

Jobsheet 16

String dan Collection (List, Set, Map)

1. Tujuan Pembelajaran

- Memahami cara penyimpanan objek menggunakan Collection dan Map.
- Mengetahui pengelompokan dari Collection.
- Mengetahui perbedaan dari interface Set, List dan Map.
- Mengetahui penggunaan class-class dari interface Set, List, dan Map.

2. Materi

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**.

Collection terbagi menjadi 3 kelompok yaitu

a. Set

Set mengikuti model himpunan, dimana objek/anggota yang tersimpan dalam Set harus unik. Urutan maupun letak dari anggota tidaklah penting, hanya keberadaan anggota saja yang penting. Kelas konkrit yang mengimplementasikan **Set** harus memastikan bahwa tidak terdapat elemen duplikat yang dapat ditambahkan ke dalam set. Yaitu, tidak terdapat dua elemen **e1** dan **e2** yang berada di dalam set yang membuat **e1.equals(e2)** bernilai **true**. Class-class yang mengimplementasikan interface Set adalah HashSet.

HashSet dapat digunakan untuk menyimpan elemen-elemen bebas-duplikat. Kelas HashSet merupakan suatu kelas konkrit yang mengimplementasikan Set. Pembuatan objek HashSet adalah sebagai berikut:

```
Set <nama_objek_HashSet> = new HashSet();
```

b. 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. Pada jobsheet ini yang akan dibahas adalah ArrayList.

ArrayList digunakan untuk membuat array yang ukurannya dinamis. Berbeda dengan array biasa yang ukurannya harus ditentukan di awal deklarasi array, dengan ArrayList, ukurannya akan fleksibel tergantung banyaknya elemen yang dimasukkan. Pendeklarasian object ArrayList sebaiknya diikuti dengan nama class yang akan dimasukkan dalam List

tersebut. Tujuannya agar method dan property dari setiap object dalam ArrayList dapat diakses secara langsung. Pembuatan objek ArrayList adalah sebagai berikut:

ArrayList <nama_objek_arraylist>=new ArrayList();

c. 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. Pada jobsheet ini yang akan dibahas adalah HashMap.

HashMap adalah class implementasi dari Map, Map itu sendiri adalah interface yang memiliki fungsi untuk memetakan nilai dengan key unik. HashMap berfungsi sebagai *memory record management*, dimana setiap *record* dapat disimpan dalam sebuah Map. kemudian setiap Map diletakkan pada vektor, list atau set yang masih turunan dari collection. Pembuatan objek HashMap adalah sebagai berikut:

HashMap <nama_objek_HashMap> = new HashMap();

3. Percobaan 1 (Set)

A. Langkah Percobaan (HashSet)

1. Bukalah program Netbeans IDE yang sudah terinstal di komputer anda!
2. Buatlah project baru, kemudian buatlah package dengan nama identifier.percobaan1 , misalnya : vivin.percobaan1.
3. Buatlah class baru dengan nama TestHashSet

```
5  |  /**
6  |  package vivin.percobaan1;
7  |
8  |  /**
9  |  *
10 |  * @author WINDOWS 10
11 |  */
12 |  public class TestHashSet {
13 |      public static void main(String[] args) {
14 |
15 |      }
16 |  }
17 |
```

4. Tambahkan source code dibawah ini kedalam class yang sudah anda buat!

```

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

        Set set = new HashSet();

        set.add("Blitar");
        set.add("Jember");
        set.add("Jombang");
        set.add("Malang");
        set.add("Probolinggo");
        set.add("Malang");

        System.out.println(set);

        Iterator<String> iterator = set.iterator();

        while(iterator.hasNext()) {
            System.out.print(iterator.next().toUpperCase() + " ");
        }
    }
}

```

5. Jalankan program diatas, dan amati apa yang terjadi!
6. Tambahkan kode program berikut ini

```

6 package vivin.percobaan1;
7 import java.util.*;
8 /**
9  *
10  * Penulis: NURBONGGA 22

```

7. Kemudian jalankan program yang anda buat!

B. Pertanyaan

1. Apakah fungsi **import java.util.*;** pada program diatas!

.....

