

Nama : Muhammad Dhiaulhaq Triyudhistira Laturua
Npm ; 233040171
Kelas : IF- A
Matkul : Praktikum Pemograman I

Latihan 1

```
public class StrukturList {  
    Node HEAD;  
  
    public StrukturList() {  
        HEAD = null;  
    }  
  
    // latihan 1  
  
    public boolean find(int x) {  
        Node curNode = HEAD;  
        boolean ketemu = false;  
  
        while (curNode != null && !ketemu) {  
            if (curNode.getData() == x) {  
                ketemu = true;  
            } else {  
                curNode = curNode.getNext();  
            }  
        }  
        return ketemu;  
    }  
}
```

Kode ini merupakan bagian dari implementasi struktur data linked list dalam Java, di mana class StrukturList menyimpan referensi ke node pertama (disebut HEAD). Konstruktor akan menginisialisasi HEAD sebagai null, menandakan list masih kosong. Fungsi find(int x) digunakan untuk mencari apakah suatu nilai x terdapat dalam list, dengan cara menelusuri node satu per satu mulai dari HEAD. Jika data pada node cocok dengan nilai x, maka fungsi akan mengembalikan true, jika tidak ditemukan hingga akhir list, akan mengembalikan false.

Latihan 2

```
3 public class StrukturListTest {
4
5     // Latihan 2
6
7     public static void main(String[] args) {
8         StrukturList list = new StrukturList();
9
10        // 1. Create list dengan keyword new
11        list.HEAD = new Node(5);
12
13        // 2. Tambah elemen 5 di akhir list
14        Node second = new Node(4);
15        list.HEAD.setNext(second);
16
17        // 3. Tambah elemen 6 di akhir list
18        Node third = new Node(6);
19        second.setNext(third);
20
21        // 5. Tampilkan elemen list
22        Node currentNode = list.HEAD;
23        while (currentNode != null) {
24            System.out.print(currentNode.getData() + " ");
25            currentNode = currentNode.getNext();
26        }
27        System.out.println();
28
29        // 6. Cari elemen bernilai 6
30        boolean found = list.find(6);
31        System.out.println(found);
32
33
34
35
36    }
37 }
```

Console X

```
<terminated> StrukturListTest (2) [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_17.0.11.
5 4 6
true
```

Buat list secara manual dengan tiga node yang nilainya 5, 4, dan 6. Setelah itu hubungkan node-node itu satu per satu pakai method `setNext()`, jadi terbentuklah satu rangkaian linked list. Setelah itu, tampilkan semua isi list dengan perulangan `while`, dan hasilnya muncul di console: 5 4 6. Terakhir, cek apakah angka 6 ada di dalam list pakai method `find()`, dan karena memang ada, hasilnya `true`.

Latihan 3

```
// latihan 3

// Menambahkan elemen di awal list
● public void addFirst(int data) {
    Node newNode = new Node(data);
    newNode.next = HEAD;
    HEAD = newNode;
}

// Fungsi size() seperti pada algoritma
● public int size() {
    Node curNode = HEAD;
    int jumlah = 0;

    while (curNode != null) {
        jumlah++;
        curNode = curNode.next;
    }

    return jumlah;
}

// Menampilkan elemen dalam list
● public void printList() {
    Node curNode = HEAD;
    while (curNode != null) {
        System.out.print(curNode.data + " ");
        curNode = curNode.next;
    }
    System.out.println();
}
}
```

buat node baru, terus arahkan next-nya ke HEAD, lalu update HEAD ke node baru itu. Method size() gunanya buat ngitung berapa banyak node yang ada di list. Dia nge-loop dari HEAD sampai akhir list, terus tiap nemu node, variabel jumlah ditambahin satu. Terakhir, ada method printList() yang bakal nampilin semua data dalam list. Sama juga, dia pakai perulangan dari awal sampai habis, terus print data di tiap node.

Latihan 4

```
2
3 public class StrukturListMain {
4
5     // latihan4
6
7
8     public static void main(String[] args) {
9         // 1. Buat list baru
10        StrukturList list = new StrukturList();
11
12        // 2. Tambah elemen: 7, 6, 4, 2, 3 (di-depan sehingga urutan input terbalik)
13        list.addFirst(3);
14        list.addFirst(2);
15        list.addFirst(4);
16        list.addFirst(6);
17        list.addFirst(7);
18
19        // 3. Tampilkan elemen list
20        list.printList(); // Output: 7 6 4 2 3
21
22        // 4. Tampilkan size elemen list
23        System.out.println(list.size()); // Output: 5
24    }
25
26 }
27
28
```

Console X

```
<terminated> StrukturListMain [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx
7 6 4 2 3
5
```

Tugas

```
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class ListOperations {
7
8     public static void main(String[] args) {
9         // 1. Create list dengan keyword new
10        List<Integer> list = new ArrayList<>();
11
12        // 2. Tambah elemen sehingga elemen list berisi (4, 3, 7, 8, 1, 9)
13        list.add(4);
14        list.add(3);
15        list.add(7);
16        list.add(8);
17        list.add(1);
18        list.add(9);
19
20        // 3. Tampilkan elemen list
21        System.out.println("Elemen List: " + list);
22
23        // 4. Hapus semua elemen list
24        list.clear();
25
26        // 5. Tampilkan elemen list
27        if (list.isEmpty()) {
28            System.out.println("List Kosong");
29        } else {
30            System.out.println("Elemen List: " + list);
31        }
32    }
33 }
34
35
```

Console X

```
<terminated> ListOperations [Java Application] /Users/dhiaulhaqlaturua/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx
Elemen List: [4, 3, 7, 8, 1, 9]
List Kosong
```