**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 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**. ClassClass 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

Perbedaaan 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 dar 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

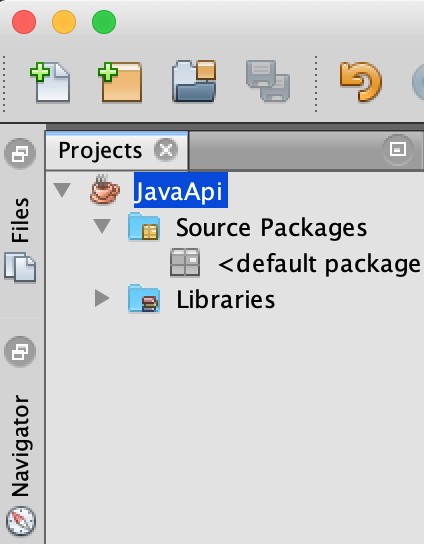
JDBC API merupakan Java Database Connectivity Application Programming Interface ( JDBC API). Pada dasarnya JDBC API terdiri dari satu set kelas dan interface yang digunakan untuk berinteraksi dengan database dari aplikasi Java. Umumnya, JDBC API melakukan 3 (tiga) fungsi berikut :

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

# 3. Percobaan 1 (Set)

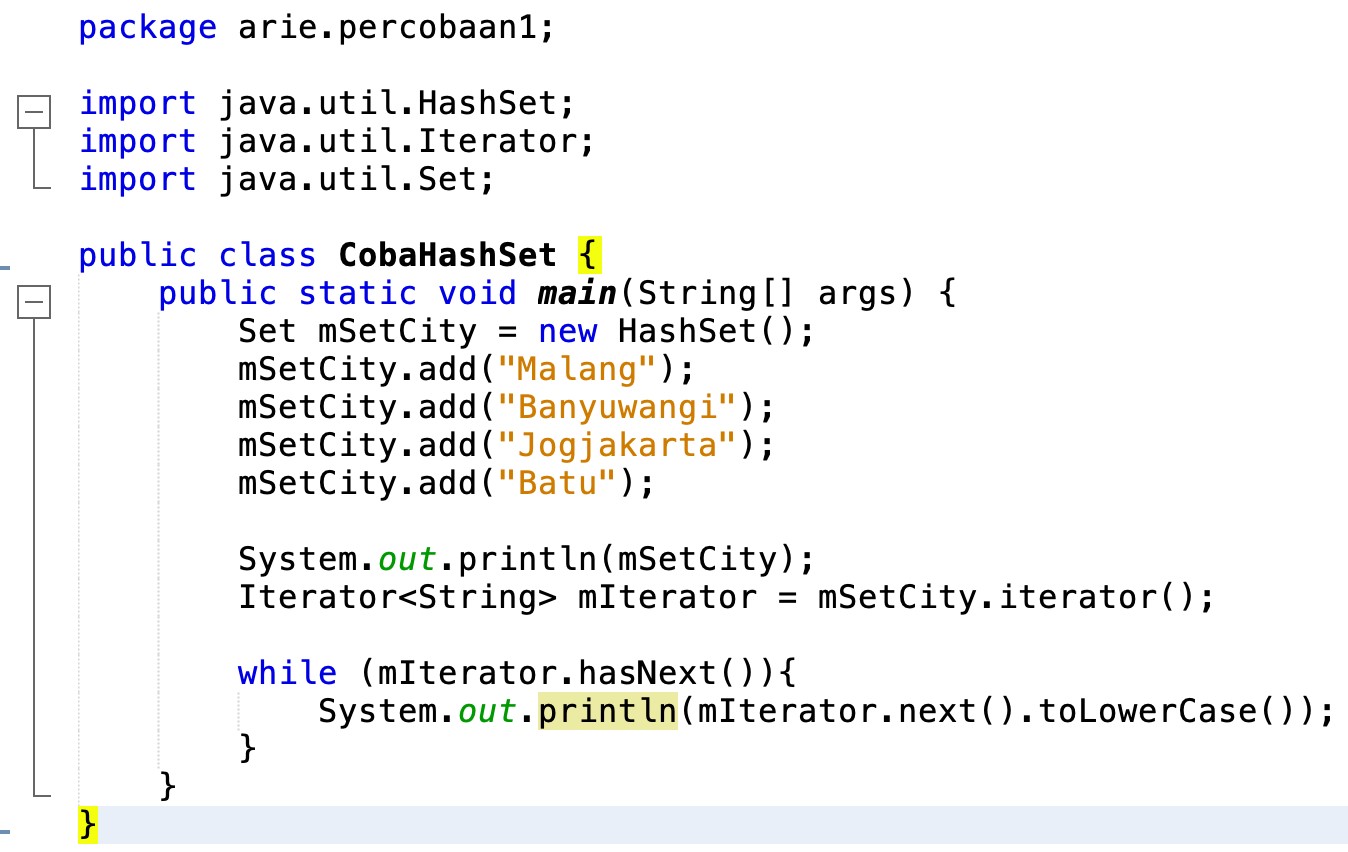
A. Langkah Percobaan (HashSet)

