp = tmp.next;
61         }
62     }
63 }
64 public void removeAt(int index) {
65     if (index < 0) {
66         System.out.println("Indeks tidak valid");
67     } else if (index == 0) {
68         removeFirst();
69     } else {
70         Node07 tmp = head;
71         for (int i = 0; i < index - 1; i++) {
72             tmp = tmp.next;
73         }
74         tmp.next = tmp.next.next;
75         if (tmp.next == null) {
76             tail = tmp;
77         }
78     }
79 }
```

```
1 System.out.println("Data pada indeks ke-1 = " + singLL.getData(1));
2 System.out.println("Data 3 berada pada indeks ke-" + singLL.indexOf(760));
3
4 singLL.remove(999);
5 singLL.print();
6 singLL.removeAt(0);
7 singLL.print();
8 singLL.removeFirst();
9 singLL.print();
10 singLL.removeLast();
11 singLL.print();
```



NAMA : AYLEEN RUHUL QISTHY
NIM : 2341720012
KELAS : 1G
MATERI : JOBSHEET IX - LINKED LIST

Verifikasi Hasil Percobaan

```
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
PS C:\Users\Admin\OneDrive - ypt.or.id\Documents\Kuliah\Semester 2>
rData_07>
```

Pertanyaan

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

Jawab : break pada remote digunakan untuk menghentikan iterasi setelah node yang dicari sudah ditemukan dan node tersebut akan dihapus. Jika break di hapus, maka iterasi akan terus berjalan walaupun node yang dicari sudah ditemukan. Sehingga dapat menyebabkan kesalahan penghapusan node.

2. Jelaskan kegunaan kode di bawah pada method remove!

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

Jawab : Kode tersebut digunakan untuk memeriksa apakah node setelah node saat ini (tmp.next) memiliki nilai data yang sama dengan key yang dicari. Kemudian memeriksa apakah node yang dihapus merupakan node terakhir dari linked list, jika benar, maka tail diperbarui untuk menunjuk ke node sebelum node terakhir, karena node terakhir telah dihapus.

Link GitHub Pribadi :

https://github.com/ayleenrq/Prak_AlgoritmaStrukturData_07.git