.....

.....

2. Pada baris program keberapakah yang berfungsi untuk menciptakan satu set Hash?

.....

.....

.....

3. Apakah fungsi potongan program dibawah ini pada percobaan 1 diatas!

```

set.add("Blitar");
set.add("Jember");
set.add("Jombang");
set.add("Malang");
set.add("Probolinggo");
set.add("Ponorogo");

```

.....

.....

.....

- Ubahlah `set.add("Ponorogo");` menjadi `set.add("Malang");` kemudian jalankan program! Amati hasilnya dan jelaskan mengapa outputnya demikian!

```
run:
[Blitar, Jember, Jombang, Malang, Probolinggo]
BLITAR JEMBER JOMBANG MALANG PROBOLINGGO BUILD SUCCESSFUL (total time: 0 seconds)
```

- Jelaskan fungsi potongan program dibawah ini pada percobaan 1 diatas!

```
Iterator<String> iterator = set.iterator();

while(iterator.hasNext()) {
    System.out.print(iterator.next().toUpperCase() + " ");
}
```

4. Percobaan 2 (List)

A. Langkah Percobaan (ArrayList)

- Buatlah package dengan nama `identifier.percobaan2` pada project yang sudah anda buat sebelumnya, misalnya : `vivin.percobaan2`.
- Buatlah class baru dengan nama `PercobaanList`

```
5  L  */
6      package vivin.percobaan2;
7
8  L  /**
9      *
10     * @author WINDOWS 10
11     */
12     public class PercobaanList {
13         public static void main(String[] args) {
14
15         }
16     }
```

- Tambahkan source code dibawah ini kedalam class yang sudah anda buat!

```
12     public class PercobaanList {
13         public static void main(String[] args) {
14             List list=new ArrayList();
15             list.add("Apel");
16             list.add("Strawberry");
17             list.add("Durian");
18             list.add("Jeruk");
19             list.add("Anggur");
20             System.out.println(list);
21             System.out.println("List ke-2 : "+list.get(2));
22             System.out.println("List ke-0 : "+list.get(0));
23
24         }
25     }
```

- Jalankan program diatas!

B. Pertanyaan

1. Jelaskan kenapa program diatas jika dijalankan terjadi error dan benarkan program tersebut sehingga dapat dijalankan menampilkan output seperti berikut:

```
Output - Josbheet16 (run)

run:
[Apel, Strawberry, Durian, Jeruk, Anggur]
List ke-2 : Durian
List ke-0 : Apel
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Apakah fungsi potongan program dibawah ini!

```
System.out.println(list);
System.out.println("List ke-2 : "+list.get(2));
System.out.println("List ke-0 : "+list.get(0));
```

3. Ganti potongan program pada soal no 2 menjadi sebagai berikut

```
System.out.println("Daftar Buah-Buahan: ");
Iterator itr = list.iterator();
while (itr.hasNext()) {
    Object element = itr.next();
    System.out.print(element + " ");
}
```

Kemudian jalankan program tersebut!

4. Jelaskan perbedaan menampilkan data pada ArrayList menggunakan potongan program pada soal no 2 dan no 3 diatas!

5. Percobaan 3 (Map)

A. Langkah Percobaan

1. Buatlah package dengan nama identifier.percobaan3 pada project yang sudah anda buat sebelumnya, misalnya : vivin.percobaan3.
2. Buatlah class baru dengan nama PercobaanHashMap

```

6      package vivin.percobaan3;
7
8      /**
9       *
10      * @author WINDOWS 10
11      */
12      public class PercobaanHashMap {
13          public static void main(String[] args) {
14
15          }
16      }

```

3. Tambahkan source code dibawah ini kedalam class yang sudah anda buat!

