

VERSION 1.0
FEBRUARI 10, 2024



[STRUKTUR DATA]

MODUL 2, ARRAY LIST & LINKED LIST

DISUSUN OLEH:
IZZA IHSAN FATHONY
MOCH IQBAL ARIZKI WIDYANSYAKH

DIAUDIT OLEH:
MUHAMMAD ILHAM PERDANA. S.TR.T., M.T.

LAB. INFORMATIKA
UNIVERSITAS MUHAMMADIYAH MALANG

[STRUKTUR DATA]

PERSIAPAN MATERI

Mahasiswa diharapkan mempelajari materi sebelum mengerjakan tugas, materi yang tercakup antara lain:

1. Bahasa Pemrograman Java
2. Class dan Object
3. Enkapsulasi
4. Array

TUJUAN

Mahasiswa mampu menguasai dan menjelaskan konsep dari Struktur ArrayList & LinkedList.

TARGET MODUL

Mahasiswa mampu memahami:

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

PERSIAPAN SOFTWARE/APLIKASI

1. Java Development Kit
2. Java Runtime Environment
3. IDE (Intelij IDEA, Eclipse, Netbeans, dll)

REFERENSI MATERI

Youtube:

<https://youtu.be/GTjuHxcBeJg?si=BFkonGmRxDdVnVc0>

https://youtu.be/yav_GO-sVbU?si=obLZpQ2G2Fq3j4rP

<https://youtu.be/1nRj4ALuw7A?si=g4P-nMn099XMQeqJ>

Artikel:

<https://www.codepolitan.com/blog/menggunakan-arraylist-di-java/>

<https://www.petanikode.com/java-array/>

<https://www.javatpoint.com/java-arraylist>

<https://www.geeksforgeeks.org/arraylist-in-java/>

https://www.w3schools.com/java/java_arraylist.asp

Note: Dari referensi tersebut mungkin terdapat sedikit perbedaan satu sama yang lain, cukup pahami konsepnya dan terapkan pada kasus di modul ini.

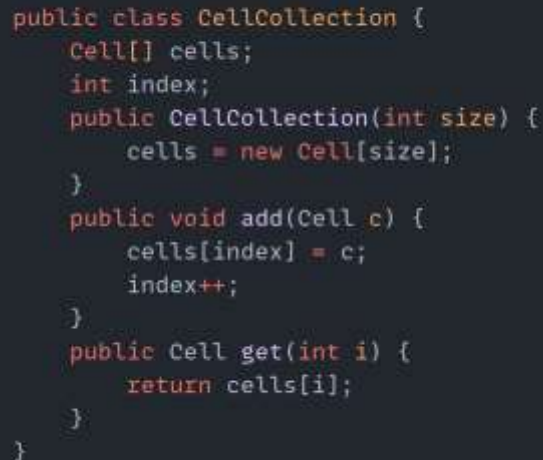
MATERI POKOK

1. 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.

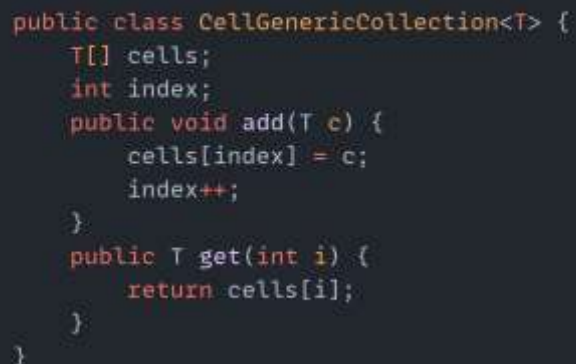
a. Collection tanpa menggunakan generics

Konstruktor diperlukan untuk menginisialisasi ukuran array.



```
public class CellCollection {
    Cell[] cells;
    int index;
    public CellCollection(int size) {
        cells = new Cell[size];
    }
    public void add(Cell c) {
        cells[index] = c;
        index++;
    }
    public Cell get(int i) {
        return cells[i];
    }
}
```

b. 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];
    }
}
```

2. 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 diakhiri dengan n-1.

a. Interface List

List digunakan untuk menyimpan sekumpulan objek berdasarkan urutan masuk (ordered) dan menerima duplikat. Cara penyimpanannya seperti array, oleh sebab itu memiliki posisi awal dan posisi akhir, menyisipkan objek pada posisi tertentu, mengakses dan menghapus isi list, Dimana semua proses ini selalu didasarkan pada urutannya. Class-class yang mengimplementasikan interface List adalah Vector, Stack, LinkedList dan ArrayList.

b. Method Untuk ArrayList

Beberapa method yang ada untuk penggunaan ArrayList.

i. add()

Digunakan untuk menambahkan elemen ke ArrayList.

