MAKALAH, PERCOBAAN, LATIHAN, DAN TUGAS
MODUL PRAKTIKUM 10 BAGIAN 1

Disusun sebagai salah satu tugas
mata kuliah PBO I

Patricia Joanne
140810160065

Dikumpulkan tanggal
14 November 2017

PROGRAM STUDI S-1 TEKNIK INFORMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PADJADJARAN
2017

Tugas Pendahuluan

1. Buatlah resume 1 halaman mengenai Java Collection Framework dan pembagian kelompok Collection dan berikan contoh program penjelasannya.

Java Collection Framework adalah sekumpulan class dan interface yang membantu penyimpanan dan pemrosesan data secara efektif. Collection adalah suatu objek yang bisa digunakan untuk menyimpan sekumpulan objek. Objek yang ada dalam Collection disebut elemen. Collection menyimpan elemen yang bertipe Object, sehingga berbagai tipe objek bisa disimpan dalam Collection.

Class-class mengenai Collection tergabung dalam Java Collection Framework. Class-class Collection diletakkan dalam package java.util dan mempunyai dua interface utama yaitu Collection dan Map. Mulai J2SE 5, semua class yang termasuk Java Collection Framework adalah class Generics. Untuk kompatibilitas dengan versi Java sebelumnya, penggunaan Generics tidak diharuskan, namun sangat disarankan. Collection terbagi menjadi 3 kelompok yaitu:

1. **Set**

   Set mengikuti model himpunan, dimana objek/anggota yang tersimpan dalam Set harus unik. Urutan maupun letak dari anggota tidak penting, hanya keberadaan anggota saja yang penting. Class-class yang mengimplementasikan interface Set adalah HashSet. Interface SortedSet merupakan subInterface dari interface Set. Untuk mengurutkan Set, kita dapat menggunakan class yang mengimplementasikan interface SortedSet yaitu class TreeSet.

2. **List**

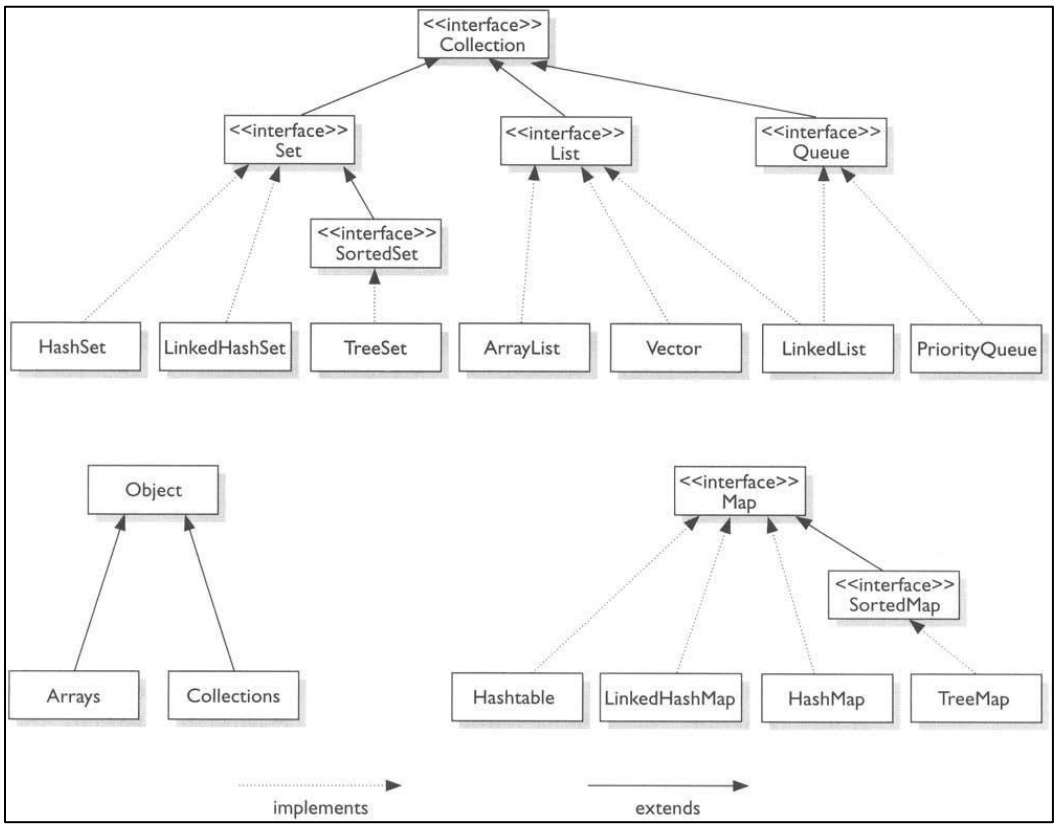
   List digunakan untuk menyimpan sekumpulan objek berdasarkan urutan masuk (ordered) dan menerima duplikat. Cara penyimpanannya seperti array, oleh sebab itu memiliki posisi awal dan posisi akhir, menyisipkan objek pada posisi tertentu, mengakses dan menghapus isi list, dimana semua proses ini selalu didasarkan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