```

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

        HashMap hMap = new HashMap();

        hMap.put("1","Adi");
        hMap.put("2","Bagus");
        hMap.put("3","Sintia");
        System.out.println(hMap);
        System.out.println("Total HashMap adalah : " +hMap.size());

        System.out.println("=====");

        Object obj = hMap.remove("2");
        System.out.println(obj + " Dihapus dari HashMap");
        System.out.println("Total HashMap adalah : " +hMap.size());
        System.out.println(hMap);

        System.out.println("=====");

        hMap.clear();
        System.out.println("Total HashMap adalah : " +hMap.size());

    }
}

```

4. Jalankan program diatas!
5. Program diatas jika dijalankan akan terjadi error, maka tambahkan

```
import java.util.*;
```

Kemudian jalankan program maka outputnya adalah sebagai berikut:

Output - Josbheet16 (run)

```

run:
{1=Adi, 2=Bagus, 3=Sintia}
Total HashMap adalah : 3
=====
Bagus Dihapus dari HashMap
Total HashMap adalah : 2
{1=Adi, 3=Sintia}
=====
Total HashMap adalah : 0
BUILD SUCCESSFUL (total time: 0 seconds)

```

B. Pertanyaan

1. Jelaskan fungsi `hMap.put("1","Adi")` pada program diatas!

.....
.....
.....

2. Jelaskan fungsi `hMap.size()` pada program diatas!

.....
.....
.....

3. Jelaskan fungsi `hMap.remove("2")` pada program diatas!

.....
.....
.....

4. Jelaskan fungsi `hMap.clear()` pada program diatas!

.....
.....
.....

5. Tambahkan kode program dibawah ini pada progra yang sudah anda buat!

```
public class PercobaanHashMap {  
    public static void main(String[] args) {  
  
        HashMap hMap = new HashMap();  
  
        hMap.put("1", "Adi");  
        hMap.put("2", "Bagus");  
        hMap.put("3", "Sintia");  
  
        System.out.println(hMap);  
  
        Collection c = hMap.values();  
        Iterator itr = c.iterator();  
        while(itr.hasNext())  
            System.out.println(itr.next());  
  
        System.out.println("Total HashMap adalah : " +hMap.size());  
  
        System.out.println("=====");  
  
        Object obj = hMap.remove("2");  
        System.out.println(obj + " Dihapus dari HashMap");  
        System.out.println("Total HashMap adalah : " +hMap.size());  
        System.out.println(hMap);  
  
        System.out.println("=====");  
  
        hMap.clear();  
        System.out.println("Total HashMap adalah : " +hMap.size());  
    }  
}
```

6. Jalankan program dan amati apa yang terjadi!

7. Apakah perbedaan program sebelumnya dan setelah ditambahkan kode program pada soal no 5 diatas? Jelaskan!

.....

.....

.....

6. Percobaan 4 (Implementasi ArrayList dalam GUI)

1. Buatlah package dengan nama identifier.percobaan4 pada project yang sudah anda buat sebelumnya, misalnya : vivin.percobaan4.
2. Buatlah class baru dengan nama Mahasiswa

```
package vivin.percobaan4;

/**
 *
 * @author WINDOWS 10
 */
public class Mahasiswa {
```

3. Deklarasikan atribut Nim, Nama, Alamat dengan acces modifier private dan bertipe data String pada class Mahasiswa yang sudah dibuat

```
private String Nim;
private String Nama;
private String Alamat;
```

4. Buatlah kontruktor pada class Mahasiswa sebagai berikut:

```
public Mahasiswa (String Nim, String Nama, String Alamat){
    this.Nim=Nim;
    this.Nama=Nama;
    this.Alamat=Alamat;
}
```

5. Buatlah method getNim(),getNama(), dan getAlamat()

```
public String getNim(){
    return Nim;
}
public String getNama(){
    return Nama;
}
public String getAlmat(){
    return Alamat;
}
```

6. Buatlah class baru dengan nama InputData


