

Nama : Arditya Adjie Rosandi

NIM : 20230801274

Pemrograman Berorientasi Objek



## Linked List

```
package Pertemuan13.Tugas;

import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class TugasLinkedList {
    Run | Debug
    public static void main(String[] args) {
        // Membuat dua objek list
        List<String> warna = new ArrayList<>();
        List<String> warnaDihapus = new LinkedList<>();

        // Mengisi list warna
        warna.add("MAGENTA");
        warna.add("RED");
        warna.add("WHITE");
        warna.add("BLUE");
        warna.add("CYAN");

        // Mengisi warnaDihapus dengan beberapa elemen yang sama
        warnaDihapus.add("RED");
        warnaDihapus.add("WHITE");

        // Hapus data dari warna yang terdapat pada warnaDihapus
        warna.removeAll(warnaDihapus);

        // Tampilkan hasil
        System.out.println("Warna: " + warna);
    }
}
```

Penggunaan koleksi ArrayList dan LinkedList di Java untuk mengelola data dalam bentuk list. Pada awalnya, dua list dibuat, yaitu warna menggunakan ArrayList untuk menyimpan data warna dan warnaDihapus menggunakan LinkedList untuk menyimpan daftar warna yang akan dihapus. List warna diisi dengan lima elemen warna: "MAGENTA", "RED", "WHITE", "BLUE", dan "CYAN", sedangkan list warnaDihapus diisi dengan warna "RED" dan "WHITE", yang juga terdapat dalam list warna. Selanjutnya, metode removeAll dipanggil pada list warna dengan argumen warnaDihapus. Metode ini secara otomatis menghapus elemen-elemen dalam warna yang juga terdapat pada warnaDihapus. Akibatnya, elemen "RED" dan "WHITE" dihapus dari list warna. Terakhir, elemen yang tersisa dalam list warna, yaitu "MAGENTA", "BLUE", dan "CYAN", ditampilkan melalui perintah System.out.println. Kode ini menunjukkan cara manipulasi data dalam list menggunakan operasi penghapusan berbasis elemen tertentu.

```

package Pertemuan13.Tugas;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.Random;

class Mahasiswa implements Comparable<Mahasiswa> {
    private String nrp;
    private String nama;
    private int nilai;

    public Mahasiswa(String nrp, String nama, int nilai) {
        this.nrp = nrp;
        this.nama = nama;
        this.nilai = nilai;
    }

    public int getNilai() {
        return nilai;
    }

    @Override
    public String toString() {
        return "NRP: " + nrp + ", Nama: " + nama + ", Nilai: " + nilai;
    }

    @Override
    public int compareTo(Mahasiswa m) {
        return Integer.compare(this.nilai, m.nilai);
    }
}

```

```

public class TugasPengurutanMahasiswa {
    Run | Debug
    public static void main(String[] args) {
        List<Mahasiswa> mahasiswaList = new ArrayList<>();
        Random rand = new Random();

        // Menambahkan 10 data mahasiswa
        mahasiswaList.add(new Mahasiswa(nrp:"202308001", nama:"Alice", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308002", nama:"Bob", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308003", nama:"Charlie", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308004", nama:"Diana", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308005", nama:"Eve", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308006", nama:"Frank", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308007", nama:"Grace", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308008", nama:"Hank", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308009", nama:"Ivy", rand.nextInt(41) + 60));
        mahasiswaList.add(new Mahasiswa(nrp:"202308010", nama:"Jack", rand.nextInt(41) + 60));

        // Tampilkan data sebelum diurutkan
        System.out.println("Data Mahasiswa sebelum diurutkan:");
        for (Mahasiswa m : mahasiswaList) {
            System.out.println(m);
        }

        // Urutkan data berdasarkan nilai
        Collections.sort(mahasiswaList);

        // Tampilkan data setelah diurutkan
        System.out.println("\nData Mahasiswa setelah diurutkan berdasarkan nilai:");
        for (Mahasiswa m : mahasiswaList) {
            System.out.println(m);
        }
    }
}

```

Program ini memanfaatkan kelas Mahasiswa, yang merepresentasikan informasi mahasiswa dengan atribut nrp (nomor registrasi mahasiswa), nama, dan nilai. Kelas ini mengimplementasikan antarmuka Comparable untuk memungkinkan pengurutan data berdasarkan atribut nilai. Metode compareTo pada kelas Mahasiswa membandingkan objek mahasiswa berdasarkan nilai mereka dengan menggunakan Integer.compare. Selain itu, metode toString diimplementasikan untuk menampilkan informasi mahasiswa dalam format yang mudah dibaca. Dalam kelas utama TugasPengurutanMahasiswa, sebuah list mahasiswaList

dideklarasikan menggunakan ArrayList dan diisi dengan sepuluh objek Mahasiswa. Setiap mahasiswa diberikan nilai acak antara 60 hingga 100 menggunakan kelas Random. Setelah itu, program menampilkan data mahasiswa sebelum diurutkan. Selanjutnya, metode Collections.sort digunakan untuk mengurutkan list berdasarkan nilai mahasiswa secara ascending, memanfaatkan implementasi compareTo. Akhirnya, data mahasiswa setelah diurutkan ditampilkan kembali. Program ini menunjukkan bagaimana cara menggunakan koleksi dan antarmuka Comparable untuk melakukan operasi pengurutan pada data.