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



1. Setelah itu buatlah package dengan nama identifier.percobaan1, misalnya : arie.percobaan1.
2. Pada package tersebut buatlah java class dengan nama DemoHashSet.

1. Tambahkan source code yang ada di method main() kedalam class yang sudah anda buat!



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

## B. Pertanyaan

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

* Untuk menggunakan utiliti yang ada dalam library java, seperti scanner dll.

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

* Pada baris yang terdapat baris program berikut :

*Set* mSetCity = new HashSet();

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

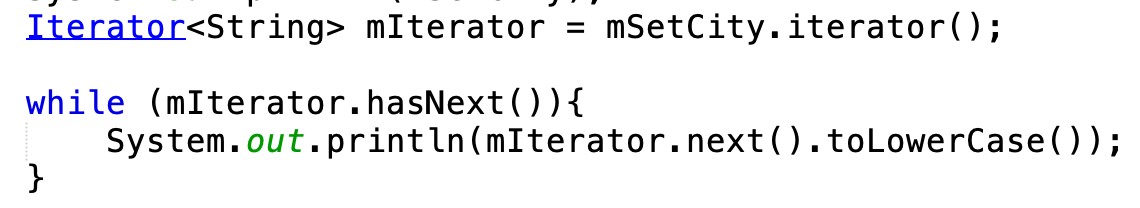


* Untuk menambahkan nama kota-kota tersebut kedalam objek mSetCity.

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

* Program tidak dapat dijalankan karena tidak ada objek dengan nama set dalam class.

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



* Pada baris program berikut :

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

Berfungsi untuk membuat objek bertipe arraylist dari class iterator. Sedangkan di baris program berikut :

while(mIterator.hasNext()) {

            System.**out**.println(mIterator.next().toLowerCase());

      }

Berfungsi untuk menampilkan nama-nama kota yang telah ditambahkan sebelumnya yang ada dalam objek mSetCity.

