

Jobsheet Minggu ke-12

Collection (List, Set, Map) dan Database

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.
- Memahami koneksi database menggunakan JDBC dan JDBC API

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 object 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 tidak penting, hanya keberadaan anggota saja yang penting. Kelas konkret 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 konkret 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();

d. JDBC API

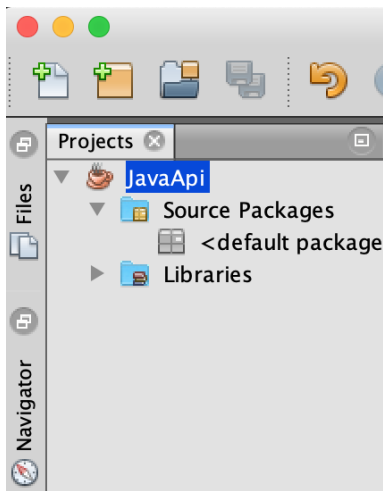
JDBC API merupakan Java Database Connectivity Application Programming Interface (JDBC API). Pada dasarnya JDBC API terdiri dari satu set kelas dan interface yang digunakan untuk berinteraksi dengan database dari aplikasi Java. Umumnya, JDBC API melakukan 3 (tiga) fungsi berikut :

- Membangun koneksi antara aplikasi Java dan database
- Membangun dan mengeksekusi query
- Memproses hasil

3. Percobaan 1 (Set)

A. Langkah Percobaan (HashSet)

1. Bukalah program Netbeans IDE yang sudah terinstal di komputer anda!
2. Buatlah project baru dengan nama JavaApi, ilustrasi project dapat dilihat pada gambar dibawah ini:



3. Setelah itu buatlah package dengan nama identifier.percobaan1, misalnya : arie.percobaan1.
4. Pada package tersebut buatlah java class dengan nama DemoHashSet.
5. Tambahkan source code yang ada di method main() kedalam class yang sudah anda buat!

```
package arie.percobaan1;

import java.util.HashSet;
import java.util.Iterator;
import java.util.Set;

public class CobaHashSet {
    public static void main(String[] args) {
        Set mSetCity = new HashSet();
        mSetCity.add("Malang");
        mSetCity.add("Banyuwangi");
        mSetCity.add("Jogjakarta");
        mSetCity.add("Batu");

        System.out.println(mSetCity);
        Iterator<String> mIterator = mSetCity.iterator();

        while (mIterator.hasNext()){
            System.out.println(mIterator.next().toLowerCase());
        }
    }
}
```

6. Jalankan program diatas, dan amati apa yang terjadi!

B. Pertanyaan

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

.....

.....

2. Pada baris program keberapakah yang berfungsi untuk menciptakan object HashSet?

.....

.....

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

```
mSetCity.add("Malang");
mSetCity.add("Banyuwangi");
mSetCity.add("Jogjakarta");
mSetCity.add("Batu");
```

4. Tambahkan **set.add("Malang")**; kemudian jalankan program! Amati hasilnya dan jelaskan apa yang terjadi!

5. Jelaskan fungsi potongan program dibawah ini pada percobaan 1!

```
Iterator<String> mIterator = mSetCity.iterator();

while (mIterator.hasNext()){
    System.out.println(mIterator.next().toLowerCase());
}
```

4. Percobaan 2 (List)

A. Langkah Percobaan (ArrayList)

1. Buatlah package dengan nama identifier.percobaan2 pada project yang sudah anda buat sebelumnya, misalnya : arie.percobaan2.
2. Tambahkan source code berikut pada class yang sudah anda buat!

```
package arie.percobaan2;

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

public class CobaArrayList {
    public static void main(String[] args) {
        List mListCountry = new ArrayList();
        mListCountry.add("Indonesia");
        mListCountry.add("Malaysia");
        mListCountry.add("German");
        System.out.println(mListCountry);
        System.out.format("List index 0 = %s\n", mListCountry.get(0));
        System.out.format("List index 2 = %s\n", mListCountry.get(2));
    }
}
```

3. Jalankan program diatas!

B. Pertanyaan

1. Apakah fungsi potongan program dibawah ini!

```
System.out.println(mListCountry);
System.out.format("List index 0 = %s\n", mListCountry.get(0));
System.out.format("List index 2 = %s\n", mListCountry.get(2));
```

2. Ganti potongan program pada soal no 1 menjadi sebagai berikut

```
System.out.println("Country List");
Iterator mIterator = mListCountry.iterator();
while(mIterator.hasNext()){
    Object element = mIterator.next();
    System.out.format("%s ",element);
}
```

Kemudian jalankan program tersebut!