Terdapat interface Queue yang cara penyimpanan seperti List, interface ini menyimpan objek menggunakan metode FIFO (First In First Out) yaitu objek yang masuk pertama keluar pertama. Class-class yang mengimplementasikan interface Queue adalah PriorityQueue dan LinkedList. Data yang tersimpan pada objek PriorityQueue akan diurutkan, data tersebut harus mengimplementasikan objek Comparable atau Comparator.

3. **Map**

   Perbedaan mendasar Map dengan Collection yang lain, untuk menyimpan objek pada Map, perlu sepasang objek, yaitu key yang bersifat unik dan nilai yang disimpan. Untuk mengakses nilai tersebut maka kita perlu mengetahui key dari nilai tersebut. Map juga dikenal sebagai dictionary/kamus. Pada saat menggunakan kamus, perlu suatu kata yang digunakan untuk pencarian. Class-class yang mengimplementasikan Map adalah Hashtable, HashMap, LinkedHashMap. Untuk mengurutkan Map menggunakan interface SortedMap, class yang mengimplementasikan interface tersebut adalah TreeMap.

Hierarki class dan interface Collection:



BAB 10 Bagian 1

Java Collection Framework: Set dan List

Percobaan 1: Memahami penggunaan class-class yang mengimplementasikan interface Set yaitu class HashSet dan class TreeSet

```
import java.util.*;

public class ContohSet {
    public static void main(String[] args){
        Set set = new HashSet();
        set.add("Asep");
        set.add("Gatot");
        set.add("Zainal");
        set.add("Asep");
        set.add("Dewi");
        System.out.print("Elemen pada HashSet : ");
        System.out.println(set);
        Set sortSet = new TreeSet(set);
        System.out.print("Elemen pada TreeSet : ");
        System.out.println(sortSet);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ContohSet.java
Note: ContohSet.java uses unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

ArrayList pada HashSet dan TreeSet karena di sini menggunakan unchecked dan unsafe operations sehingga harus kita ganti sesuai aturan Generics. Setelah diperbaiki:

```
import java.util.*;

public class ContohSet {
    public static void main(String[] args){
        Set<String> set = new HashSet<>();
        set.add("Asep");
        set.add("Gatot");
        set.add("Zainal");
        set.add("Asep");
        set.add("Dewi");
        System.out.print("Elemen pada HashSet : ");
        System.out.println(set);
        Set<String> sortSet = new TreeSet<>(set);
        System.out.print("Elemen pada TreeSet : ");
        System.out.println(sortSet);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ContohSet.java

E:\DOCS\task.bbr\Programming\Java>java ContohSet
Elemen pada HashSet : [Asep, Dewi, Gatot, Zainal]
Elemen pada TreeSet : [Asep, Dewi, Gatot, Zainal]
```

Percobaan 2: Penggunaan Class HashSet

