

Tugas 4
Praktikum Pemrograman 1

Nama : Fawwas Nawwaf Sabil
NRP : 223040114
Kelas : D

Link repository gitHubPraktikum Pemrograman :
https://github.com/FawwasSabil25/PrakPrmro1_223040114.git

Latihan 1

```
1 package Pertemuan4;
2
3 public class Node {
4     private double data; //untuk tipe data hanya mengganti int menjadi double atau sebaliknya
5     private Node next;
6
7     /* inialisasi atribut node */
8     public Node (double data2) { //mengganti int menjadi double atau sebaliknya
9         this.data = data2;
10    }
11
12    /* setter & getter */
13    /* getter Data */
14    public double getData() { //mengganti int menjadi double atau sebaliknya
15        return data;
16    }
17
18    /* setter data */
19    public void setData(double data) { //mengganti int menjadi double atau sebaliknya
20        this.data = data;
21    }
22
23    /* getter next */
24    public Node getNext() {
25        return next;
26    }
```

```
28    /* setter next */
29    public void setNext(Node next) {
30        this.next = next;
31    }
32 }
33
```

Latihan 2

```

1 package Pertemuan4;
2
3 public class StrukturList {
4     private Node HEAD;
5
6     //add head (menambahkan node di awal list)
7     public void addHead(int data) {
8         Node newNode = new Node(data);
9
10        if(isEmpty()) {
11            HEAD = newNode;
12        }else {
13            newNode.setNext(HEAD);
14            HEAD = newNode;
15        }

```

AddHead, fungsi untuk menambahkan node diawal list. Dimana kita membuat data baru dan menyimpannya dalam variable newNode. Lalu mengecek apakah linked list kosong, jika iya maka "HEAD" (ujung awal list) dimasukkan ke dalam variable newNode. Karena newNode merupakan variable pertama dalam list. Lalu jika linked list tidak kosong, newNode dihubungkan dengan simpul yang saat ini menjadi HEAD. Selanjutnya, HEAD diatur menjadi variabel baru tersebut, sehingga newNode menjadi ujung awal linked list.

```

18 //add mid (menyelipkan node di tengah tengah list)
19 public void addMid(int data, int position) {
20     Node newNode = new Node(data);
21     Node curNode = null;
22     Node posNode = null;
23     int i;
24
25     if(isEmpty()) {
26         HEAD = newNode;
27     }else {
28         curNode = HEAD;
29         if(position == 1) {
30             newNode.setNext(curNode);
31             HEAD = newNode;
32         }else {
33             i = 1;
34             while((curNode != null)&&(i < position)) {
35                 posNode = curNode;
36                 curNode = curNode.getNext();
37                 i++;
38             }
39             posNode.setNext(newNode);
40             newNode.setNext(curNode);
41         }
42     }
43 }

```

addMid, untuk menambahkan simpul baru di Tengah Tengah list. Fungsi ini Mengecek apakah linked list kosong. Jika iya, atur HEAD ke simpul baru. Jika linked list tidak kosong, lakukan langkah-langkah penambahan di tengah-tengah :

Jika position == 1, artinya ingin memasukkan simpul baru di awal linked list, maka atur HEAD ke simpul baru, dan hubungkan simpul baru ke simpul yang awalnya di HEAD.

Jika position bukan 1, lakukan iterasi untuk menemukan posisi yang diinginkan:

Selama belum mencapai posisi yang diinginkan dan belum sampai akhir linked list ($\text{curNode} \neq \text{null}$ dan $i < \text{position}$), lakukan iterasi ke simpul berikutnya.

