

Praktikum 10

Akses Basis Data

1. Tujuan

- Mahasiswa mampu menerapkan JDBC untuk melakukan akses basis data relasional
- Mahasiswa mampu menggunakan persistent object sebagai model basis data relasional
- Mahasiswa mampu menambahkan komponen GUI sebagai antarmuka pengguna

2. Landasan Teori

- Slide Kuliah: Akses Basis Data Relasional dengan JDBC
- Penjelasan slide: https://youtu.be/zp_XeJv055I

3. Langkah Praktikum

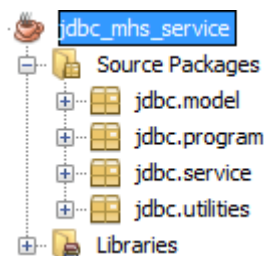
3.1. Akses Basis Data dengan JDBC dan Model

a. Menyiapkan Basis Data (RDBMS)

- Jalankan server basis data, buat basis data
- catat nama basis data, username, dan password yang berhak mengaksesnya
- buat tabel **mahasiswa** dengan atribut **id**, **nama**

| Column Name | Datatype | PK | NN | UQ | B | UN | ZF | AI | G |
|-------------|-------------|-------------------------------------|-------------------------------------|--------------------------|--------------------------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| id | INT(11) | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| nama | VARCHAR(20) | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

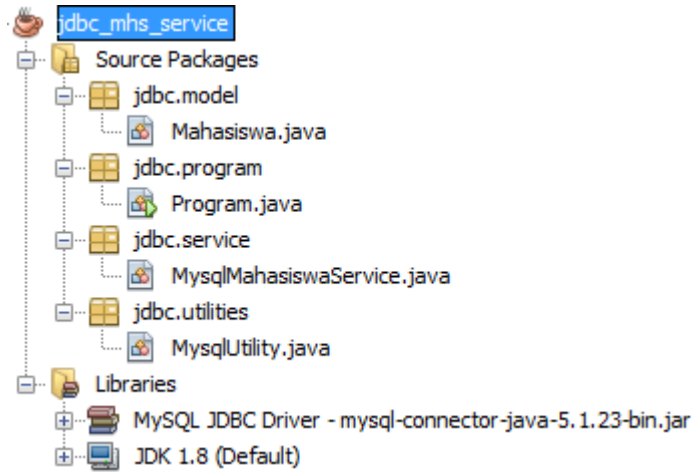
- Buat project jdbc_mhs_service, kemudian tambahkan package: model, program, service, utilities.



Praktikum 10

Akses Basis Data

c. Lengkapi tiap package dengan kelas yang sesuai



Implementasikan kelas-kelas tersebut secara urut sebagai berikut:

1. MysqlUtility.java (utilities)

- Kelas ini digunakan untuk mengelola koneksi dengan basis data yang digunakan
- Contoh disini menggunakan nama MysqlUtility karena basis data yang digunakan adalah MySQL. Silahkan mengubah nama jika menggunakan DBMS yang lain.
- Lengkapi dengan method dan segala keperluan yang sesuai untuk membangun koneksi ke database

```
public class MysqlUtility {  
  
    private static Connection koneksi;  
  
    public static Connection getConnection() {  
        if (koneksi == null) {  
            try {  
                Class.forName("com.mysql.jdbc.Driver");  
                // Sesuaikan host, port, nama db  
                String url = "jdbc:mysql://localhost:3306/jdbc_mhs";  
                // sesuaikan username dan password  
                String user = "student";  
                String password = "rahasia";  
                koneksi = DriverManager.getConnection(url, user, password);  
                if (koneksi != null) {  
                    System.out.println("Koneksi berhasil");  
                }  
            } catch (ClassNotFoundException cne) {  
                System.out.println("Gagal load driver : " + cne.getMessage());  
            } catch (SQLException sqle) {  
                System.out.println("Gagal Koneksi : " + sqle.getMessage());  
            }  
        }  
        return koneksi;  
    }  
}
```

- [Diskusi, tanya jawab]