# 4. Percobaan 2 (List)

A. Langkah Percobaan (ArrayList)

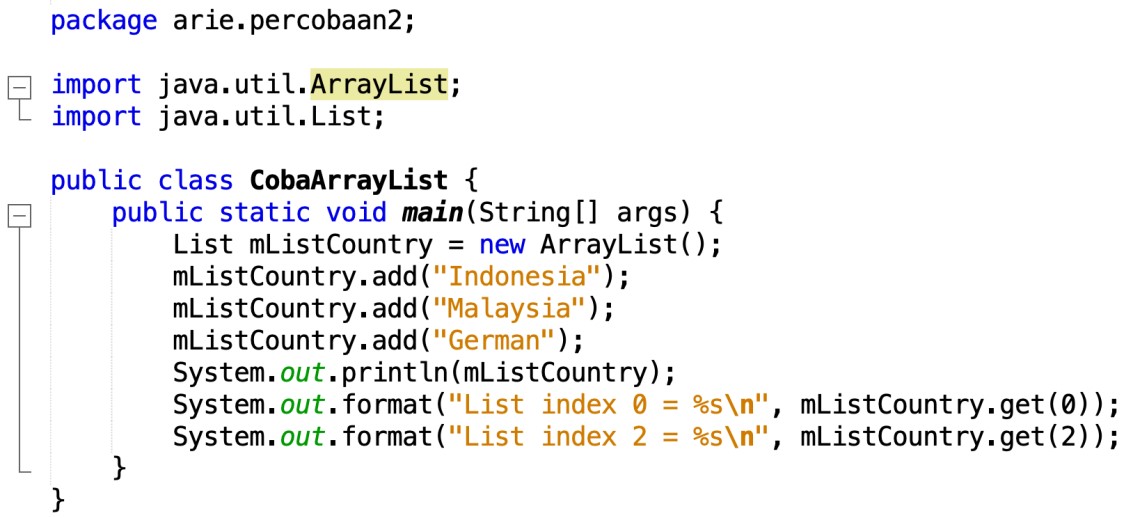
1. Buatlah package dengan nama identifier.percobaan2 pada project yang sudah anda buat sebelumnya, misalnya : arie.percobaan2.
2. Tambahkan source code berikut pada class yang sudah anda buat!

3.

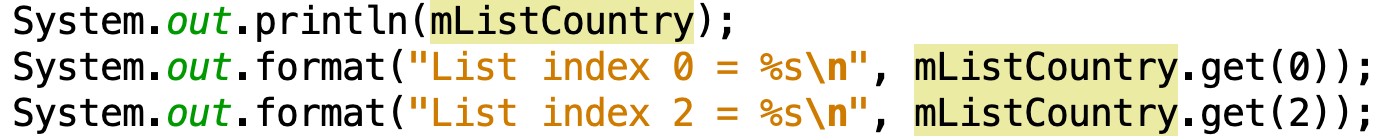
Jalankan program diatas!

B.

Pertanyaan



1. Apakah fungsi potongan program dibawah ini!



* Pada bagian baris program berikut :

System.**out**.println(mListCountry);

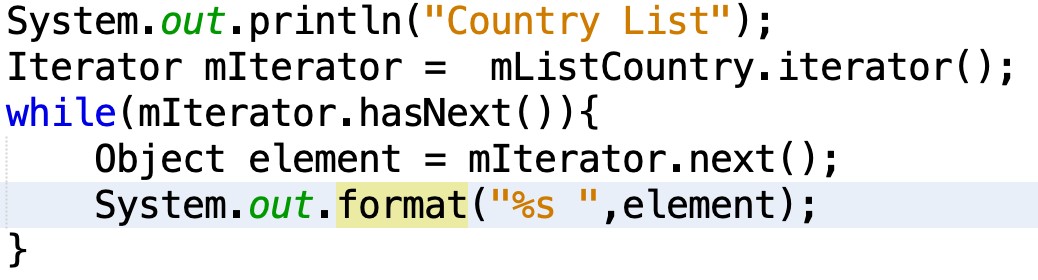
Befungsi untuk menampilkan isi arraylist dari objek mListCountry. Sedangkan baris program berikut :

System.**out**.format("List index 0 : %s\n", mListCountry.get(0));

System.**out**.format("List index 2 : %s\n", mListCountry.get(2));

Berfungsi untuk menampilkan isi arraylist dari mSetCountry pada index tertentu.

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



Kemudian jalankan program tersebut!

* Program dapat dijalankan dan akan menampilkan seluruh isi arraylist dari mSetCountry.



