

Worksheet pertemuan 6 - 1
Algoritma dan Struktur Data
LinkedList

NIM: 20523164

Nama: Fajrun Shubhi

A. Membuat Folder Untuk Menyimpan Hasil Praktikum

1. Siapkan folder kosong dengan nama menggunakan NIM masing-masing. Jika folder NIM pada pertemuan sebelumnya mau dimanfaatkan, jangan lupa pindahkan dulu isinya ke folder lain sebagai arsip.
2. Folder ini akan dijadikan tempat untuk menyimpan semua pdf dari worksheet ini beserta file praktikum lainnya.

B. Membuat class LinkedList

1. Silakan buat sebuah class dengan nama **LinkedList**
2. Kemudian salin tempel kode program di bawah ini

```
package linkedlist;

/**
 *
 * @param <E>
 */
public class LinkedList<E> {
    Element first;
    Element last;
    int size = 0;

    private static final class Element<E>{
        E data;
        Element next;
        Element previous;

        Element(E data){
            this.data = data;
        }
    }

    public LinkedList(){

    }

    public Object getFirst(){
        if(size == 0){
            return null;
        }else{
```

```

        return first.data;
    }
}

public Object getLast(){
    if(size == 0){
        return null;
    }else{
        return last.data;
    }
}

public Object get(int index){
    if (index < 0 || index > size){
        System.out.println("Index out of bounds");
        System.exit(-1);
    }
    return getElement(index).data;
}

Element getElement(int n){
    Element e;
    if (n < size / 2){
        e = first;
        // n less than size/2, iterate from start
        while (n-- > 0){
            e = e.next;
        }
    }else{
        e = last;
        // n greater than size/2, iterate from end
        while (++n < size){
            e = e.previous;
        }
    }
    return e;
}

void removeElement(Element e){
    size--;
    if (size == 0){
        first = last = null;
    }else{
        if (e == first){
            first = e.next;
            e.next.previous = null;
        }else if(e == last){

```

```

        last = e.previous;
        e.previous.next = null;
    }else{
        e.next.previous = e.previous;
        e.previous.next = e.next;
    }
}

}

public boolean remove(Object o){
    Element e = first;

    while (e != null){
        if (o.equals(e.data)){
            removeElement(e);
            return true;
        }
        e = e.next;
    }
    return false;
}

public Object remove(int index){
    if (index < 0 || index > size){
        System.out.println("Index out of bounds");
        System.exit(-1);
    }
    Element e = getElement(index);
    removeElement(e);
    return e.data;
}

public Object remove(){
    return removeFirst();
}

public Object removeFirst(){
    if(size == 0){
        return null;
    }else{
        size--;
        Object removed = first.data;

        if (first.next != null){
            first.next.previous = null;
        }else{
            last = null;
        }
    }
}

```

```

        }
        first = first.next;
        return removed;
    }
}

public Object removeLast(){
    if(size == 0){
        return null;
    }else{
        size--;
        Object removed = last.data;

        if (last.previous != null){
            last.previous.next = null;
        }else{
            first = null;
        }
        last = last.previous;
        return removed;
    }
}

public void add(E obj){
    addLast(obj);
}

public void addFirst(E obj){
    Element e = new Element(obj);

    if (size == 0){
        first = last = e;
    }else{
        e.next = first;
        first.previous = e;
        first = e;
    }
    size++;
}

public void addLast(E obj){
    addLastElement(new Element (obj));
}

private void addLastElement(Element e){
    if (size == 0){
        first = last = e;
    }
}

```

```

        }else{
            e.previous = last;
            last.next = e;
            last = e;
        }
        size++;
    }

    public void add(int index, E obj){
        if (index < 0 || index > size){
            System.out.println("Index out of bounds");
            System.exit(-1);
        }
        Element e = new Element(obj);
        if (index < size){
            Element after = getElement(index);
            e.next = after;
            e.previous = after.previous;
            if (after.previous == null){
                first = e;
            }else{
                after.previous.next = e;
            }
            after.previous = e;
            size++;
        }else{
            addLastElement(e);
        }
    }

    public void clear(){
        if (size > 0){
            first = null;
            last = null;
            size = 0;
        }
    }

    public int size(){
        return size;
    }

    public boolean contains(Object o){
        Element e = first;
        while (e != null){
            if (o.equals(e.data)){
                return true;
            }
        }
    }

