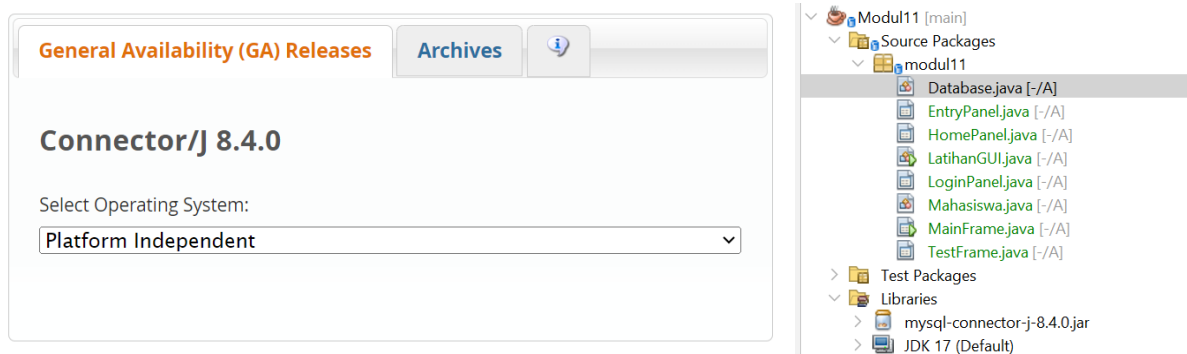


Nama : La Ode Muhammad Gazali
NIM : 222212696
Kelas : 2KS2

MODUL 11 PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK (Database Access/JDBC)

Untuk praktikum kali ini, kita akan melanjutkan GUI yang telah dibuat pada pertemuan sebelumnya yang akan dikembangkan untuk mengintegrasikannya dengan database. Namun, terlebih dulu perlu untuk mendownload MySQL Java Connector. Kemudian add JAR pada library yang ada di panel project.



Kemudian kita perlu membuat database bernama pbo pada localhost. Disini saya membuatnya dengan mysql query pada cmd.

```
MariaDB [(none)]> create database pbo;  
Query OK, 1 row affected (0.003 sec)
```

Kemudian buat table mahasiswa dengan atribut nim, nama, jenis_kelamin, umur, alamat, provinsi, dan hobi, seperti berikut:

```
MariaDB [pbo]> create table mahasiswa (  
-> nim varchar(10) primary key,  
-> nama varchar(100) not null,  
-> jenis_kelamin varchar(50) not null,  
-> umur int(11) not null,  
-> alamat text not null,  
-> provinsi varchar(100) not null,  
-> hobi text not null);  
Query OK, 0 rows affected (0.059 sec)
```

Jika berhasil maka, dapat ditampilkan struktur table nya seperti berikut.

```
MariaDB [pbo]> desc mahasiswa;
```

Field	Type	Null	Key	Default	Extra
nim	varchar(10)	NO	PRI	NULL	
nama	varchar(100)	NO		NULL	
jenis_kelamin	varchar(50)	NO		NULL	
umur	int(11)	NO		NULL	
alamat	text	NO		NULL	
provinsi	varchar(100)	NO		NULL	
hobi	text	NO		NULL	

```
7 rows in set (0.055 sec)
```

Kemudian modifikasi class Database.java dengan kode berikut:

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this
template
 */
package modul11;

import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
import java.sql.*;
import java.util.Arrays;

/**
 *
 * @author U53R
 */
public class Database implements Serializable{
    public static Database instance;
    private final String DB_TYPE = "mysql";
    private final String DB_HOST = "localhost";
    private final String DB_PORT = "3306";
    private final String DB_NAME = "pbo";
    private final String DB_USER = "root";
```

```

private final String DB_PASS = "";
private Database(){
}
public static synchronized Database getInstance(){
    if(instance == null){
        instance = new Database();
    }
    return instance;
}

public void insertMahasiswa(Mahasiswa mahasiswa) throws SQLException{
    Connection conn = getConnection();
    try{
        String sql="INSERT INTO mahasiswa VALUES(?,?,?,?,?,?,?)";
        PreparedStatement pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, mahasiswa.getNim());
        pstmt.setString(2, mahasiswa.getNama());
        pstmt.setString(3, mahasiswa.getJenisKelamin());
        pstmt.setInt(4, mahasiswa.getUmur());
        pstmt.setString(5, mahasiswa.getAlamat());
        pstmt.setString(6, mahasiswa.getProvinsi());
        pstmt.setString(7, String.join(",",
            mahasiswa.getHobi()));
        pstmt.executeUpdate();
    } catch(SQLException ex){
        throw ex;
    } finally{
        if(conn!=null){
            conn.close();
        }
    }
}