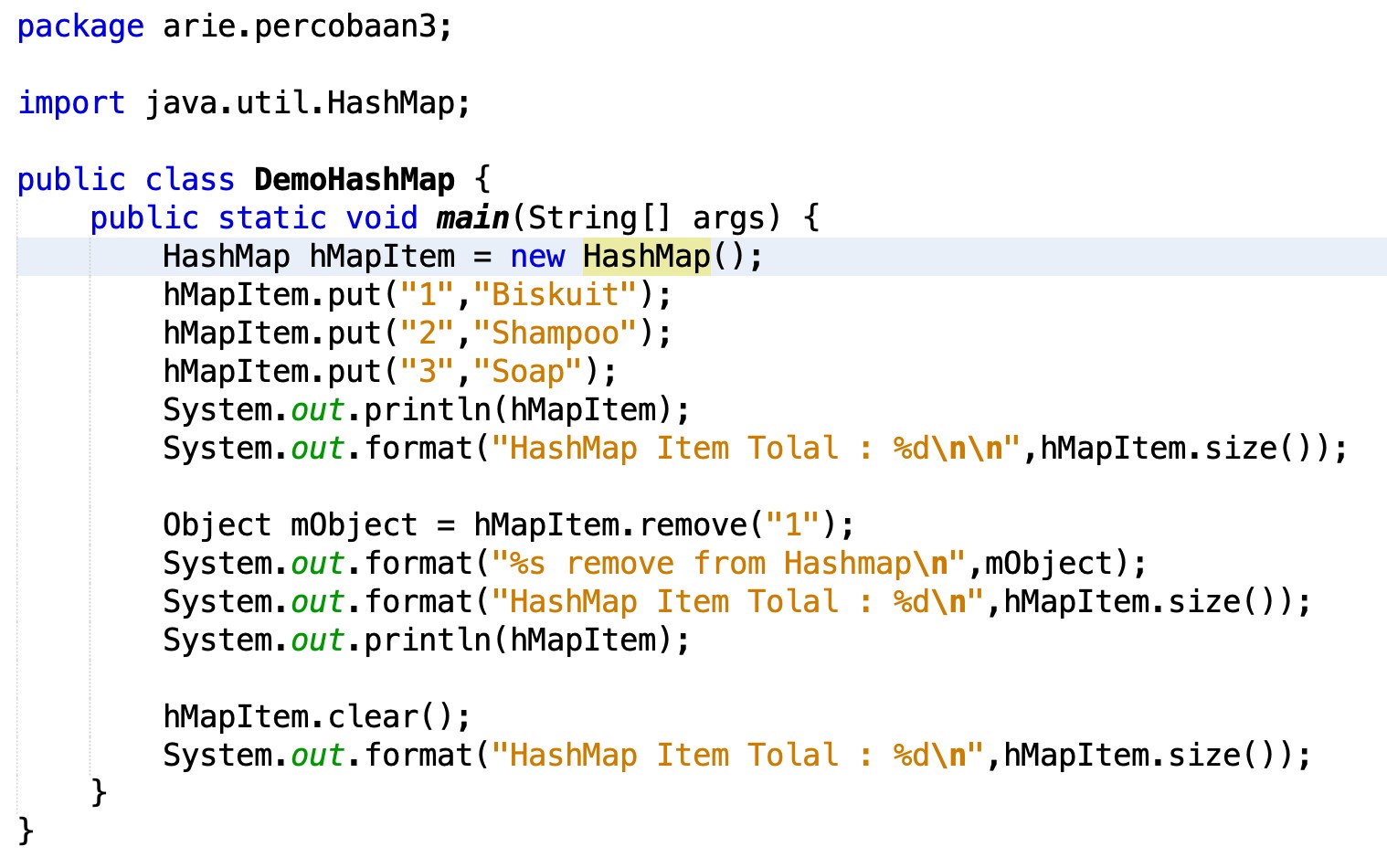
1. Jelaskan perbedaan menampilkan data pada ArrayList menggunakan potongan program pada soal no 1 dan no 2!

* Pada soal no 1 hanya menampilkan secara spesifik pada index ke berapa dalam isi mSetCountry, sedangkan pada no 2 menampilkan keseluruhan dari mSetCountry.