```java
import java.util.*;

public class CariDuplikasi {
    public static void main(String[] args) {
        Set<String> s = new HashSet<String>();
        for(String a : args)
            if (!s.add(a))
                System.out.println("Terdeteksi duplikasi: "+a);
                System.out.println(s.size()+" kata berbeda: "+s);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac CariDuplikasi.java

E:\DOCS\task.bbr\Programming\Java>java CariDuplikasi
0 kata berbeda: []

E:\DOCS\task.bbr\Programming\Java>java CariDuplikasi saya datang saya lihat saya menang
Terdeteksi duplikasi: saya
Terdeteksi duplikasi: saya
4 kata berbeda: [datang, saya, menang, lihat]
```

```java
import java.util.*;

public class CariDuplikasi2 {
    public static void main(String[] args) {
        Set<String> unik = new HashSet<String>();
        Set<String> duplikat = new HashSet<String>();
        for (String a : args)
            if (!unik.add(a))
                duplikat.add(a);
            unik.removeAll(duplikat);
            System.out.println("Kata unik: " +unik);
            System.out.println("Kata duplikat: " +duplikat);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac CariDuplikasi2.java

E:\DOCS\task.bbr\Programming\Java>java CariDuplikasi2 saya datang saya lihat saya menang
Kata unik: [datang, menang, lihat]
Kata duplikat: [saya]
```

Percobaan 3: Interface Set menerapkan konsep himpunan. Mengetahui implementasi konsep himpunan pada interface Set.

```java
import java.util.*;

public class LatSet {
    public static void main(String[] args){
        Set<String> s1 = new HashSet<>();
        s1.add("Surabaya");
        s1.add("Jakarta");
        s1.add("Medan");
        s1.add("Bali");

        Set<String> s2 = new HashSet<>();
        s2.add("Jakarta");
        s2.add("Bandung");

        Set<String> union = new TreeSet<>(s1);
        union.addAll(s2); // gabungan dari s1 dan s2
        print("Gabungan",union);

        Set<String> intersect = new HashSet<>(s1);
        intersect.retainAll(s2); // irisan dari s1 dan s2
        print("Irisan",intersect);
    }
    protected static void print(String label, Collection c){
        System.out.println("----- "+label+" ------");
        Iterator it=c.iterator();
        while(it.hasNext()){
            System.out.println(it.next());
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac LatSet.java

E:\DOCS\task.bbr\Programming\Java>java LatSet
----- Gabungan ------
Bali
Bandung
Jakarta
Medan
Surabaya
----- Irisan ------
Jakarta
```

Percobaan 4: Memahami penggunaan class-class yang mengimplementasikan interface List yaitu ArrayList dan LinkedList

```java
import java.util.*;

public class ContohList {
    public static void main(String[] args){
        List<String> list = new ArrayList<>();
        list.add("Jakarta");
        list.add("Bekasi");
        list.add("Bandung");
        list.add("Aceh");
        list.add("Yogya");

        System.out.println(list);
        System.out.println("3 : "+list.get(3));
        System.out.println("1 : "+list.get(1));

        LinkedList<String> queue = new LinkedList<>();
        queue.addFirst("Jakarta");
        queue.addFirst("Bekasi");
        queue.addFirst("Bandung");
        queue.addFirst("Aceh");
        queue.addFirst("Yogya");
        System.out.println(queue);

        queue.removeLast();
        queue.removeLast();
        System.out.println(queue);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ContohList.java

E:\DOCS\task.bbr\Programming\Java>java ContohList
[Jakarta, Bekasi, Bandung, Aceh, Yogya]
3 : Aceh
1 : Bekasi
[Yogya, Aceh, Bandung, Bekasi, Jakarta]
[Yogya, Aceh, Bandung]
```

Percobaan 5: Penggunaan Class Vector

```java
import java.util.Vector;

public class ContohVector {
    public static void main(String[] args){
        Vector<String> vc = new Vector<String>();
        vc.add("Vector Object 1");
        vc.add("Vector Object 2");
        vc.add("Vector Object 3");
        vc.add("Vector Object 4");
        vc.add("Vector Object 5");
        vc.add(3, "Element at fix position");