Praktikum 10

Akses Basis Data

2. Mahasiswa.java (model)

- Kelas ini merupakan persistent object yang berperan sebagai model bagi tabel mahasiswa
- Lengkapi dengan atribut yang sesuai (dengan tabel mahasiswa)
- Lengkapi dengan Constructor yang diperlukan
- Lengkapi dengan getter setter

```
public class Mahasiswa {  
    // atribut  
    private int id;  
    private String nama;  
  
    // Constructor  
    public Mahasiswa() {...3 lines }  
  
    public Mahasiswa(int id, String nama) {...4 lines }  
  
    // getter and setter  
    public int getId() {...3 lines }  
  
    public void setId(int id) {...3 lines }  
  
    public String getNama() {...3 lines }  
  
    public void setNama(String nama) {...3 lines }  
  
    @Override  
    public String toString() {  
        return "Mahasiswa{" + "id=" + id + ", nama=" + nama + '}';  
    }  
}
```

- [Diskusi, tanya jawab]

3. MysqlMahasiswaService.java (service)

- Kelas ini berperan menyediakan layanan (service/controller) bagi program utama dalam mengelola basisdata sehingga interaksi dengan basis data dapat dilakukan secara object oriented (dengan memanfaatkan model dan DB utility)
- Lengkapi dengan Constructor yang diperlukan
- Lengkapi dengan method CRUD yang diperlukan

Praktikum 10

Akses Basis Data

```
public class MysqlMahasiswaService {
    Connection koneksi = null;

    // Constructor
    public MysqlMahasiswaService() {...3 lines }

    /** Membuat objek mahasiswa ...4 lines */
    public Mahasiswa makeMhsObject() {...3 lines }

    /** Menambahkan data mahasiswa ...3 lines */
    public void add(Mahasiswa mhs) {...23 lines }

    /** Update data mahasiswa ...4 lines */
    public void update(Mahasiswa mhs) {...22 lines }

    /** Delete data mahasiswa sesuai id ...5 lines */
    public void delete(int id) {...22 lines }

    /** Ambil mahasiswa sesuai id ...6 lines */
    public Mahasiswa getById(int id) {...33 lines }

    /** Ambil semua isi tabel mahasiswa ...5 lines */
    public List<Mahasiswa> getAll() {...39 lines }
}
```

- [Diskusi, tanya jawab]

4. Program.java (program)

- Kelas ini sebagai program utama
- Implementasikan kode untuk mensimulasikan CRUD

Praktikum 10

Akses Basis Data

```
public class Program {  
  
    static MysqlMahasiswaService service = new MysqlMahasiswaService();  
    public static void main(String[] args) {  
        List<Mahasiswa> listmhs = new ArrayList<>();  
        System.out.println("");  
  
        // insert  
        System.out.println("===insert");  
        Mahasiswa mhsAdd = new Mahasiswa(5, "Haryo");  
        service.add(mhsAdd);  
        System.out.println("berhasil insert: " + mhsAdd);  
        displayAll();  
  
        // update  
        System.out.println("===update");  
        Mahasiswa mhsUpdate = service.getById(5);  
        System.out.println("Akan diupdate data lama: " + mhsUpdate);  
        mhsUpdate.setNama("Dinaya");  
        System.out.println("dengan data baru: " + mhsUpdate);  
        service.update(mhsUpdate);  
        displayAll();  
  
        // delete  
        System.out.println("===delete");  
        System.out.println("akan di delete: " + service.getById(5));  
        service.delete(5);  
        displayAll();  
    }  
}
```

```
run:  
Koneksi berhasil  
  
===insert  
Berhasil insert  
berhasil insert: Mahasiswa{id=5, nama=Haryo}  
===displayAll  
Mahasiswa{id=1, nama=Nina}  
Mahasiswa{id=2, nama=Rudi}  
Mahasiswa{id=3, nama=Beni}  
Mahasiswa{id=5, nama=Haryo}  
  
===update  
Akan diupdate data lama: Mahasiswa{id=5, nama=Haryo}  
dengan data baru: Mahasiswa{id=5, nama=Dinaya}  
Berhasil update  
===displayAll  
Mahasiswa{id=1, nama=Nina}  
Mahasiswa{id=2, nama=Rudi}  
Mahasiswa{id=3, nama=Beni}  
Mahasiswa{id=5, nama=Dinaya}  
  
===delete  
akan di delete: Mahasiswa{id=5, nama=Dinaya}  
Berhasil delete  
===displayAll  
Mahasiswa{id=1, nama=Nina}  
Mahasiswa{id=2, nama=Rudi}  
Mahasiswa{id=3, nama=Beni}
```