3. Jelaskan perbedaan menampilkan data pada ArrayList menggunakan potongan program pada soal no 1 dan no 2!

5. Percobaan 3 (Map)

A. Langkah Percobaan

1. Buatlah package dengan nama identifier.percobaan3 pada project yang sudah anda buat sebelumnya, misalnya : arie.percobaan3.
2. Buatlah class baru dengan nama DemoHashMap. Tambahkan source code dibawah ini kedalam class!

```
package arie.percobaan3;

import java.util.HashMap;

public class DemoHashMap {
    public static void main(String[] args) {
        HashMap hMapItem = new HashMap();
        hMapItem.put("1","Biskuit");
        hMapItem.put("2","Shampoo");
        hMapItem.put("3","Soap");
        System.out.println(hMapItem);
        System.out.format("HashMap Item Tolal : %d\n",hMapItem.size());

        Object mObject = hMapItem.remove("1");
        System.out.format("%s remove from Hashmap\n",mObject);
        System.out.format("HashMap Item Tolal : %d\n",hMapItem.size());
        System.out.println(hMapItem);

        hMapItem.clear();
        System.out.format("HashMap Item Tolal : %d\n",hMapItem.size());
    }
}
```

3. Jalankan program diatas, maka outputnya adalah sebagai berikut:

```

run:
{1=Biskuit, 2=Shampoo, 3=Soap}
HashMap Item Tolal : 3

Biskuit remove from Hashmap
HashMap Item Tolal : 2
{2=Shampoo, 3=Soap}
HashMap Item Tolal : 0
BUILD SUCCESSFUL (total time: 0 seconds)

```

B. Pertanyaan

1. Jelaskan fungsi **hMapItem.put("1","Biskuit")** pada program!

.....

.....

.....

2. Jelaskan fungsi **hMapItem.size()** pada program!

.....

.....

.....

3. Jelaskan fungsi **hMapItem.remove("1")** pada program!

.....

.....

.....

4. Jelaskan fungsi **hMapItem.clear()** pada program!

.....

.....

.....

5. Tambahkan kode program yang di blok pada program yang sudah anda buat!

```

public class DemoHashMap {
    public static void main(String[] args) {
        HashMap hMapItem = new HashMap();
        hMapItem.put("1","Biskuit");
        hMapItem.put("2","Shampoo");
        hMapItem.put("3","Soap");
        System.out.println(hMapItem);

        Collection mCollection = hMapItem.values();
        Iterator mIterator = mCollection.iterator();
        while (mIterator.hasNext()) {
            System.out.println(mIterator.next());
        }

        System.out.format("HashMap Item Tolal : %d\n\n",hMapItem.size());

        Object mObject = hMapItem.remove("1");
        System.out.format("%s remove from Hashmap\n",mObject);
        System.out.format("HashMap Item Tolal : %d\n",hMapItem.size());
        System.out.println(hMapItem);

        hMapItem.clear();
        System.out.format("HashMap Item Tolal : %d\n",hMapItem.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 : arie.percobaan4.
2. Buatlah class baru dengan nama 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.Alatam=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
7. Lakukan import ArrayList dengan menambahkan **import java.util.ArrayList;**
8. Buatlah ArrayList dengan nama ListMahasiswa dari class Mahasiswa
`ArrayList<Mahasiswa> ListMahasiswa;`
9. Buatlah kontruktor dari class InputData kemudian isi kontruktor 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 |
| textField | - | textFieldNim |
| textField | - | textFieldNama |
| textField | - | textFieldAlamat |
| jTable | - | JTableMhs |

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

14. Deklarasikan DefaultTableModel sebagai TabMahasiswa