        System.out.println("Vector Size : "+vc.size());
        for(int i=0;i<vc.size();i++){
        System.out.println("Vector Element "+i+": "+vc.get(i));
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac ContohVector.java

E:\DOCS\task.bbr\Programming\Java>java ContohVector
Vector Size : 6
Vector Element 0: Vector Object 1
Vector Element 1: Vector Object 2
Vector Element 2: Vector Object 3
Vector Element 3: Element at fix position
Vector Element 4: Vector Object 4
Vector Element 5: Vector Object 5
```

Percobaan 6: Penggunaan Iterator

```java
import java.util.*;

public class IteratorDemo {
    public static void main(String args[]){
        ArrayList<String> al = new ArrayList<>();
        al.add("C");
        al.add("A");
        al.add("E");
        al.add("B");
        al.add("D");
        al.add("F");
```

```
        System.out.print("Original contents of al: ");
        Iterator itr = al.iterator();
        while (itr.hasNext()){
            Object element = itr.next();
            System.out.print(element + " ");
        }

        System.out.println();
        ListIterator litr = al.listIterator();
        while (litr.hasNext()){
            Object element = litr.next();
            litr.set(element + "+");
        }

        System.out.print("Modified contents of al: ");
        itr = al.iterator();
        while (itr.hasNext()){
            Object element = itr.next();
            System.out.print(element + " ");
        }
        System.out.println();
```

```
        System.out.print("Modified list backwards: ");
        while (litr.hasPrevious()){
            Object element = litr.previous();
            System.out.print(element + " ");
        }
        System.out.println();
    }
}
```

```
Original contents of al: C A E B D F
Modified contents of al: C+ A+ E+ B+ D+ F+
Modified list backwards: F+ D+ B+ E+ A+ C+
```

Percobaan 7: Penggunaan Enumeration

```
import java.util.Vector;
import java.util.Enumeration;

public class EnumerationTester {
    public static void main(String args[]){
        Enumeration days;
        Vector<String> dayNames = new Vector<>();
```

```
            dayNames.add("Sunday");
            dayNames.add("Monday");
            dayNames.add("Tuesday");
            dayNames.add("Wednesday");
            dayNames.add("Thursday");
            dayNames.add("Friday");
            dayNames.add("Saturday");
            days = dayNames.elements();
            while (days.hasMoreElements ())
                System.out.println(days.nextElement());
        }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac EnumerationTester.java

E:\DOCS\task.bbr\Programming\Java>java EnumerationTester
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

Percobaan 8: Membuat Array List dari Enumerasi

```
import java.util.*;
import java.util.Vector;
import java.util.Enumeration;

public class ArrayListFromEnumeration {
    public static void main(String[] args) {
        Vector<String> v = new Vector<>();
        v.add("A");
        v.add("B");
        v.add("D");
        v.add("E");
        v.add("F");
        System.out.println("Vector contains: " +v);
        Enumeration e = v.elements();
        ArrayList aList = Collections.list(e);
        System.out.println("Arraylist contains: " +aList);
    }
}
```

,

```
Vector contains: [A, B, D, E, F]
Arraylist contains: [A, B, D, E, F]
```

Percobaan 9: Mengkopikan element dari ArrayList ke Vector

```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.Vector;

public class CopyElementsOfArrayListToVectorExample {
    public static void main(String[] args) {
        ArrayList<String> arrayList = new ArrayList<>();
        arrayList.add("1");
        arrayList.add("4");
        arrayList.add("2");
        arrayList.add("5");
        arrayList.add("3");

        Vector<String> v = new Vector<>();
        v.add("B");
        v.add("D");
        v.add("E");
        v.add("F");
        v.add("G");
        v.add("H");
        System.out.println("Before copy, Vector Contains : " +v);

        Collections.copy(v,arrayList);
        System.out.println("After Copy, Vector Contains : " +v);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac CopyElementsOfArrayListToVectorExample.java

E:\DOCS\task.bbr\Programming\Java>java CopyElementsOfArrayListToVectorExample
Before copy, Vector Contains : [B, D, E, F, G, H]
After Copy, Vector Contains : [1, 4, 2, 5, 3, H]
```