```
ArrayList<String> daftarNama = new ArrayList<>();  
daftarNama.add("Izza");  
daftarNama.add("Iqbal");
```

ii. get()

Digunakan untuk mendapatkan elemen pada indeks tertentu dari ArrayList.

```
String namaPertama = daftarNama.get(0);
```

iii. set()

Digunakan untuk mengganti elemen pada indeks tertentu dengan elemen yang baru.

```
daftarNama.set(1, "Pak Ilham");
```

iv. remove()

Digunakan untuk menghapus elemen pada indeks tertentu dari ArrayList.

```
daftarNama.remove(0);
```

v. **clear()**

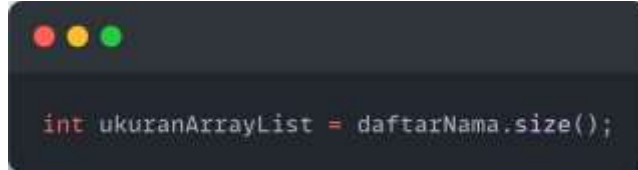
Digunakan untuk menghapus semua elemen dari ArrayList.



```
daftarNama.clear();
```

vi. **size()**

Digunakan untuk mendapatkan jumlah elemen dalam ArrayList.



```
int ukuranArrayList = daftarNama.size();
```

3. 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 datanya memiliki pointer ke data berikutnya dan data sebelumnya. Terdapat 3 jenis LinkedList yaitu: Single LinkedList, Double LinkedList, Circular LinkedList.

a. 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.

b. 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.

c. Circular LinkedList

Dalam Circular LinkedList ini masih dibagi lagi menjadi beberapa bagian diantaranya yaitu:

- **Single Circular LinkedList**

LinkedList yang pointer next-nya menunjuk ke dirinya sendiri, jika terdiri dari beberapa node maka pointer terakhirnya akan menunjuk ke pointer terdempunya.

- **Double Circular LinkedList**

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

4. Implementasi penggunaan beberapa method 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));
    }
}
```

5. Membuat LinkedList secara manual

a. Membuat Class Node

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

b. Membuat Class 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("Linklist is Empty");
            return;
        }
        Node current = head;
        while (current.next != null) {
            current = current.next;
            int data = current.data;
            System.out.println(data);
        }
    }
}
```

c. Membuat Class Main

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

CODELAB

LATIHAN 1

```
import java.util.ArrayList;

public class LatihanArrayList {

    public static void main(String[] args) {
        ArrayList<String> daftarNama = new ArrayList<>();

        daftarNama.add("Iqbal");
        daftarNama.add("Izza");
        daftarNama.add("Fathony");
        daftarNama.add("Arizki");

        System.out.println("Daftar Nama:");
        for (String nama : daftarNama) {
            System.out.println(nama);
        }

        System.out.println("Elemen pada indeks 2: " + daftarNama.get(2));

        daftarNama.set(1, "Paijo");

        System.out.println("Daftar Nama setelah perubahan:");
        for (String nama : daftarNama) {
            System.out.println(nama);
        }

        daftarNama.remove(0);

        System.out.println("Daftar Nama setelah penghapusan:");
        for (String nama : daftarNama) {
            System.out.println(nama);
        }

        System.out.println("Jumlah elemen dalam ArrayList: " + daftarNama.size());

        System.out.println("Apakah ArrayList kosong? " + daftarNama.isEmpty());
    }
}
```

1. Membuat objek **ArrayList** yang dapat menyimpan data bertipe String dengan nama **daftarNama**.
2. Menambahkan beberapa nama ke dalam **ArrayList** menggunakan method **add()**.
3. Menampilkan isi **ArrayList** dengan menggunakan perulangan for-each untuk mengiterasi melalui setiap elemen.
4. Mengakses elemen pada indeks ke-2 menggunakan method **get()**.
5. Mengubah elemen pada indeks ke -1 dengan nama **Paijo** menggunakan method **set()**.
6. Menghapus elemen pada indeks ke-0 menggunakan method **remove()**.
7. Menampilkan jumlah elemen dalam **ArrayList** menggunakan method **size()**.
8. Memeriksa apakah **ArrayList** kosong menggunakan method **isEmpty()**.

LATIHAN 2

```
public class Node {
    int data;
    Node next;

    public Node(int data) {
        this.data = data;
        this.next = null;
    }
}

public class LatihanLinkedListManual {
    public static void main(String[] args) {
        Node head = new Node(5);

        Node second = new Node(10);
        head.next = second;

        Node third = new Node(15);
        second.next = third;