Selama iterasi, simpan simpul saat ini (curNode) dan simpul sebelumnya (posNode). Setelah menemukan posisi yang diinginkan, hubungkan simpul baru ke simpul sebelumnya (posNode), dan hubungkan simpul baru ke simpul yang sebelumnya ada di posisi tersebut (curNode).

```
45 //add tail (menambahkan node di akhir list)
46 public void addTail(int data) {
47     Node newNode = new Node(data);
48
49     if (isEmpty()) {
50         HEAD = newNode;
51     }else {
52         Node posNode = null;
53         Node curNode = HEAD;
54
55         while(curNode != null) {
56             posNode = curNode;
57             curNode = curNode.getNext();
58         }
59
60         posNode.setNext(newNode);
61     }
62 }
63
64 private boolean isEmpty() {
65     return HEAD == null;
66 }
67
```

`addTail`, menambahkan simpul di akhir list, dengan :

Mengecek apakah linked list kosong. Jika iya, atur HEAD ke simpul baru (`newNode`).

Jika linked list tidak kosong, lakukan langkah-langkah penambahan di ujung (ekor):

Inisialisasi dua variabel, `posNode` (simpul sebelum ujung) dan `curNode` (variabel iterasi yang dimulai dari HEAD).

Iterasi melalui linked list sampai mencapai ujung (sampai `curNode` menjadi null).

Setelah iterasi selesai, `posNode` menunjuk ke simpul terakhir di linked list.

Hubungkan simpul baru (`newNode`) ke simpul terakhir (`posNode`).

```
63
64 private boolean isEmpty() {
65     return HEAD == null;
66 }
67
68 public void displayElement() { //untuk menampilkan hasil ke layar
69     Node curNode = HEAD;
70
71     while(curNode != null) {
72         System.out.print(curNode.getData() + " ");
73         curNode = curNode.getNext();
74     }
75 }
76 }
77
```

Latihan 3

```
1 package Pertemuan4;
2
3 public class StrukturListTest {
4
5     Run | Debug
6     public static void main(String[] args) {
7
8         StrukturList myList = new StrukturList();
9
10        myList.addTail(data:3);
11        myList.addTail(data:4);
12        myList.addMid(data:7, position:2);
13        myList.addMid(data:8, position:2);
14        myList.addHead(data:5);
15        myList.displayElement();
16    }
17
18 }
19
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS Run: StrukturListTest + v

ExceptionMessages' '-cp' 'C:\Users\ASUS\Documents\Kulyeah\Semester 4\Praktikum Pemro1\bin' 'Pertemuan4.st'

5 3 8 7 4

PS C:\Users\ASUS\Documents\Kulyeah\Semester 4\Praktikum Pemro1>

Urutan dari pen inputan data tidak mempengaruhi hasil, karena tiap data sudah di assign ke lokasinya masing masing sesuai dengan fungsi yang digunakan, addHead untuk didepan, addMid untuk ditengah, dan addTail untuk diakhir

Tugas 1

Didalam file Node.java kita hanya perlu mengganti tipe data int menjadi double

```
1 package Pertemuan4;
2
3 public class Node {
4     private double data; //untuk tipe data hanya mengganti int menjadi double atau sebaliknya
5     private Node next;
6
7     /* inisialisasi atribut node */
8     public Node (double data2) { //mengganti int menjadi double atau sebaliknya
9         this.data = data2;
10    }
11
12    /* setter & getter */
13    /* getter Data */
14    public double getData() { //mengganti int menjadi double atau sebaliknya
15        return data;
16    }
17 }
```

```

22
23     /* getter next */
24     public Node getNext() {
25         return next;
26     }
27
28     /* setter next */
29     public void setNext(Node next) {
30         this.next = next;
31     }
32 }

```

Dan saya membuat file baru StrukturListDouble Dimana semua isi dari file tersebut sama persis dengan file StrukturList hanya saja semua tipe data diganti dari int menjadi double

```

1  package Pertemuan4;
2
3  public class StrukturListDouble {
4      private Node HEAD;
5
6      //add head
7      public void addHead(double data) { //int digantikan menjadi double
8          Node newNode = new Node(data);
9
10         if(isEmpty()) {
11             HEAD = newNode;
12         }else {
13             newNode.setNext(HEAD);
14             HEAD = newNode;
15         }
16     }
17

```