```
DefaultTableModel TabMahasiswa;
```

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

16. Inisialisasi objek DataMahasiswa dari class Input Data didalam kontruktor 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 kontruktor 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

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

21. Tambahkan data berikut kemudian klik simpan

NIM: 156150100111006

Nama: vivin ayu lestari

Alamat: Jl. Vinolia No 1A

Simpan

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

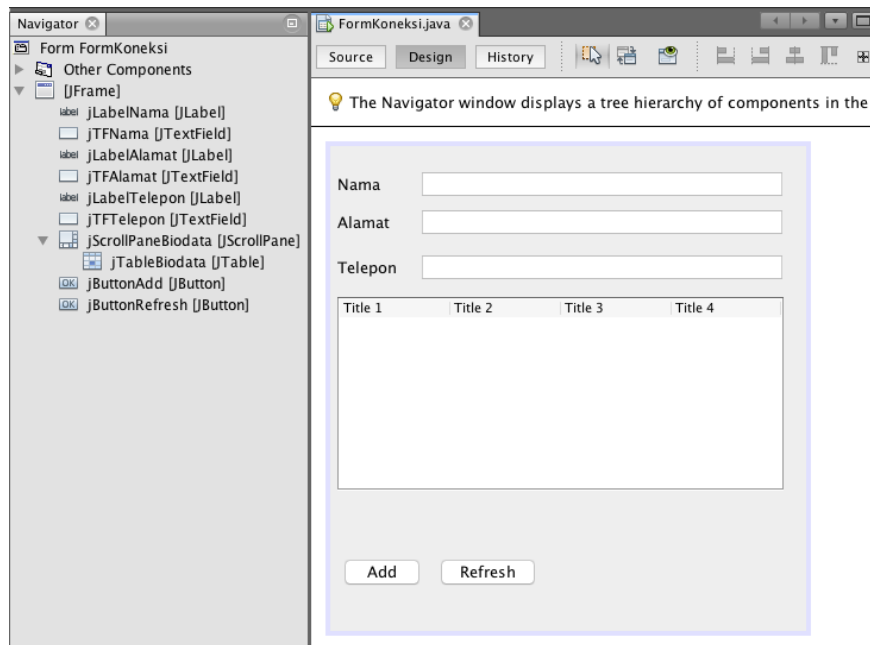
22. Maka data akan tersimpan pada Jtable

7 Percobaan 5 (Aplikasi Biodata)

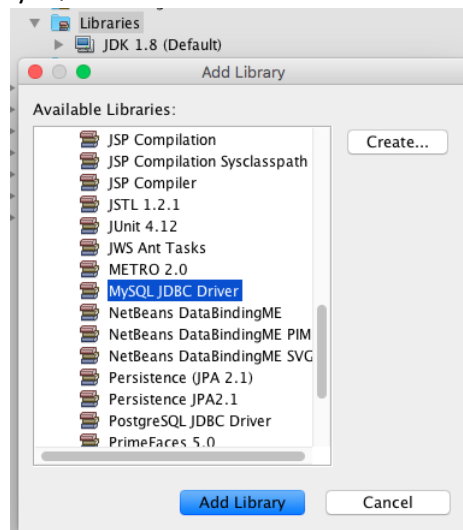
1. Buatlah database dengan MySQL dengan desain tabel sebagai berikut:

```
CREATE TABLE anggota (id INT NOT NULL AUTO_INCREMENT, nama
VARCHAR(50) NOT NULL, alamat VARCHAR(100) NOT NULL, telp
VARCHAR(30) NOT NULL, PRIMARY KEY (id ));
```

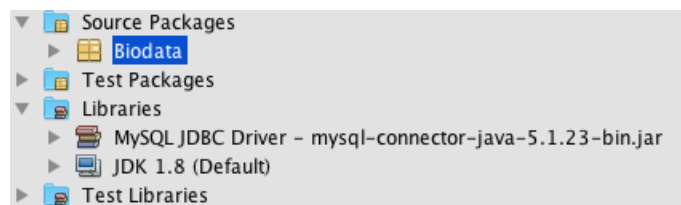
1. Buatlah Package baru dengan nama Biodata atau membuat project baru dengan cara
 - a. File → New Project
 - b. Pilih categories → Java dengan Projects → Java Application → klik next
 - c. Isi nama Project dengan Biodata dan atur letak project akan disimpan
 - d. Kemudian tambahkan JFrame Form dengan nama FormKoneksi.
 - e. Desain FormKoneksi seperti berikut



- Langkah selanjutnya tambahkan libraries, dengan cara klik kanan pada Libraries. Kemudian pilih add library dan pilih MySQL JDBC Driver.



Tampilan Libraries setelah ditambah MySQL JDBC Driver.



- Pada tahap ini aplikasi sudah siap untuk bertransaksi dengan database yang sudah kita buat pada langkah 1. Selanjutnya kita buat method untuk melakukan koneksi dengan database, seperti berikut:

```

public class FormKoneksi extends javax.swing.JFrame {
    private static Connection koneksi;
    public FormKoneksi() {
        initComponents();
    }
    private static void buka_koneksi() {
        if (koneksi == null) {
            try {
                String url = "jdbc:mysql://localhost:6606/Biodata";
                String user = "root";
                String password = "";
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                koneksi = DriverManager.getConnection(url, user, password);
            } catch (SQLException t) {
                System.out.println("Error membuat koneksi");
            }
        }
    }
}

```

Pada method buka_koneksi untuk variabel url terdapat angka 6606 adalah port mysql yang digunakan oleh penulis. Pada umumnya port mysql secara default adalah **3306**. Sedangkan untuk Biodata adalah nama database yang digunakan oleh penulis.

4. Pada button add klik tambahkan code sebagai berikut:

```

buka_koneksi();
String sqlkode="Insert into anggota (nama,alamat,telp) "
+ "values (""+this.jTFNama.getText()+"", "
+ """"+this.jTFAlamat.getText()+"", "
+ """"+this.jTFTelepon.getText()+"")";

try {
    PreparedStatement mStatement = koneksi.prepareStatement(sqlkode);
    mStatement.executeUpdate();
    mStatement.close();
    JOptionPane.showMessageDialog(this,"Data Berhasil Ditambah");
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(this,"Terjadi Kesalahan "+ex.getMessage());
}

```

5. Setelah menambah code pada action button klik, coba jalankan program dan tambahkan data. Apakah program berhasil menambahkan data? Jika tidak apakah penyebabnya.

.....

6. Jelaskan maksud source code untuk melakukan insert data diatas?

.....

7. Buat Table model yang digunakan untuk memanipulasi tampilan pada Jtable, seperti pada code dibawah ini:

```

public class FormKoneksi extends javax.swing.JFrame {

    private static Connection koneksi;
    private DefaultTableModel model;

    public FormKoneksi() {
        initComponents();
        model = new DefaultTableModel();
        this.jTableBiodata.setModel(model);
        model.addColumn("ID");
        model.addColumn("Nama");
        model.addColumn("Alamat");
        model.addColumn("Telepon");
        ambil_data_tabel();
    }

```

8. Selanjutnya buat method `ambil_data_tabel` seperti berikut

```

private void ambil_data_tabel() {
    model.getDataVector().removeAllElements();
    model.fireTableDataChanged();
    try {
        buka_koneksi();
        Statement s = koneksi.createStatement();
        String sql = "Select * from anggota";
        ResultSet r = s.executeQuery(sql);
        while (r.next()) {
            Object[] o = new Object[4];
            o[0] = r.getString("id");
            o[1] = r.getString("nama");
            o[2] = r.getString("alamat");
            o[3] = r.getString("telp");
            model.addRow(o);
        }
        r.close();
        s.close();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, "Terjadi kesalahan " + e.getMessage());
    }
}

```

Jalankan program maka data yang sudah kita add akan tampil seperti berikut:

| ID | Nama | Alamat | Telepon |
|----|------|----------------|-------------|
| 1 | Arie | Jalan Pulosaei | 08161577... |

9. Jelaskan alur dari method `ambil_data_tabel`?
.....
10. Buat fungsi untuk merefresh sehingga data yang baru dapat ditampilkan pada tabel.
2.