Percobaan 10: Menambahkan elemen yang tersimpan di Collection pada ArrayList

```java
import java.util.ArrayList;
import java.util.Vector;

public class AppendAllElementsOfOtherCollectionToArrayListExample {
    public static void main(String[] args) {
        ArrayList<String> arrayList = new ArrayList<>();
        arrayList.add("1");
        arrayList.add("2");
        arrayList.add("3");
```

```
        Vector<String> v = new Vector<>();
        v.add("4");
        v.add("5");
        arrayList.addAll(v);

        System.out.println("After appending all elements of Vector, ArrayList contains: ");
        for (int i=0;i<arrayList.size();i++){
            System.out.println(arrayList.get(i));
        }
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac AppendAllElementsOfOtherCollectionToArrayListExample.java

E:\DOCS\task.bbr\Programming\Java>java AppendAllElementsOfOtherCollectionToArrayListExample
After appending all elements of Vector, ArrayList contains:
1
2
3
4
5
```

Percobaan 11: Memahami Penggunaan dari class PriorityQueue

```
import java.util.*;

public class PriorityQueueDemo {
    static PriorityQueue<String> stringQueue;
    public static void main(String[] args){
        stringQueue = new PriorityQueue<String>();
        stringQueue.add("ab");
        stringQueue.add("abcd");
        stringQueue.add("abc");
        stringQueue.add("a");

        while(stringQueue.size() > 0)
            System.out.println(stringQueue.remove());
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac PriorityQueueDemo.java

E:\DOCS\task.bbr\Programming\Java>java PriorityQueueDemo
a
ab
abc
abcd
```

Percobaan 12: Memahami Penggunaan dari class PriorityQueue dan data yang tersimpan dalam objek PriorityQueue mengimplementasikan interface Comparator

```
import java.util.Comparator;
import java.util.PriorityQueue;

public class PQueueTest {
    public static void main(String[] args){
        PriorityQueue<Integer> pQueue = new PriorityQueue<Integer>(10, new Comparator<Integer>(){
            public int compare(Integer int1, Integer int2){
                boolean flag1 = isPrime(int1);
                boolean flag2 = isPrime(int2);
                if (flag1 == flag2){
                    return int1.compareTo(int2);
                }
                else if (flag1){
                    return -1;
                }
                else if(flag2){
                    return 1;
                }
                return 0;
            }
        });
```

```
        pQueue.add(1); pQueue.add(5); pQueue.add(6); pQueue.add(4); pQueue.add(2);
        pQueue.add(9); pQueue.add(7); pQueue.add(8); pQueue.add(10); pQueue.add(3);
        while(true){
            Integer head = pQueue.poll();
            if(head == null){
                break;
            }
            System.out.print(head+" <-- ");
        }
    }
    /**
    *
    * @param n
    * @return
    */
```

```
    public static boolean isPrime(int n){
        if (n <= 1){
            return false;
        }
        if (n == 2){
            return true;
        }
        if (n % 2 == 0){
            return false;
        }
```

```
        long m = (long) Math.sqrt(n);
        for (long i=3;i<=m;i+=2){
            if (n % i == 0){
                return false;
            }
        }
        return true;
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac PQueueTest.java

E:\DOCS\task.bbr\Programming\Java>java PQueueTest
2 <-- 3 <-- 5 <-- 7 <-- 1 <-- 4 <-- 6 <-- 8 <-- 9 <-- 10 <--
```