```

```

        }
        e = e.next;
    }
    return false;
}
}

```

C. Membuat dan Menjalankan Main Method ke-1

1. Silakan buat sebuah main method class dengan nama **TestLinkedList**
2. Kemudian salin tempel kode program di bawah ini

```

package linkedlist;

public class TestLinkedList {
    public static void main(String[] args) {
        LinkedList<String> cars = new LinkedList<>();

        cars.add("Volvo");
        cars.add("BMW");
        cars.add("Ford");
        cars.add("Mazda");

        cars.addFirst("Toyota");
        cars.addLast("Mitsubishi");

        System.out.println("Size of Linked List: "+cars.size());
        System.out.println("First element: "+cars.getFirst());
        System.out.println("Last element: "+cars.getLast());
        System.out.println("Element at index 4 -> "+cars.get(4));

        cars.add(6, "Wuling");
        System.out.println("\nAdd Wuling at index 6");
        System.out.println("Wuling in this LinkedList? "+cars.contains("Wuling"));

        System.out.println("Remove first -> "+cars.removeFirst());

        System.out.println("Remove last -> "+cars.removeLast());

        System.out.println("Remove at index 3 -> "+cars.remove(3));

        System.out.println("\nSize of Linked List: "+cars.size());

        System.out.println("\nAdd last -> Nissan");
        cars.addLast("Nissan");
        System.out.println("Add first -> Ferrari");
        cars.addFirst("Ferrari");
        System.out.println("\nSize of Linked List: "+cars.size());
        System.out.println("First element: "+cars.getFirst());
        System.out.println("Last element: "+cars.getLast());
    }
}

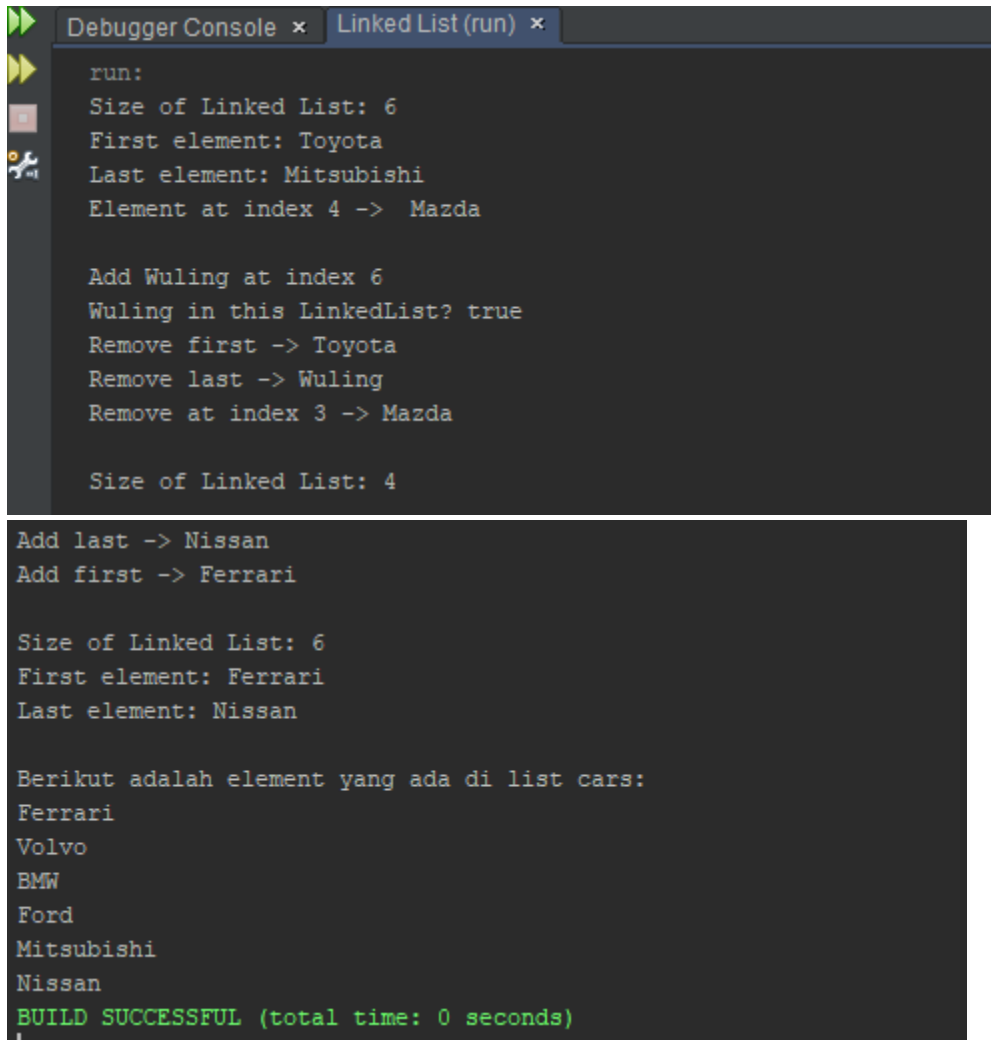
```

```

        /*
        Replace this code to display all elements of LinkedList
        */
    }
}

