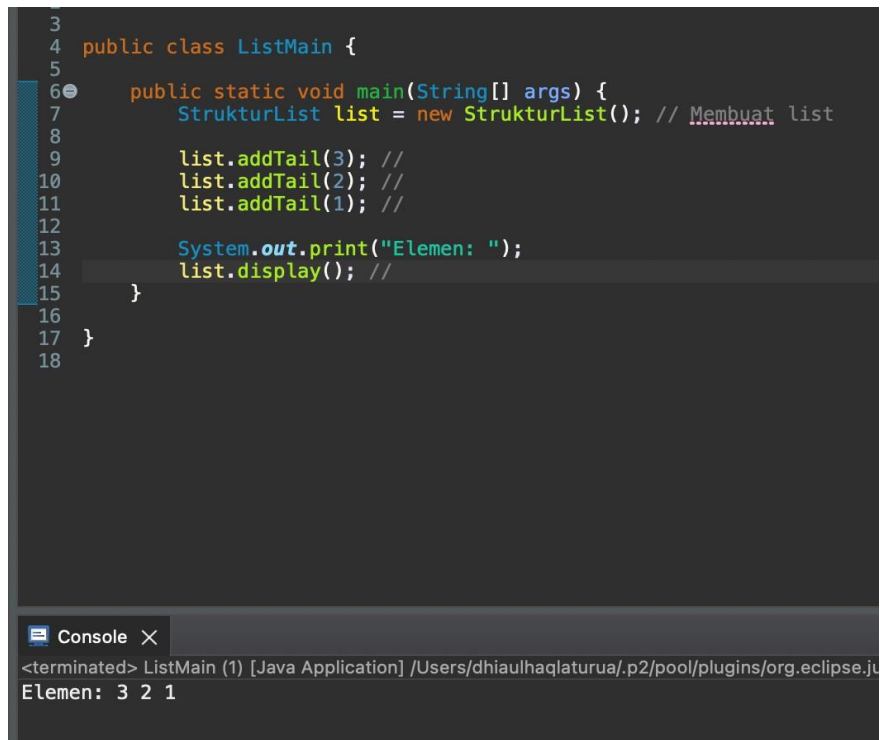


# PRAKTIKUM PEMOGRAMAN I

Nama : Roy Martin Silaban

Nrp : 233040130

Kelas : A



```
3
4 public class ListMain {
5
6     public static void main(String[] args) {
7         StrukturList list = new StrukturList(); // Membuat list
8
9         list.addTail(3); //
10        list.addTail(2); //
11        list.addTail(1); //
12
13        System.out.print("Elemen: ");
14        list.display(); //
15    }
16 }
17
18
```

Console X

<terminated> ListMain (1) [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.ju  
Elemen: 3 2 1

1. A

Penjelasan : Pada Tes-1 a, program membuat sebuah linked list menggunakan kelas StrukturList. Metode addTail() digunakan untuk menambahkan elemen ke akhir list. Elemen yang dimasukkan adalah 3, 2, dan 1 secara berurutan. Karena metode ini menambahkan elemen di bagian akhir, maka saat display() dipanggil, list akan ditampilkan sesuai urutan penambahan: **3 2 1**.

```
3
4 public class ListMain {
5
6     public static void main(String[] args) {
7         StrukturList list = new StrukturList();
8
9         list.addTail(1);
10        list.addTail(4);
11        list.addTail(5);
12        list.addTail(7);
13
14        System.out.print("Elemen: ");
15        list.display();
16    }
17
18 }
19
```

Console X

<terminated> ListMain (1) [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.ju  
Elemen: 1 4 5 7