5. Percobaan 3 (Map)

# A. Langkah Percobaan

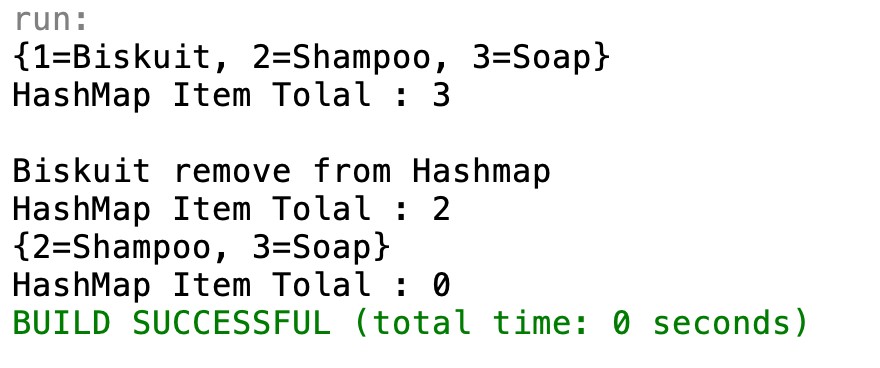
1. Buatlah package dengan nama identifier.percobaan3 pada project yang sudah anda buat sebelumnya, misalnya : arie.percobaan3.
2. Buatlah class baru dengan nama DemoHashMap. Tambahkan source code dibawah ini kedalam class!



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

B.

Pertanyaan



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

* Untuk menambahkan data tersebut kedalam objek hMapItem.

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

* Fungsi tersebut digunakan untuk menampilkan jumlah item yang ada dalam hMapItem.

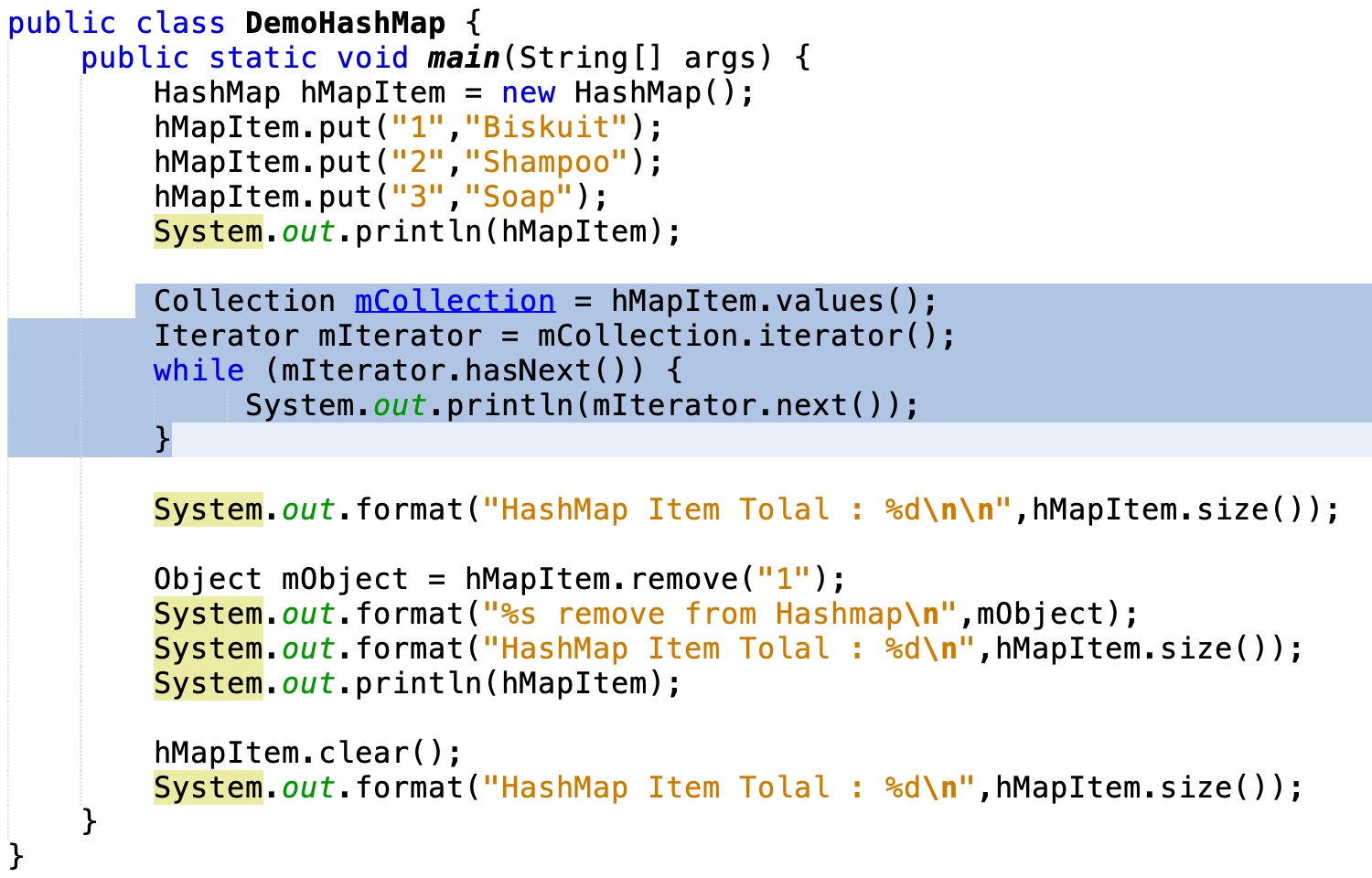
1. Jelaskan fungsi **hMapItem.remove("1")** pada program!

