

VERSI 2.2
SEPTEMBER, 2020



[STRUKTUR DATA]

MODUL 2, ARRAY LIST & LINKED LIST

TIM PENYUSUN: - DOSEN
- DICKY PRABOWO OCTIANTO

PRESENTED BY: LAB. TEKNIK INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

[STRUKTUR DATA]

CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu menguasai & menjelaskan konsep dari struktur data *array list & linked list*

SUB CAPAIAN PEMBELAJARAN MATA KULIAH

Mahasiswa mampu memahami:

1. Collection tanpa menggunakan generics
2. Collection menggunakan generics
3. ArrayList
4. LinkedList

PERSYARATAN PEMAHAMAN

1. Class dan Object
2. Enkapsulasi
3. ArrayList
4. LinkedList

KEBUTUHAN HARDWARE & SOFTWARE

- Java Development Kit
- Java Runtime Environment
- IDE (Intellij IDEA, Eclipse, Netbeans, dll.)

REFERENSI MATERI

Oracle iLearning Java Programming section 6-2 & 6-3

MATERI POKOK

Collection

Collection adalah suatu obyek yang bisa digunakan untuk menyimpan sekumpulan obyek. Kelas Collection merupakan kelas generic, sehingga dapat digunakan untuk menampung objek yang memiliki tipe data berbeda.

Collection tanpa menggunakan generics

```
public class CellCollection {
    Cell[] cells;
    int index;
    public void add(Cell c) {
        cells[index]=c;
        index++;
    }
    public Cell get(int i) {
        return cells[i];
    }
    //more methods...
}
```

Kita juga memerlukan konstruktor untuk menginisialisasi ukuran array

```
Public CellCollection(int size) {
    cells = new Cell[size];
}
```

Collection menggunakan generics

```
public class CellGenericCollection<T> {
    T[] cells;
    int index;
    public void add(T c) {
        cells[index]=c;
        index++;
    }
    public T get(int i) {
        return cells[i];
    }
}
```

ArrayList

ArrayList adalah sebuah class untuk array yang resizable, class ini mengimplementasikan Interface List. ArrayList hanya mendukung tipe data object dan tidak mendukung tipe data primitif. Index dalam ArrayList dimulai dari 0 dan di akhiri dengan n-1 .

LinkedList

LinkedList dapat diilustrasikan seperti kereta api, dimana kereta api terdiri dari gerbong-gerbong yang saling terhubung yang dapat mengangkut penumpang (Data). Gerbong (*Node/Simpul*) disini berfungsi untuk menyimpan data. LinkedList merupakan struktur data yang setiap data nya memiliki pointer ke data berikutnya dan data sebelumnya.

Terdapat 3 jenis LinkedList : Single LinkedList , Double LinkedList , Circular LinkedList

Single LinkedList

Setiap node pada LinkedList mempunyai field yang berisi pointer ke node berikutnya dan juga memiliki field yang berisi data. Akhir LinkedList ditandai dengan node terakhir akan menunjuk ke null yang akan digunakan sebagai kondisi berhenti saat pembacaan LinkedList

Double LinkedList

LinkedList dimana setiap node memiliki 3 field yaitu 1 field pointer yang menunjuk ke pointer berikutnya (next) , 1 field pointer yang menunjuk ke pointer sebelumnya (prev) dan field yang berisi data dari Node tersebut. Pointer next dan prev menunjuk ke null.

Circular LinkedList**a. Single Circular LinkedList**

LinkedList yang pointer next-nya menunjuk ke diri nya sendiri, jika terdiri dari beberapa node maka pointer terakhir nya akan menunjuk ke pointer terdepan nya.

b. Double Circular Linkedlist

Double LinkedList yang pointer next nya dan prev nya menunjuk ke diri nya sendiri secara circular.

MATERI PRAKTIKUM

Biasakan untuk mengetik secara manual tidak menggunakan copy paste sehingga mengerti alur proses dari kode.