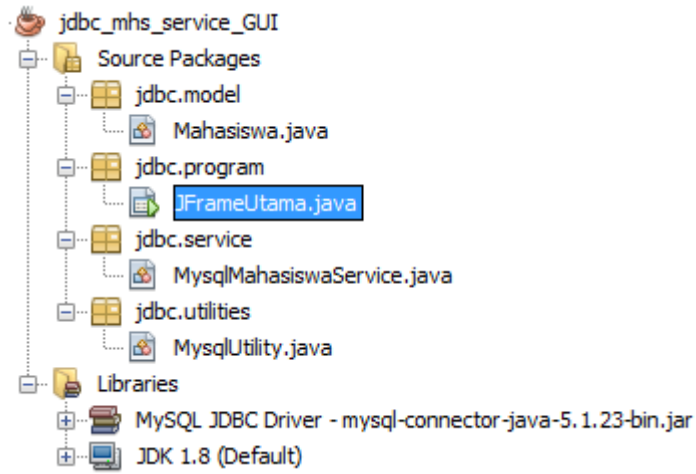
- [Diskusi, tanya jawab]

Praktikum 10

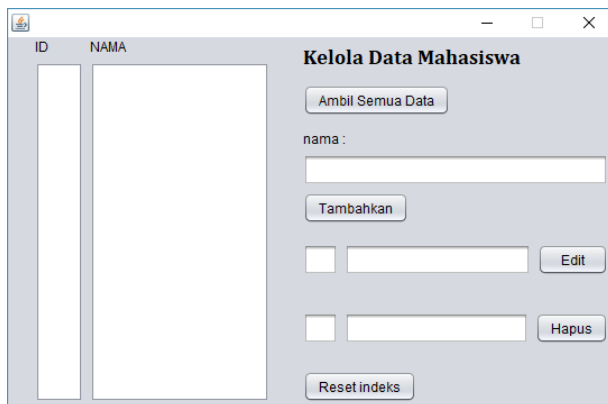
Akses Basis Data

3.2. Menambahkan GUI sebagai antarmuka pengguna

- Salin project `jdbc_mhs_service`, beri nama baru `jdbc_mhs_service_GUI`
- Hapus main class yang lama, gantikan dengan JForm Sebagai main Class



- Rancang GUI yang sesuai untuk mendemonstrasikan aktifitas CRUD



Tambahkan beberapa method pada kelas Service jika dibutuhkan, misalnya sebagai berikut:

Praktikum 10

Akses Basis Data

```
// Constructor
public MysqlMahasiswaService() {...3 lines }

/** Membuat objek mahasiswa ...4 lines */
public Mahasiswa makeMhsObject() {...3 lines }

/** Menambahkan data mahasiswa ...3 lines */
public void add(Mahasiswa mhs) {...23 lines }

/** Update data mahasiswa ...4 lines */
public void update(Mahasiswa mhs) {...22 lines }

/** Delete data mahasiswa sesuai id ...5 lines */
public void delete(int id) {...22 lines }

/** Ambil mahasiswa sesuai id ...6 lines */
public Mahasiswa getById(int id) {...33 lines }

/** Ambil semua isi tabel mahasiswa ...5 lines */
public List<Mahasiswa> getAll() {...39 lines }

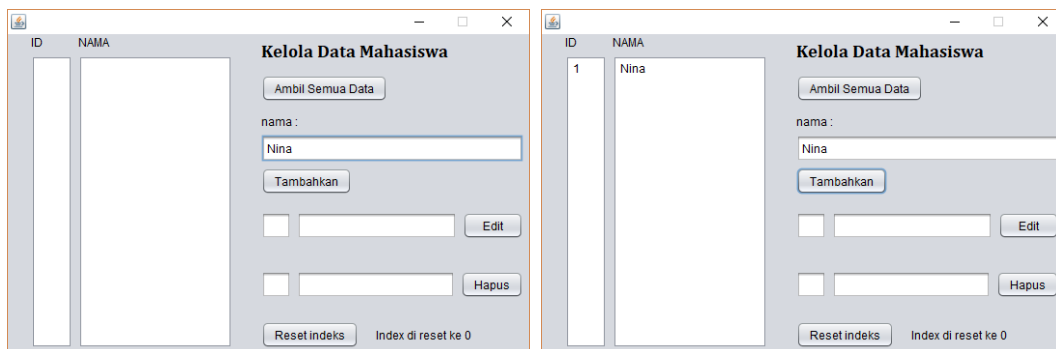
/** Reset indeks tabel mahasiswa ke 1 ...3 lines */
public void indexReset() {...21 lines }

/** Periksa apakah tabel kosong ...5 lines */
public boolean isEmpty() {...23 lines }

/** Menutup koneksi ke DB ...3 lines */
public void closeConnection() {...8 lines }
```