```

18     //add mid
19     public void addMid(double data, int position) { //int digantikan menjadi double
20         Node newNode = new Node(data);
21         Node curNode = null;
22         Node posNode = null;
23         int i;
24
25         if(isEmpty()) {
26             HEAD = newNode;
27         }else {
28             curNode = HEAD;
29             if(position == 1) {
30                 newNode.setNext(curNode);
31                 HEAD = newNode;
32             }else {
33                 i = 1;
34                 while((curNode != null)&&(i < position)) {
35                     posNode = curNode;
36                     curNode = curNode.getNext();
37                     i++;
38                 }
39                 posNode.setNext(newNode);
40                 newNode.setNext(curNode);
41             }
42         }
43     }

```

```

45 //add tail
46 public void addTail(double data) { //int digantikan menjadi double
47     Node newNode = new Node(data);
48
49     if (isEmpty()) {
50         HEAD = newNode;
51     }else {
52         Node posNode = null;
53         Node curNode = HEAD;
54
55         while(curNode != null) {
56             posNode = curNode;
57             curNode = curNode.getNext();
58         }
59
60         posNode.setNext(newNode);
61     }
62 }
63
64
65 private boolean isEmpty() {
66     return HEAD == null;
67 }
68
69
70 //latihan 3
71 public void displayElement() {
72     Node curNode = HEAD;
73
74     while(curNode != null) {
75         System.out.print(curNode.getData()+ " ");
76         curNode = curNode.getNext();
77     }
78 }
79
80

```

Tugas 2

a.

```

1 package Pertemuan4;
2
3 public class StrukturListDoubleTest {
4     Run | Debug
5     public static void main(String[] args) {
6         StrukturListDouble myList2 = new StrukturListDouble();
7
8         myList2.addHead(data:2.1);
9         myList2.addTail(data:3.4);
10        myList2.addTail(data:4.5);
11        myList2.displayElement();
12    }
13 }

```

Hasil nya :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

raktikum Pemro1\bin' 'Pertemuan4.StrukturListDoubleTest'
2.1 3.4 4.5
PS C:\Users\ASUS\Documents\Kulyeah\Semester 4\Praktikum Pemro1>
```

Bisa saja menggunakan addTails semua, atau bisa jg di persulit dengan mengacak urutan sehingga harus menggunakan fungsi yang tepat. Tapi terlihat 2.1 ada di awal list, 3.4 ada di Tengah, dan 4.5 ada diakhir

b.

```
1 package Pertemuan4;
2
3 public class StrukturListDoubleTest2 {
4     Run | Debug
5     public static void main(String[] args) {
6         StrukturListDouble myList3 = new StrukturListDouble();
7
8         myList3.addTail(data:4.5);
9         myList3.addTail(data:5.5);
10        myList3.addHead(data:3.4);
11        myList3.addMid(data:2.1, position:2);
12        myList3.addMid(data:1.1, position:3);
13        myList3.displayElement();
14    }
15 }
```

```
ah\Semester 4\Praktikum Pemro1'; & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-XX:+ShowCodeDetailsIn
ExceptionMessages' '-cp' 'C:\Users\ASUS\Documents\Kulyeah\Semester 4\Praktikum Pemro1\bin' 'Pertemuan4
.StrukturListDoubleTest2'
3.4 2.1 1.1 4.5 5.5
PS C:\Users\ASUS\Documents\Kulyeah\Semester 4\Praktikum Pemro1>
```

dy Connected to Discord Ln 15, Col 1 Spaces: 4 UTF-8 CRLF {} Java

Disini saya mencoba untuk mengacak urutan penulisan namun tetap sesuai dengan fungsinya untuk menentukan hasil yang serupa dengan apa yang ditugaskan. Benar saja hasilnya adalah

3.4 2.1 1.1 4.5 5.5