        System.out.println("Daftar Angka:");
        Node current = head;
        while (current != null) {
            System.out.println(current.data);
            current = current.next;
        }

        System.out.println("Elemen kedua: " + head.next.data);

        head.data = 25;

        System.out.println("Daftar Angka setelah perubahan:");
        current = head;
        while (current != null) {
            System.out.println(current.data);
            current = current.next;
        }

        head.next = head.next.next;

        System.out.println("Daftar Angka setelah penghapusan:");
        current = head;
        while (current != null) {
            System.out.println(current.data);
            current = current.next;
        }
    }
}
```

1. Class Node

- a. Variabel **data** menyimpan nilai node, dan **next** adalah referensi ke node berikutnya dalam **LinkedList**.
- b. Membuat konstruktor **Node** dengan nilai **data** yang diberikan dan menginisialisasi **next** menjadi null.

2. Class LatihanLinkedListManual

- a. Membuat simpul pertama (head) dengan nilai 5.
- b. Menambahkan simpul kedua dengan nilai 10 dan menghubungkannya ke simpul pertama.
- c. Menambahkan simpul ketiga dengan nilai 15 dan menghubungkannya ke simpul kedua.
- d. Menggunakan perulangan untuk menampilkan isi **LinkedList** dari simpul pertama hingga simpul terakhir.
- e. Mengakses elemen kedua dengan mencapai simpul pertama dan mengikuti referensi **next** pada code **head.next.data**.
- f. Mengubah nilai elemen pertama menjadi 25,
- g. Menampilkan isi **LinkedList** setelah dilakukan perubahan pertama pada elemen pertama.
- h. Menghapus elemen kedua dengan mengubah referensi **next** pada elemen pertama sehingga langsung menghubungkannya ke elemen ketiga pada code **head.next = head.next.next**.
- i. Menampilkan isi **LinkedList** setelah elemen kedua dihapus.

TUGAS

TUGAS 1

Buatlah program sederhana untuk manajemen kontak menggunakan ArrayList. Program ini harus memungkinkan pengguna untuk menambahkan kontak baru, melihat daftar kontak, dan mencari kontak berdasarkan nama. Berikut adalah ketentuan tugas:

1. Class Contact memiliki 2 atribut.
2. Class ContactManager memiliki ArrayList untuk menyimpan daftar kontak. Harus memiliki method untuk menambahkan, menampilkan, dan mencari.
3. Program harus dapat berjalan secara berulang sehingga pengguna dapat terus mengeksekusi perintah hingga memilih untuk keluar.
4. Pastikan untuk menangani input pengguna dengan baik dan memberikan pesan yang jelas ketika kontak ditemukan atau tidak ditemukan.

Berikut adalah contoh output program:

```
Selamat datang di Manajemen Kontak!
1. Tambah Kontak
2. Tampilkan Kontak
3. Cari Kontak
4. Keluar
Pilih menu (1/2/3/4): 1
Masukkan nama kontak: Izza
Masukkan nomor telepon: 081234567890
Kontak berhasil ditambahkan!

Selamat datang di Manajemen Kontak!
1. Tambah Kontak
2. Tampilkan Kontak
3. Cari Kontak
4. Keluar
Pilih menu (1/2/3/4): 2
Daftar Kontak:
1. Izza - 081234567890

Selamat datang di Manajemen Kontak!
1. Tambah Kontak
2. Tampilkan Kontak
3. Cari Kontak
4. Keluar
Pilih menu (1/2/3/4): 3
Masukkan nama kontak yang ingin dicari: Izza
Kontak ditemukan!
Nama: Izza
Nomor Telepon: 081234567890
```

TUGAS 2

Ubahlah program sederhana untuk manajemen kontak pada tugas sebelumnya menggunakan LinkedList manual. Anda perlu menggantikan penggunaan ArrayList dengan LinkedList dan mengimplementasikan operasi-operasi dasar seperti menambah, menampilkan, dan mencari kontak secara manual menggunakan simpul-simpul. Pastikan untuk memodifikasi kode agar sesuai dengan struktur dan karakteristik dari LinkedList manual.

Mahasiswa diperbolehkan melakukan improvisasi dari tugas yang diberikan dengan syarat tidak mengurangi ketentuan yang ada.

Catatan:

1. Tugas 1 : Gunakan library untuk mengerjakan.
2. Tugas 2 : Buat LinkedList secara manual tidak diperkenankan menggunakan library

KRITERIA & DETAIL PENILAIAN

Kriteria	Nilai
Codelab 1	10
Codelab 2	10
Tugas 1	30
Tugas 2	30
Pemahaman	20
Total	100