

Tugas remove head, tail & mid
Pertemuan 5
Praktikum Pemrograman 1

Nama : Fawwas Nawwaf Sabil

NRP : 223040114

Kelas : B

Link commit git hub

https://github.com/FawwasSabil25/PBO_223040114/commit/0a603938304d0702a088015e0b2cdc9070bc6f28

Tugas 1 (remove head & tail)

StrukturList.java

//remove head



```
67
68 //delete head
69 public void removeHead(){
70     if(isEmpty()){
71         System.out.println("List Kosong");
72     }else{
73         Node temp = HEAD;
74         HEAD = HEAD.getNext();
75         dispose(temp);
76     }
77 }
78
79 //fungsi dispose
80 private void dispose(Node temp) {
81     temp.setNext(null);
82     temp = null;
83 }
```

Delete head : Head pada list akan di assign mnjadi temporary lalu status head akan bergeser ke node selanjutnya. Lalu Temporary akan dihapus menggunakan fungsi dispose yang meng set value temp menjadi null

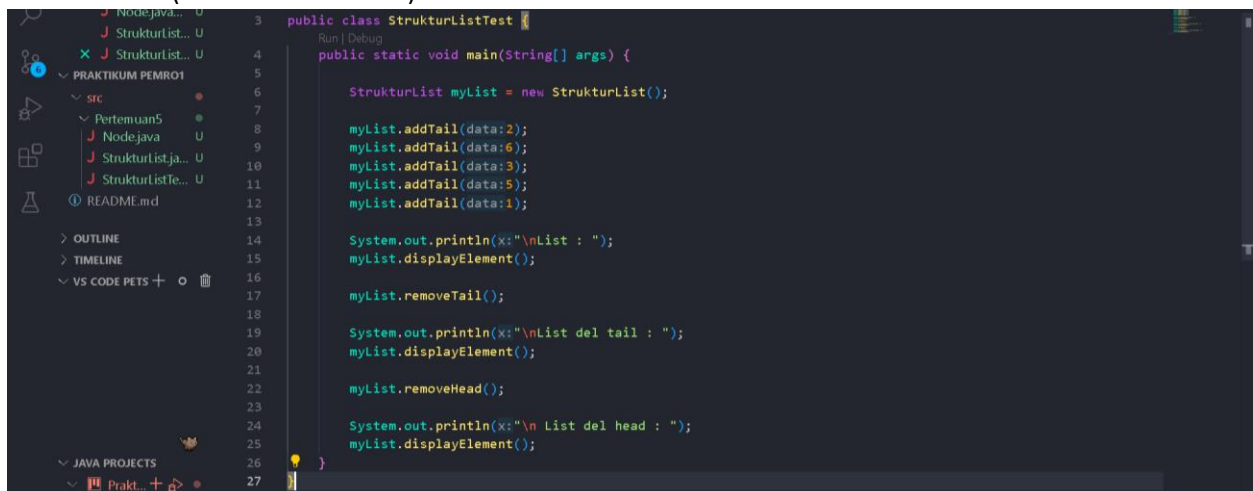
//remove tail



```
86
87 public void removeTail(){
88     Node preNode = null;
89     Node lastNode;
90
91     if (HEAD != null) {
92         if (HEAD.getNext() == null) {
93             HEAD = null;
94         } else {
95             lastNode = HEAD;
96             while (lastNode.getNext() != null) {
97                 preNode = lastNode;
98                 lastNode = lastNode.getNext();
99             }
100             preNode.setNext(next:null);
101             dispose(lastNode);
102         }
103     }
```