1. B

Penjelasan : Pada Tes-1 b, program yang sama digunakan tetapi dengan elemen yang berbeda, yaitu 1, 4, 5, dan 7. Setiap elemen baru ditambahkan di bagian akhir list. Oleh karena itu, saat display() dipanggil, elemen ditampilkan dalam urutan **1 4 5 7**, sesuai dengan urutan penambahan.

```
20
21 // ==== Tes-2 ====
22 System.out.println("Tes-2:");
23 StrukturList list1 = new StrukturList();
24
25 list1.addHead(5);
26 list1.addHead(4);
27 list1.addHead(3);
28
29 System.out.print("Output: ");
30 list1.display(); // Output yang diharapkan: 3 4 5 3 4 5
31
32 System.out.println("\n-----");
33
34 // ==== Tes-3 (a) ====
35 System.out.println("Tes-3 (a):");
36 StrukturList list2 = new StrukturList();
37
38 list2.addHead(1);
39 list2.addHead(2);
40 list2.addHead(3);
41
42 System.out.print("Output: ");
43 list2.display(); // Output yang diharapkan: 3 2 1
44
45 System.out.println("\n-----");
46
47 // ==== Tes-3 (b) ====
48 System.out.println("Tes-3 (b):");
49 StrukturList list3 = new StrukturList();
50
51 list3.addHead(7);
52 list3.addHead(5);
53 list3.addHead(4);
54 list3.addHead(1);
55
56 System.out.print("Output: ");
57 list3.display(); // Output yang diharapkan: 1 4 5 7
58 }
59 }
60
61
```

Console X

<terminated> ListMain (1) [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.j

Tes-1:  
Elemen: 1 4 5 7

-----

Tes-2:  
Output: 3 4 5

-----

Tes-3 (a):  
Output: 3 2 1

-----

Tes-3 (b):  
Output: 1 4 5 7

2.

Penjelasan : Pada Tes-2, program membuat linked list `list1` dan menambahkan elemen menggunakan `addHead()`, yang menempatkan elemen baru di awal list. Urutan penambahan:

1. `addHead(3)` ; → 3 jadi HEAD
2. `addHead(4)` ; → 4 jadi HEAD, 3 bergeser
3. `addHead(5)` ; → 5 jadi HEAD, 4 → 3 bergeser

Karena elemen terakhir yang ditambahkan berada di depan, outputnya: **3 4 5**.

3. A

```
20
21 // === Tes-2 ===
22 System.out.println("Tes-2:");
23 StrukturList list1 = new StrukturList();
24
25 list1.addHead(5);
26 list1.addHead(4);
27 list1.addHead(3);
28
29 System.out.print("Output: ");
30 list1.display(); // Output yang diharapkan: 3 4 5 3 4 5
31
32 System.out.println("\n-----");
33
34 // === Tes-3 (a) ===
35 System.out.println("Tes-3 (a):");
36 StrukturList list2 = new StrukturList();
37
38 list2.addHead(1);
39 list2.addHead(2);
40 list2.addHead(3);
41
42 System.out.print("Output: ");
43 list2.display(); // Output yang diharapkan: 3 2 1
44
45 System.out.println("\n-----");
46
47 // === Tes-3 (b) ===
48 System.out.println("Tes-3 (b):");
49 StrukturList list3 = new StrukturList();
50
51 list3.addHead(7);
52 list3.addHead(5);
53 list3.addHead(4);
54 list3.addHead(1);
55
56 System.out.print("Output: ");
57 list3.display(); // Output yang diharapkan: 1 4 5 7
58
59 }
60
61
```

Console X

<terminated> ListMain (1) [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.j

Tes-1:  
Elemen: 1 4 5 7

-----

Tes-2:  
Output: 3 4 5

-----

Tes-3 (a):  
Output: 3 2 1

-----

Tes-3 (b):  
Output: 1 4 5 7

Penjelasan : Tes-3 (a) menguji skenario di mana elemen dimasukkan menggunakan metode `addHead()` dengan urutan input **3, 2, 1**. Karena `addHead()` menambahkan elemen di bagian depan list, elemen terakhir yang dimasukkan (1) akan menjadi elemen pertama dalam list. Jika di gambarkan 3 -> null, setelah itu 2 -> 3 -> null, setelah itu 1 -> 2 -> 3 -> null, dan outputnya 1 2 3.

### 3.B

Penjelasan Pada Tes-3 (b), elemen dimasukkan ke dalam list menggunakan metode `addHead()`, tetapi dengan urutan input **1, 4, 5, 7**. Karena metode `addHead()` menambahkan elemen di depan list setiap kali dipanggil, elemen terakhir yang dimasukkan akan berada di posisi pertama. Jika di gambarkan 1 -> null, setelah itu 4 -> 1 -> null, setelah itu 5 -> 4 -> 1 -> null, setelah itu 7 -> 5 -> 4 -> 1 -> null. Maka Outputnya 7 5 4 1.