ArrayList

Beberapa method yang ada untuk ArrayList

No.	Class Methods	Description
1	<code>boolean add(E e)</code>	Appends the specified element to the end of this list.
2	<code>void add(int index, E element)</code>	Inserts the specified element at the specified position
3	<code>boolean addAll(Collection<? extends E> c)</code>	Appends all of the elements in the specified collection to the end of this list, in the order that they
4	<code>boolean addAll(int index, Collection<? extends E> c)</code>	Inserts all of the elements in the specified collection into this list, starting at the specified position.
5	<code>void clear()</code>	Removes all of the elements from this list.
6	<code>Object clone()</code>	Returns a shallow copy of this ArrayList instance.
7	<code>boolean contains(Object o)</code>	Returns true if this list contains the specified element.
8	<code>void ensureCapacity(int minCapacity)</code>	Increases the capacity of this ArrayList instance, if necessary, to ensure that it can hold at least the number of elements specified by the minimum capacity argument.
9	<code>E get(int index)</code>	Returns the element at the specified position in this list.
10	<code>int indexOf(Object o)</code>	Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
11	<code>boolean isEmpty()</code>	Returns true if this list contains no elements.

12	<code>int lastIndexOf(Object o)</code>	Returns the index of the last occurrence of the specified element in this list, or -1 if this list does not contain the element.
13	<code>E remove(int index)</code>	Removes the element at the specified position in this list.
14	<code>boolean remove(Object o)</code>	Removes the first occurrence of the specified element from this list, if it is present.
15	<code>protected void removeRange(int fromIndex, int toIndex)</code>	Removes from this list all of the elements whose index is between fromIndex, inclusive, and toIndex, exclusive.
16	<code>E set(int index, E element)</code>	Replaces the element at the specified position in this list with the specified element.
17	<code>int size()</code>	Returns the number of elements in this list.
18	<code>Object[] toArray()</code>	Returns an array containing all of the elements in this list in proper sequence (from first to last element).
19	<code><T> T[] toArray(T[] a)</code>	Returns an array containing all of the elements in this list in proper sequence (from first to last element); the runtime type of the returned array is that of the specified array.
20	<code>void trimToSize()</code>	Trims the capacity of this ArrayList instance to be the list's current size.

Menggunakan beberapa method yang ada dalam ArrayList

```
import java.util.ArrayList;

public class CthArrayList {

    public static void main(String args[]) {
        ArrayList<Integer> data = new ArrayList<>();
    }
}
```

```
data.add(1);
data.add(2);
data.add(3);
data.add(4);
data.add(5);

System.out.println(data.get(0));
System.out.println(data.get(1));
System.out.println(data.get(2));
System.out.println(data.get(3));
System.out.println(data.get(4));

}
}
```

LinkedList

Membuat LinkedList manual

1. Membuat kelas Node

```
public class Node {
    int data;
    Node next;
    Node prev;
    public Node (int data){
        this.data = data;
    }
}
```

2. Membuat kelas Link

```
public class Link {
    Node head;

    public void add(int data) {
        if (head == null) {
            head = new Node(data);
        }
    }
}
```

```

        Node current = head;
        while (current.next != null) {
            current = current.next;
        }
        current.next = new Node(data);
    }

    public void showData() {
        if (head == null) {
            System.out.println("LinkedList's Empty");
            return;
        }
        Node current = head;
        while (current.next != null) {
            current = current.next;
            int data = current.data;
            System.out.println(data);
        }
    }
}

```

3. Membuat kelas main LinkedList

```

public class LinkedList {

    public static void main(String[] args) {
        Link myLink = new Link();
        myLink.add(6);
        myLink.add(8);
        myLink.add(4);
        myLink.add(5);
        myLink.showData();
    }
}

```


LEMBAR KERJA**KEGIATAN 1**

Buat lah sebuah ArrayList of String (Diperkenankan menggunakan library) dengan nama-nama Mahasiswa, lalu isi Arraylist tersebut dengan ketentuan sebagai berikut