```

L  : */
    package vivin.percobaan4;

    /**
     *
     * @author WINDOWS 10
     */
    public class InputData {

    }

```

7. Lakukan import ArrayList dengan menambahkan **import java.util.ArrayList;**

8. Buatlah ArrayList dengan nama ListMahasiswa dari class Mahasiswa

```
ArrayList<Mahasiswa> ListMahasiswa;
```

9. Buatlah kontruktur dari class InputData kemudian isi kontruktur tersebut dengan inisialisasi dari ArrayList **ListMahasiswa** yang sudah anda deklarasikan sebelumnya

```

public InputData () {
    ListMahasiswa= new ArrayList();
}

```

10. Buatlah method isi data sebagai berikut:

```

public void isiData (String Nim, String Nama, String Alamat){
    Mahasiswa mhs=new Mahasiswa (Nim, Nama, Alamat);
    ListMahasiswa.add(mhs);
}

```

11. Buatlah method getData() untuk mengambil seleuruh data yan ada pada ListMahasiswa

```

public ArrayList<Mahasiswa> getData(){
    return ListMahasiswa;
}

```

12. Buatlah class JFrame baru dengan nama TampilGui kemudian buat desain sebagai berikut:

Ketentuan:

Nama komponen	Text	Nama Variabel
jLabel	NIM	jLabel1
jLabel	Nama	jLabel2
jLabel	Alamat	jLabel3
jButton	Simpan	jButton1
jTextField	-	jTextFieldNim
jTextField	-	jTextFieldNama
jTextField	-	jTextAlamat
jTable	-	JTableMhs

13. Tambahkan `import javax.swing.table.DefaultTableModel;`

14. Deklarasikan DefaultTableModel sebagai TabBuku

```
DefaultTableModel TabMahasiswa;
```

15. Kemudian buatlah objek dari class InputData `InputData DataMahasiswa;`

16. Inisialisasi objek DataMahasiswa dari class Input Data didalam konstruktor TampilGui()

```
DataMahasiswa= new InputData();
```

17. Buatlah method LihatDataMahasiswa()

```

public final void LihatDataMahasiswa() {
    String[] NamaKolom={"NIM","Nama Mahasiswa", "Alamat"};
    Object[][] objekMahasiswa=new Object [DataMahasiswa.getData().size()][3];
    int i=0;
    for (Mahasiswa mhs: DataMahasiswa.getData()) {
        String arrayMahasiswa[]={mhs.getNim(),mhs.getNama(),mhs.getAlamat()};
        objekMahasiswa[i]=arrayMahasiswa;
        i++;
    }
    TabMahasiswa=new DefaultTableModel(objekMahasiswa, NamaKolom);
    jTableMhs.setModel(TabMahasiswa);
}

```

18. Panggil method LihatDataMahasiswa() yang sudah dibuat didalam konstruktor TampilGui()

```
LihatDataMahasiswa();
```

19. Double klik pada Button Simpan kemudian tambahkan kode berikut:

```

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    DataMahasiswa.isiData(jTextFieldNim.getText(), jTextFieldNama.getText(), jTextFieldAlamat.getText());
    LihatDataMahasiswa();
}

```

20. Jalankan program

21. Tambahkan data berikut kemudian klik simpan

A screenshot of a Java Swing application window. The window has a title bar with standard minimize, maximize, and close buttons. Inside the window, there are three text input fields stacked vertically. The first field is labeled 'NIM' and contains the text '156150100111006'. The second field is labeled 'Nama' and contains the text 'vivin ayu lestari'. The third field is labeled 'Alamat' and contains the text 'Jl. Vinolia No 1A'. Below these fields is a button labeled 'Simpan'. At the bottom of the window is a JTable with three columns: 'NIM', 'Nama Mahasiswa', and 'Alamat'. The table is currently empty of data rows.

NIM	Nama Mahasiswa	Alamat
-----	----------------	--------

22. Maka data akan tersimpan pada JTable