Latihan 1: Penerapan konsep himpunan pada interface Set

```
import java.util.*;

public class Lat1 {
    public static void main(String[] args){
        System.out.println("Patricia Joanne 140810160065");
        Set<Integer> a = new HashSet<Integer>();
        for (int i=1; i<=5; i++)
            a.add(i);
        Set<Integer> b = new HashSet<Integer>();
        for(int i=5; i<=10; i++)
            b.add(i);
        System.out.println("Set A = " + a);
        System.out.println("Set B = " + b);

        Set<Integer> min = new TreeSet<Integer>(a);
        min.removeAll(b);
        System.out.println("\nA - B = " + min);

        Set<Integer> irisan = new TreeSet<Integer>(a);
        irisan.retainAll(b);
        System.out.println("A n B = " +irisan);

        Set<Integer> gabung = new TreeSet<Integer>(a);
        gabung.addAll(b);
        System.out.println("A u B = " +gabung);

        System.out.println("A c B = " +b.containsAll(a));
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Lat1.java

E:\DOCS\task.bbr\Programming\Java>java Lat1
Patricia Joanne 140810160065
Set A = [1, 2, 3, 4, 5]
Set B = [5, 6, 7, 8, 9, 10]

A - B = [1, 2, 3, 4]
A n B = [5]
A u B = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
A c B = false
```

Latihan 2: Memahami penggunaan interface List

```java
import java.util.*;

public class Lat2 {
    public static void main(String[] args){
    System.out.println("Patricia Joanne 140810160065");
        List<String> l = new ArrayList<>();
        l.add("ayah");
        l.add("ibu");
        l.add("kakak");
        l.add("adik");

        System.out.print("Menampilkan isi List: ");
        for(Object object : l){
            String element = (String) object;
            System.out.print(element+" ");
        }

        Collections.reverse(l);
        System.out.print("\nMembalikkan isi List: " +l);
        Collections.shuffle(l);
        System.out.print("\nMengacak isi List: " +l);
        Collections.sort(l);
        System.out.print("\nMengurutkan isi List: " +l);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Lat2.java

E:\DOCS\task.bbr\Programming\Java>java Lat2
Patricia Joanne 140810160065
Menampilkan isi List: ayah ibu kakak adik
Membalikkan isi List: [adik, kakak, ibu, ayah]
Mengacak isi List: [adik, kakak, ayah, ibu]
Mengurutkan isi List: [adik, ayah, ibu, kakak]
```

Latihan 3: Memahami penggunaan interface List (2)

```java
import java.util.*;

class Mahasiswa implements Comparable<Mahasiswa>{
    protected String nrp;
    protected String nama;

    Mahasiswa(String nrp, String nama){
        this.nrp = nrp;
        this.nama = nama;
    }
    void setNrp(String nrp){
        this.nrp = nrp;
    }
    void setNama(String nama){
        this.nama = nama;
    }
    String getNrp(){
        return this.nrp;
    }
    String getNama(){
        return this.nama;
    }
    public String toString(){
        return "("+this.nrp+") "+this.nama;
    }
    public int compareTo(Mahasiswa mhs){
        return nrp.compareTo(mhs.getNrp());
    }
}
```

```java
public class Lat3 {
    public static void main(String[] args){
        System.out.println("Patricia Joanne 140810160065");
        List<Mahasiswa> l = new ArrayList<>();
        Mahasiswa mhs1 = new Mahasiswa("01", "Andi");
        l.add(mhs1);
        Mahasiswa mhs2 = new Mahasiswa("11", "Beni");
        l.add(mhs2);
        Mahasiswa mhs3 = new Mahasiswa("21", "Coki");
        l.add(mhs3);
        Mahasiswa mhs4 = new Mahasiswa("31", "Deni");
        l.add(mhs4);

        System.out.print("Menampilkan isi List: [ ");
        Iterator<Mahasiswa> it = l.iterator();
        while(it.hasNext()){
            Mahasiswa mhs = it.next();
            System.out.print("("+mhs.getNrp()+") "+mhs.getNama()+" ");
        }
        System.out.print("]");
```

```
        Collections.reverse(l);
        System.out.print("\nMembalikkan isi List: " +l);

        Collections.shuffle(l);
        System.out.print("\nMengacak isi List: " +l);

        Collections.sort(l);
        System.out.print("\nMengurut isi List: " +l);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Lat3.java

E:\DOCS\task.bbr\Programming\Java>java Lat3
Patricia Joanne 140810160065
Menampilkan isi List: [ (01) Andi (11) Beni (21) Coki (31) Deni ]
Membalikkan isi List: [(31) Deni, (21) Coki, (11) Beni, (01) Andi]
Mengacak isi List: [(21) Coki, (31) Deni, (01) Andi, (11) Beni]
Mengurut isi List: [(01) Andi, (11) Beni, (21) Coki, (31) Deni]
```

