

# Data Structure and Algorithm Practicum

## UJIAN AKHIR SEMESTER GENAP

### TAHUN AKADEMIK 2022/2023



**Name**

Muhammad Baihaqi Aulia Asy'ari

**NIM**

2241720145

**Class**

1I

**Department**

Information Technology

**Study Program**

D4 Informatics Engineering

---

## Soal UAS

Telah disediakan sebuah project tentang Double Linked List yang terdiri dari 3 class, yaitu class **Node2P**, class **DLL** dan class **DLLMain**. Project tersebut dibuat menggunakan Netbeans, jika Anda menggunakan Netbeans bisa langsung dibuka project tersebut, jika Anda menggunakan tools yang lain, bisa diambil saja class-class nya. Fungsi-fungsi utama terkait Double Linked List telah disediakan di dalam class **DLL**, seperti **isEmpty()**, **size()**, **addLast()**, **deleteFirst()**, **deleteLast()**, **deleteByData()**, **getDataByIndex()** dan **print()**. Method **deleteByData()** digunakan untuk menghapus elemen dari list berdasarkan nilai datanya. Sedangkan method **getDataByIndex()** digunakan untuk mengambil nilai data berdasarkan posisinya.

## Pertanyaan

Lengkapi kode program di class **DLL**, untuk method-method di bawah ini:

1. **toArray()**: method ini akan menghasilkan sebuah variable array, dimana datanya berasal dari data di dalam objek linked list yang sudah ada.

```
int[] toArray(){
    int dataCount = 0;
    Node2P current = head;
    if (isEmpty()) {
        dataCount = 0;
        System.out.println("list is empty");
    } else {
        dataCount = 1;
        do {
            dataCount++;
            current = current.next;
        } while (current.next != null);
    }
    int[] data = new int[dataCount];
    for (int i = 0; i < dataCount; i++) {
        data[i] = get(i);
    }
    return data;
}
```

2. **sublist(int start, int end)**: method ini digunakan untuk mengembalikan list baru yang mengambil sebagian dari data yang sudah ada di list dari posisi **start** sampai posisi **end**

---

```

DLL sublist(int start, int end){
    Node2P current = head;
    DLL data = new DLL();
    for (int i = 0; i <= end; i++) {
        if (i >= start) {
            data.addLast(get(i));
        }
        current = current.next;
    }
    return data;
}

```

3. **addAll(DLL list)**: method ini digunakan untuk menambahkan data yang ada di list ke dalam list yang sudah ada

```

void addAll(DLL list){
    Node2P currentList = list.head;
    if (isEmpty()) {
        System.out.println("List is empty");
    } else {
        do {
            addLast(currentList.data);
            currentList = currentList.next;
        } while (currentList != null);
    }
}

```

4. **containsAll(DLL list)**: method ini akan mengecek apakah semua data yang ada di dalam list, ada di dalam list yang sudah ada

```

boolean containsAll(DLL list){
    int[] thisList = toArray();
    int[] inputList = list.toArray();
    boolean bool3 = false;
    boolean[] bool2 = new boolean[inputList.length];
    for (int i = 0; i < inputList.length; i++) {
        boolean[] bool = new boolean[thisList.length];
        for (int j = 0; j < thisList.length; j++) {
            bool[j] = thisList[j] == inputList[i];
        }
        for (int j = 0; j < bool2.length; j++) {
            for (boolean b : bool) {
                if (b) {
                    bool2[j] = b;
                }
            }
        }
    }
    return bool3;
}

```

---

```

        }
    }
}
for (boolean b : bool2) {
    if (b) {
        bool3 = b;
    }
}
return bool3;
}

```

5. **removeAll(DLL list)**: method ini akan menghapus data dari dalam list yang sudah ada berdasarkan nilai yang ada di dalam **list**

```

void removeAll(DLL list){
    Node2P currentList = list.head;
    if (isEmpty()) {
        System.out.println("list is empty");
    } else {
        do {
            deleteByData(currentList.data);
            currentList = currentList.next;
        } while (currentList != null);
    }
}

```

Jika class DLLMain dijalankan, maka akan didapatkan hasil seperti di bawah ini:

```

1  Tampilan data awal DLL:
2  10-100-34-20-200-75
3  Tampilan data array hasil dari fungsi toArray():
4  10, 100, 34, 20, 200, 75,
5  Tampilan data dari list hasil dari fungsi sublist(1,3):
6  100-34-20
7  Tampilan data dari list hasil dari fungsi addAll():
8  10-100-34-20-200-75-212-212-212
9  Tampilan data dari fungsi containsAll():
10 true
11 Tampilan data dari fungsi removeAll():
12 10-20-200-75

```