d. Create (insert/add)

Tambahkan **Nina** ke tabel mahasiswa, kemudian tampilkan isi tabel pada GUI



Periksa isi basis data untuk memastikan bahwa Nina berhasil ditambahkan:

```
mysql> select * from jdbc_mhs.mahasiswa;
+-----+-----+
| id | nama |
+-----+-----+
| 1 | Nina |
+-----+-----+
```

Praktikum 10

Akses Basis Data

e. Read (selectAll/getAll)

Tambahkan Rudi dan Beni

The screenshot shows a web application titled "Kelola Data Mahasiswa". On the left, there is a table with two columns: "ID" and "NAMA". The table contains three rows: ID 1 with name "Nina", ID 2 with name "Rudi", and ID 3 with name "Beni". To the right of the table, there are several controls: a button "Ambil Semua Data", a text input field labeled "nama:" containing "Beni", a "Tambahkan" button, an "Edit" button with an empty text input, a "Hapus" button with an empty text input, and a "Reset indeks" button with the text "Index di reset ke 0".

Periksa basis data:

```
mysql> select * from jdbc_mhs.mahasiswa;
+-----+
| id | nama |
+-----+
| 1 | Nina |
| 2 | Rudi |
| 3 | Beni |
+-----+
```

f. Update (edit)

Ubah nama Rudi menjadi Agus

The three screenshots show the application at different stages of the update process. In the first screenshot, the table shows Nina, Rudi, and Beni. The "nama:" input field contains "Beni". In the second screenshot, the "nama:" input field now contains "Agus", and the "Edit" button is highlighted. In the third screenshot, the "nama:" input field is empty, and the "Edit" button is still highlighted. The table and other controls remain the same throughout the sequence.

Periksa basis data:

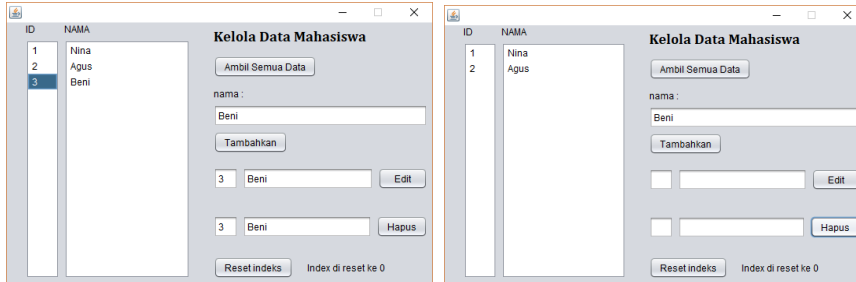
```
mysql> select * from jdbc_mhs.mahasiswa;
+-----+
| id | nama |
+-----+
| 1 | Nina |
| 2 | Agus |
| 3 | Beni |
+-----+
```


Praktikum 10

Akses Basis Data

g. Delete

Hapus Beni dengan cara menyorot ID nomor 3, kemudian klik Hapus



Periksa basis data:

```
mysql> select * from jdbc_mhs.mahasiswa;
+----+-----+
| id | nama |
+----+-----+
| 1  | Nina |
| 2  | Agus |
+----+-----+
```