```

3. Jalankan *main method* **TestLinkedList** dan hasil tangkapan layar keluaran dari program silakan letakkan di bawah ini



```

Debugger Console x  LinkedList(run) x
run:
Size of Linked List: 6
First element: Toyota
Last element: Mitsubishi
Element at index 4 -> Mazda

Add Wuling at index 6
Wuling in this LinkedList? true
Remove first -> Toyota
Remove last -> Wuling
Remove at index 3 -> Mazda

Size of Linked List: 4

Add last -> Nissan
Add first -> Ferrari

Size of Linked List: 6
First element: Ferrari
Last element: Nissan

Berikut adalah element yang ada di list cars:
Ferrari
Volvo
BMW
Ford
Mitsubishi
Nissan
BUILD SUCCESSFUL (total time: 0 seconds)

```

D. Membuat Class Mahasiswa

1. Anda akan disuruh membuat class dengan nama **Mahasiswa**
2. Class ini nanti akan menjadi tipe data dari LinkedList yang akan dibuat (pada praktik poin C menggunakan tipe data String)
3. Salin tempel kode program di bawah ini di class **Mahasiswa** yang Anda buat

```

package linkedlist;

public class Mahasiswa {
    String nama = null;
    String NIM = null;
}

```

```

Double IPK = 0.0;

public Mahasiswa(String NIM, String nama, Double ipk){
    this.NIM = NIM;
    this.nama = nama;
    this.IPK = ipk;
}

public String getNama(){
    return this.nama;
}

public String getNIM(){
    return this.NIM;
}

public Double getIPK(){
    return this.IPK;
}

/*
Ganti kode ini untuk menambahkan method setNama(), setNIM() dan setIPK()
*/
}

```

4. Class **Mahasiswa** mempunyai 3 atribut yaitu nama, NIM dan IPK

5. *Method* yang dimiliki yaitu

- getNama() – untuk mengambil data nama dari objek mahasiswa
- getNIM() - untuk mengambil data NIM dari objek mahasiswa
- getIPK() - untuk mengambil data IPK dari objek mahasiswa

E. Membuat dan Menjalankan Main Method ke-2

1. Buatlah *main method* dengan nama **Main** salin tempel kode program di bawah ini

```

package linkedlist;

public class Main {
    public static void main(String[] args) {

        LinkedList<Mahasiswa> mhs = new LinkedList<>();

        mhs.add(new Mahasiswa("16523001", "Idris", 3.88));
        mhs.add(new Mahasiswa("16523002", "Agus", 3.18));
        mhs.add(new Mahasiswa("16523003", "Dani", 3.42));
        mhs.add(new Mahasiswa("16523008", "Roni", 3.67));
        mhs.add(new Mahasiswa("16523020", "Sinta", 3.05));

        LinkedList.Element ptr = mhs.first;

        while(ptr != null) {

```



```

        Mahasiswa m = (Mahasiswa) ptr.data;
        System.out.println("NIM: "+m.getNIM());
        System.out.println("Nama: "+m.getNama());
        System.out.println("IPK: "+m.getIPK());
        System.out.println("");
        ptr = ptr.next;
    }

    /*
    Ganti kode ini untuk mengganti IPK mahasiswa di indeks ke-3
    Kemudian tampilkan data mahasiswa yang ada di indeks ke-3
    */
}
}

```

2. Jalankan *main class* **Main** dan taruh hasil tangkapan layar dari keluarannya di bawah ini

```

Debugger Console x Linked List (run) x
NIM: 16523020
Nama: Sinta
IPK: 3.05

NIM: 16523001
Nama: Idris
IPK: 3.88

NIM: 16523002
Nama: Agus
IPK: 3.18

NIM: 16523003
Nama: Dani
IPK: 3.42

Berikut adalah hasil dari mengganti IPK pada index ke-3:
NIM: 16523008
Nama: Roni
IPK: 3.9

NIM: 16523020
Nama: Sinta
IPK: 3.05

BUILD SUCCESSFUL (total time: 0 seconds)