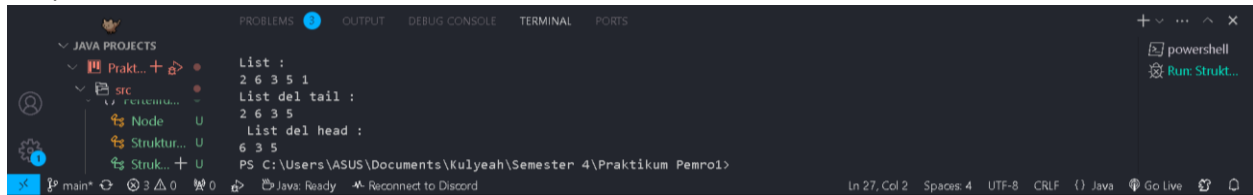
Fungsi akan mencari node terakhir menggunakan prenode = lastnode dan lastnode. Ketika node terakhir ditemukan akan memutus hubungan dengan node lainnya dengan prenode = setnext(null) sehingga last node tidak berhubungan lagi. Lalu di dispose (dihapus)

Latihan 1 & 2 (remove head & tail)



```
3 public class StrukturListTest
4 {
5     public static void main(String[] args) {
6
7         StrukturList myList = new StrukturList();
8
9         myList.addTail(data:2);
10        myList.addTail(data:6);
11        myList.addTail(data:3);
12        myList.addTail(data:5);
13        myList.addTail(data:1);
14
15        System.out.println(x:"\nList : ");
16        myList.displayElement();
17
18        myList.removeTail();
19
20        System.out.println(x:"\nList del tail : ");
21        myList.displayElement();
22
23        myList.removeHead();
24
25        System.out.println(x:"\n List del head : ");
26        myList.displayElement();
27    }
28 }
```

Output :



```

List :
2 6 3 5 1
List del tail :
2 6 3 5
List del head :
6 3 5
PS C:\Users\ASUS\Documents\Kulyeah\Semester 4\Praktikum Pemro1>

```

Tugas 2 (remove mid)

StrukturList.java



```

public class StrukturList {
    public void removeMid(int e){
        Node preNode = new Node(data:0);
        Node tempNode;
        int i;
        boolean ketemu;

        if(isEmpty()){
            System.out.println("List Kosong");
        }else{
            ketemu = false;
            i = 1;
            tempNode = HEAD;

            while(tempNode.getNext() != null && !ketemu){
                if(tempNode.getData() == e){
                    ketemu = true;
                }else{
                    preNode = tempNode;
                    tempNode = tempNode.getNext();
                    i++;
                }
            }

            if(ketemu == true){
                if(i == 1){
                    HEAD = null;
                }else{
                    preNode.setNext(tempNode.getNext());
                    dispose(tempNode);
                }
            }
        }
    }
}

```

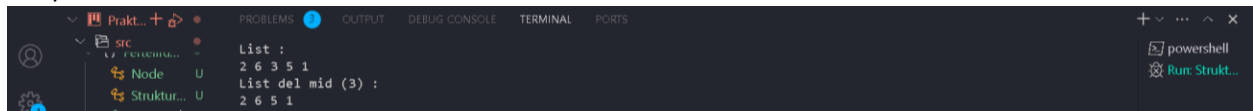
e digunakan untuk input nilai yang akan dicari dalam list. Tempnode digunakan untuk mencari nilai yg sama dengan e didalam list dengan tempnode.getdata() == e, jika data tidak ditemukan, prenode akan diupdate untk merujuk ke tempnode. Dan tempnode bergerak lagi mencari (i++). Ketika nilai ketemu (ketemu == true). Jika posisi pertama yg dihapus maka head = null. Jika bukan maka preNode.setNext(tempNode.getNext()) dipanggil. Ini memutus tautan dari preNode ke tempNode (node yang akan dihapus) dan langsung menghubungkannya ke node setelah tempNode. Lalu dihapus menggunakan dispose

Latihan remove mid



```
1 public class StrukturListTest {
2     public static void main(String[] args) {
3
4         StrukturList myList = new StrukturList();
5
6         myList.addTail(data:2);
7         myList.addTail(data:6);
8         myList.addTail(data:3);
9         myList.addTail(data:5);
10        myList.addTail(data:1);
11
12        System.out.println("x:\nList : ");
13        myList.displayElement();
14
15        myList.removeMid(e:3);
16
17        System.out.println("x:\nList del mid (3) : ");
18        myList.displayElement();
19    }
20 }
21
22 }
```

Output:



```
1 List :
2 2 6 3 5 1
3 List del mid (3) :
4 2 6 5 1
```