Index 0 = Lala

Index 1 = Shani

Index 2 = Gracia

Index 3 = Kyla

Index 4 = Gio

Index	0	1	2	3	4	5	6
Elemen	Lala	Shani	Gracia	Kyla	Gio		

1. Ubah isi Elemen pada index ke-3 menjadi Gina

Keadaan setelah index ke-3 diubah isinya

Index	0	1	2	3	4	5	6
Elemen	Lala	Shani	Gracia	Gina	Gio		

2. Masukkan Elemen baru dengan cara menyisipkan pada posisi (index) tertentu, dengan ketentuan sebagai berikut :

Index 1 = Daniel

Index 3 = Rusli

Keadaan setelah di masukan elemen pada posisi (index) tertentu

Index	0	1	2	3	4	5	6
Elemen	Lala	Daniel	Shani	Rusli	Gracia	Gina	Gio

3. Hapus Element diantara index ke-1 dan index ke-5

Keadaan setelah element antara index ke-1 dan index ke-5 dihapus

Index	0	1	2	3	4	5	6
Elemen	Lala	Daniel	Gina	Gio			

4. Tampilkan jumlah element pada arraylist

5. Cari ada ke posisi (index) berapa Lala dan Gina

Output:

```

Output - Modul2Strukdat (run)
run:
[Lala, Shani, Gracia, Kyla, Gio]
[Lala, Shani, Gracia, Gina, Gio]
[Lala, Daniel, Shani, Rusli, Gracia, Gina, Gio]
[Lala, Daniel, Gina, Gio]
Jumlah Element = 4
Index Lala = 0
Index Gina = 2
BUILD SUCCESSFUL (total time: 0 seconds)

```

KEGIATAN 2

Buat lah Program LinkedList secara manual, tidak diperkenankan menggunakan library dengan ketentuan sebagai berikut :

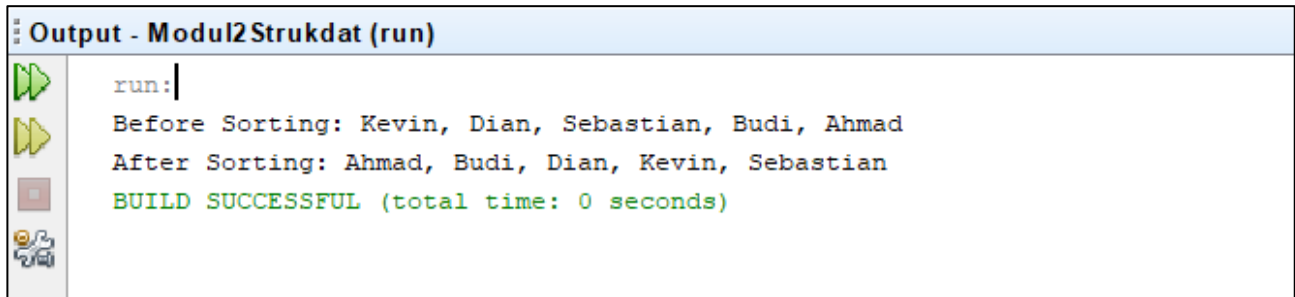
Inputan berupa nama dengan tipe data String. Ketika user menambah kan Node baru pada LinkedList , program akan memeriksa apakah nilai dari *Node baru* tersebut mempunyai nilai yang lebih besar atau lebih kecil daripada data yang lain secara alphabetical. Sehingga LinkedList akan mengurutkan node dari yang terkecil hingga terbesar berdasarkan urutan datanya.

Contoh :

```

LinkedList list = new LinkedList();
list.insert("Kevin");
list.insert("Dian");
list.insert("Sebastian");
list.insert("Budi");
list.insert("Ahmad");
list.display();

```

Output:

```
run:
Before Sorting: Kevin, Dian, Sebastian, Budi, Ahmad
After Sorting: Ahmad, Budi, Dian, Kevin, Sebastian
BUILD SUCCESSFUL (total time: 0 seconds)
```

CATATAN

Kegiatan 1 : Gunakan Library untuk mengerjakan

Kegiatan 2 : Buat Linkedlist secara manual tidak diperkenankan menggunakan library

Aturan umum penulisan bahasa JAVA agar mudah di koreksi oleh asisten:

1. Untuk nama kelas,interface,enum, dan yang lainnya biasakan menggunakan gaya CamelCase (diawali dengan huruf besar pada tiap kata untuk mengganti spasi) seperti: Kursi , JalanRaya, ParkiranGedung, dan lain seterusnya.
2. Untuk penulisan nama method, dan attribute diawali dengan huruf kecil di awal kata dan menggunakan huruf besar untuk kata setelahnya, seperti: getNamaJalan, namaJalan, harga, setNamaJalan, dan lain seterusnya.
3. Jika menggunakan IDE IntelliJ jangan lupa untuk memformat penulisan kode agar terlihat rapi menggunakan menu code -> show reformat file dialog -> centang semua field dan klik ok.

Silahkan dikerjakan tanpa copy – paste

RUBRIK PENILAIAN

Soal	Nilai
Kegiatan 1:	50%
Kegiatan 2:	50%