```

3. Pada class **Mahasiswa** silakan tambahkan beberapa *method* di bawah ini dan taruh kode programnya di kotak di bawah ini juga
setNama() – untuk mengeset/mengganti nama mahasiswa

```
public void setName(String namaBaru){  
    nama = namaBaru;  
}
```

setNIM() – untuk mengeset/mengganti NIM mahasiswa

```
public void setNIM (String nimBaru){  
    NIM = nimBaru;  
}
```

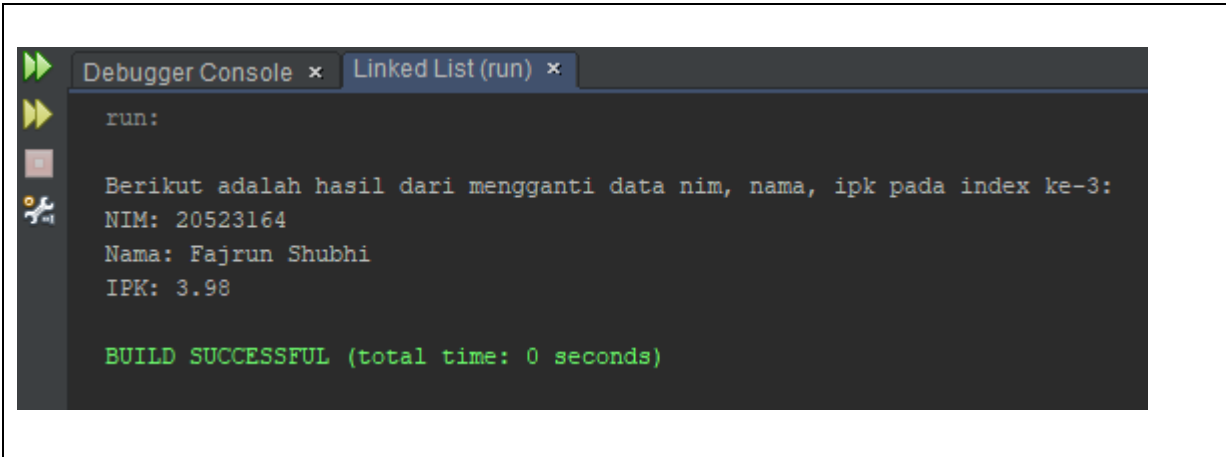
setIPK() – untuk mengeset/mengganti IPK mahasiswa

```
public void setIPK (Double ipkBaru){  
    IPK = ipkBaru;  
}
```

4. Pada bagian akhir main class Main silakan tambahkan beberapa kode untuk mengganti nama, NIM dan IPK mahasiswa yang berada di indeks ke-3 (menggunakan method **setName()**, **setNIM()** dan **setIPK()** yang sudah Anda buat) dan menampilkan data mahasiswa tersebut saja. Tulis kode programnya di bawah ini juga.

```
LinkedList.Element ubah = mhs.first;  
for (int i=0; i<mhs.size(); i++){  
    Mahasiswa mh = (Mahasiswa) ubah.data;  
    if (i == 3){  
        System.out.println("Berikut adalah hasil dari mengganti data nim, nama, ipk pada  
index ke-3: ");  
        mh.setIPK(3.98);  
        mh.setNIM("20523164");  
        mh.setName("Fajrun Shubhi");  
        System.out.println("NIM: "+mh.getNIM());  
        System.out.println("Nama: "+mh.getName());  
        System.out.println("IPK: "+mh.getIPK());  
        System.out.println("");  
        ubah = ubah.next;  
    }  
}
```

5. Jalankan kembali *main class* **Main** dan taruh hasil tangkapan layar dari keluarannya di bawah ini



```
run:

Berikut adalah hasil dari mengganti data nim, nama, ipk pada index ke-3:
NIM: 20523164
Nama: Fajrun Shubhi
IPK: 3.98

BUILD SUCCESSFUL (total time: 0 seconds)
```

***Catatan**

- Jangan lupa simpan juga file worksheet ini (yang sudah diisi) sebagai file pdf di folder NIM anda.
- Sertakan juga file **TestLinkedList.java**, **Mahasiswa.java** dan **Main.java** di dalam folder yang Anda gunakan
- Kompres folder ini sebagai file ZIP kemudian kumpulkan di classroom atau ruang pengumpulan lain di kelas masing-masing.