Tugas 1: Penggunaan class LinkedList pada interface List

```
import java.util.*;

public class Tgs1 {
    public static void main(String[] args){
        System.out.println("Patricia Joanne 140810160065");
        List<String> warna = new ArrayList<String>();
        List<String> warnaDihapus = new ArrayList<String>();
        warna.add("MAGENTA");
        warna.add("RED");
        warna.add("WHITE");
        warna.add("BLUE");
        warna.add("CYAN");
        warnaDihapus.add("RED");
        warnaDihapus.add("WHITE");
        warnaDihapus.add("BLUE");

        System.out.println("\nWarna:");
        System.out.println(warna);
        System.out.println("\nWarna yand dihapus:");
        System.out.println(warnaDihapus);
        warna.removeAll(warnaDihapus);
        System.out.println("\nOutput:");
        System.out.println(warna);
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Tgs1.java

E:\DOCS\task.bbr\Programming\Java>java Tgs1
Patricia Joanne 140810160065

Warna:
[MAGENTA, RED, WHITE, BLUE, CYAN]

Warna yand dihapus:
[RED, WHITE, BLUE]

Output:
[MAGENTA, CYAN]
```

Tugas 2: Pengurutan data mahasiswa berdasarkan nilai

```java
import java.util.*;

class Mahasiswa implements Comparable<Mahasiswa>{
    private String Nama;
    private String Nrp;
    private Float Nilai;

    public Mahasiswa(String Nama, String Nrp, Float Nilai){
        this.Nama = Nama;
        this.Nrp = Nrp;
        this.Nilai = Nilai;
    }
    public void setNama(String nama){
        Nama = nama;
    }
    public void setNrp(String nrp){
        Nrp = nrp;
    }
    public void setNilai(Float nilai){
        Nilai = nilai;
    }
    public String getNama(){
        return this.Nama;
    }
    public String getNrp(){
        return this.Nrp;
    }
    public Float getNilai(){
        return this.Nilai;
    }
```

```java
    @Override
    public int compareTo(Mahasiswa mhs){
        return Nilai.compareTo(mhs.getNilai());
    }
}

public class Tgs2 {
    public static void main(String[] args){
        System.out.println("Patricia Joanne 140810160065");
        List<Mahasiswa> listMhs = new ArrayList<Mahasiswa>();
        String[] nama = {"A","H","G","W","T","Q","B","K"};
        String[] nrp = {"062","001","010","057","029","045","017","032"};
        double nilai;

        for(int i=0;i<nama.length;i++){
            nilai = 60 + (Math.random()*(float)(101-60));
            listMhs.add(new Mahasiswa(nama[i], nrp[i], (float)nilai));
        }
        Collections.sort(listMhs);
        System.out.println(
            "----------------------\n" +
            "NRP \t Nama \t Nilai" +
            "\n----------------------"
        );
        for(Iterator<Mahasiswa> it = listMhs.iterator(); it.hasNext();){
            Mahasiswa mhs = it.next();
            System.out.print(mhs.getNama()+" \t "+mhs.getNrp()+" \t ");
            System.out.format("%.1f\n", mhs.getNilai());
        }
        System.out.println("----------------------");
    }
}
```

```
E:\DOCS\task.bbr\Programming\Java>javac Tgs2.java

E:\DOCS\task.bbr\Programming\Java>java Tgs2
Patricia Joanne 140810160065
----------------------
NRP      Nama    Nilai
----------------------
A        062     60.6
Q        045     62.0
W        057     63.5
B        017     68.0
H        001     68.5
T        029     72.8
K        032     85.3
G        010     99.7
----------------------
```