* Fungsi tersebut digunakan untuk menghapus item dengan data yang mengandung “1” didalamnya.

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

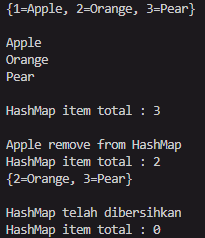
* Fungsi tersebut digunakan untuk menghapus semua data yang ada dalam hMapItem.

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

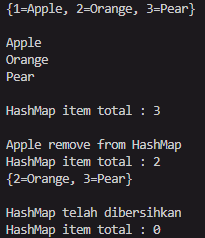


1. Jalankan program dan amati apa yang terjadi!

* Yang terjadi adalah item-item dalam hMapItem akan ditampilkan seluruhnya, namun tidak dengan angka yang mewakili item tersebut.



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

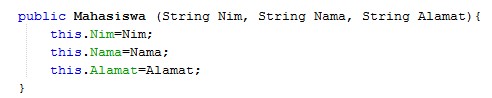
* Perbedaanya adalah penampilan semua item dalam hMapItem, namun untuk soal no 5 tidak menampilkan angka yang mewakili item tersebut. 

## 6 Percobaan 4 (Implementasi ArrayList dalam GUI)

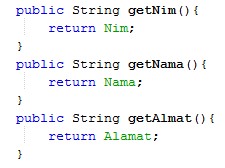
1. Buatlah package dengan nama identifier.percobaan4 pada project yang sudah anda buat sebelumnya, misalnya : arie.percobaan4.
2. Buatlah class baru dengan nama Mahasiswa
3. Deklarasikan atribut Nim,Nama, Alamat dengan acces modifier private dan bertipe data String pada class Mahasiswa yang sudah dibuat



1. Buatlah kontruktor pada class Mahasiswa sebagai berikut:



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



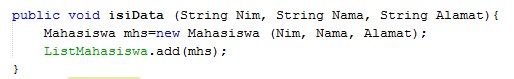
1. Buatlah class baru dengan nama InputData
2. Lakukan import ArrayList dengan menambahkan **import java.util.ArrayList;** 8. Buatlah ArrayList dengan nama ListMahasiswa dari class Mahasiswa



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



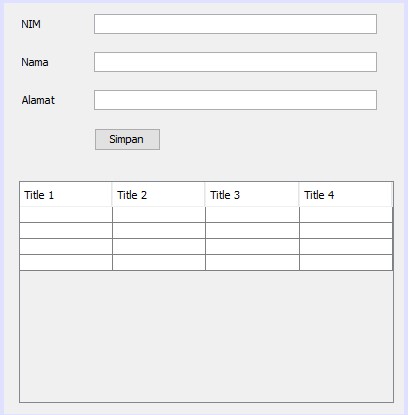
1. Buatlah method isi data sebagai berikut:



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



1. 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 |

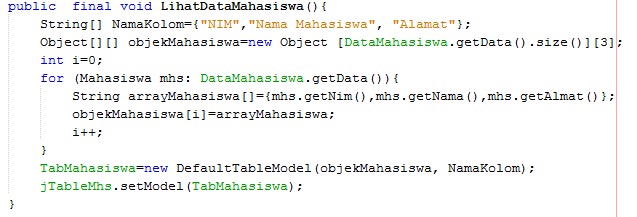
1. Tambahkan 
2. Deklarasikan DefaultTableModel sebagai TabMahasiswa
3. Kemudian buatlah objek dari class InputData



1. Inisialisasi objek DataMahasiswa dari class Input Data didalam kontruktor TampilGui()



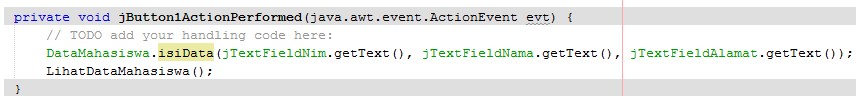
1. Buatlah method LihatDataMahasiswa()



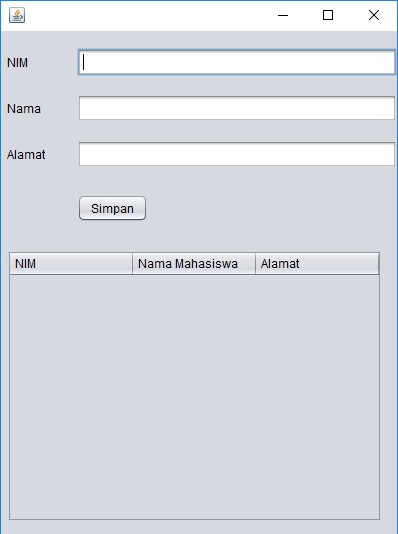
1. Panggil method LihatDataMahasiswa() yang sudah dibuat didalam kontruktor TampilGui()



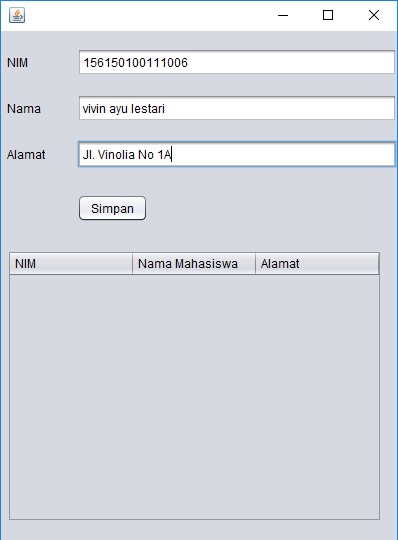
1. Double klik pada Button Simpan kemudian tambahkan kode berikut:



1. Jalankan program



1. Tambahkan data berikut kemudian klik simpan



1. Maka data akan tersimpan pada Jtable

## 7 Percobaan 5 (Aplikasi Biodata)

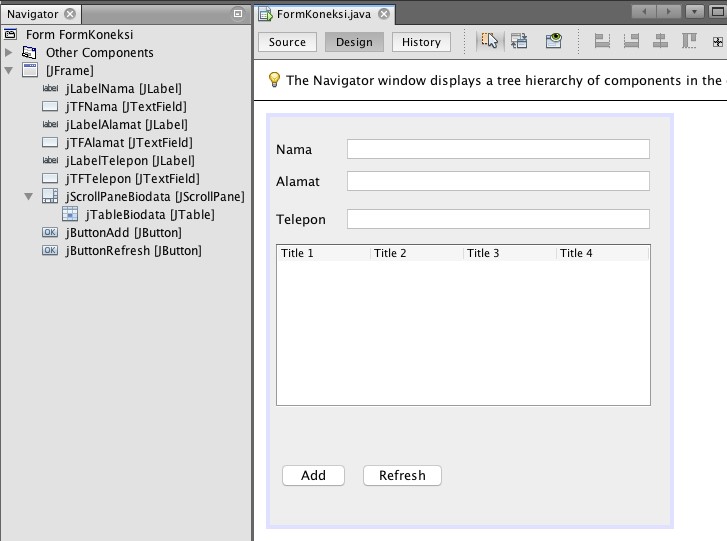
1. Buatlah database dengan MySQL dengan desain tabel sebagai berikut:

CREATE TABLE anggota (id INT NOT NULL AUTO\_INCREMENT, nama

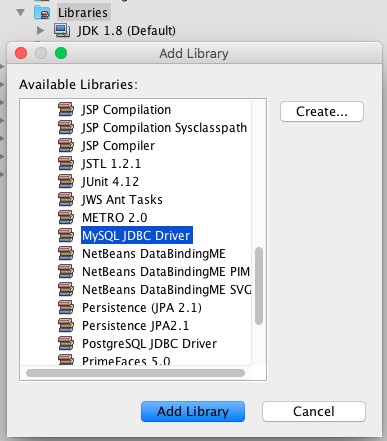
VARCHAR(50) NOT NULL, alamat VARCHAR(100) NOT NULL, telp

VARCHAR(30) NOT NULL, PRIMARY KEY (id ));

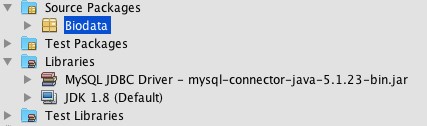
1. Buatlah Package baru dengan nama Biodata atau membuat project baru dengan cara
   1. File → New Project
   2. Pilih categories → Java dengan Projects → Java Application → klik next
   3. Isi nama Project dengan Biodata dan atur letak project akan disimpan
   4. Kemudian tambahkan Jframe Form dengan nama FormKoneksi.
   5. Desain FormKoneksi seperti berikut



1. 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.



1. 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:



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.

1. 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());  } |

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

* Tidak dapat menambahkan karena pada baris program ini

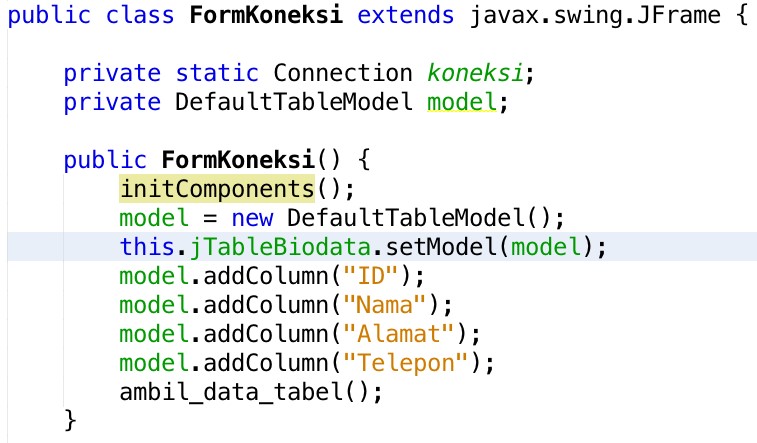
com.mysql.jdbc.Driver()

tidak diketahui.

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

* Mengkoneksikan ke database lalu mengambil nilai yang ada pada text area yang kemudian akan dimasukkan lagi nilainya untuk menambah nilai pada database. Jika data berhasil ditambah maka akan muncul pesan "Data Berhasil Ditambah". Dan jikaa gagal akan muncul pesan gagal.

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



1. 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:



1. Jelaskan alur dari method ambil\_data\_tabel?

* Untuk mengambil data yang ada dalam database

1. Buat fungsi untuk merefresh sehingga data yang baru dapat ditampilkan pada tabel.

    private void refreshActionPerformed(java.awt.event.ActionEvent evt) {

        ambil\_data\_tabel();

    }