public List<Mahasiswa> getListMahasiswa() throws SQLException{
    List<Mahasiswa> mhsList = new ArrayList<>();
    Connection conn = getConnection();
    try{
        String sql = "SELECT * FROM mahasiswa";
        Statement stmt = conn.createStatement();
        ResultSet rs = stmt.executeQuery(sql);
        while(rs.next()){
            Mahasiswa mhs = new Mahasiswa();
            mhs.setNim(rs.getString("nim"));
            mhs.setNama(rs.getString("nama"));
            mhs.setJenisKelamin(rs.getString("jenis_kelamin"));

```

```

        mhs.setUmur(rs.getInt("umur"));
        mhs.setAlamat(rs.getString("alamat"));
        mhs.setProvinsi(rs.getString("provinsi"));
        mhs.setHobi(new
ArrayList<>(Arrays.asList(rs.getString("hobi").split(","))));
        mhsList.add(mhs);
    }
} catch (SQLException ex) {
    throw ex;
} finally {
    if (conn != null) {
        conn.close();
    }
}

return mhsList;
}

private Connection getConnection() throws SQLException {
    return
DriverManager.getConnection("jdbc:" + DB_TYPE + "://" + DB_HOST + ":" + DB_PORT + "/" + DB_NAME
,DB_USER, DB_PASS);
}
}

```

Kemudian ada juga perubahan pada EntryPanel.java pada method insert dan load data table. Jangan lupa untuk menambahkan `import java.sql.SQLException;` pada class EntryPanel.java

```

private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) { //GEN-
FIRST:event_simpanButtonActionPerformed
    // TODO add your handling code here:
    Mahasiswa mhs = new Mahasiswa();
    mhs.setNim(nimTextField.getText());
    mhs.setNama(namaTextField.getText());
    if (lakiRadioButton.isSelected()) {
        mhs.setJenisKelamin("Laki-laki");
    }
    if (perempuanRadioButton.isSelected()) {
        mhs.setJenisKelamin("Perempuan");
    }
    int umur = (Integer) umurSpinner.getValue();
    mhs.setUmur(umur);
    mhs.setAlamat(alamatTextArea.getText());
    mhs.setProvinsi(provinsiComboBox.getSelectedItem().toString());
}

```

```

        ArrayList<String> hobiList = new ArrayList<>();
        if(membacaCheckBox.isSelected()){
            hobiList.add("membaca");
        }
        if(menyanyiCheckBox.isSelected()){
            hobiList.add("menyanyi");
        }
        if(berenangCheckBox.isSelected()){
            hobiList.add("berenang");
        }

        mhs.setHobi(hobiList);

        try {
            Database.getInstance().insertMahasiswa(mhs);
            clearForm();
            JOptionPane.showMessageDialog(this, "Sukses Tersimpan");
            loadTableData();
        } catch (SQLException ex) {
            System.err.println(ex);
            JOptionPane.showMessageDialog(this, "Gagal menyimpandata", "Gagal",
JOptionPane.ERROR_MESSAGE);
        }
    } //GEN-LAST:event_simpanButtonActionPerformed

    private void loadTableData(){
        DefaultTableModel dtm = (DefaultTableModel) mahasiswaTable.getModel();
        //refresh tabel
        while(dtm.getRowCount()>0){
            dtm.removeRow(0);
        }
        try {
            //isi tabel
            for(Mahasiswa mhs:Database.getInstance().getListMahasiswa()){
                dtm.addRow(new
Object[]{mhs.getNim(),mhs.getNama(),mhs.getJenisKelamin(),mhs.getUmur()});
            }
        } catch (SQLException ex) {
            System.err.println(ex);
            JOptionPane.showMessageDialog(this, "Gagal mengambil data", "Gagal",
JOptionPane.ERROR_MESSAGE);
        }
    }
}

```

Jika sudah, lalu jalankan program dan inputkan data melalui GUI yang tersedia, seperti berikut

NIM	Nama	Jenis Kelamin	Umur
111222	Gazali	Laki-laki	21

Selain, akan muncul pada panel form, data akan juga terupdate pada tabel mahasiswa di database

```
MariaDB [pbo]> select * from mahasiswa;
+-----+-----+-----+-----+-----+-----+-----+
| nim   | nama   | jenis_kelamin | umur | alamat       | provinsi       | hobi           |
+-----+-----+-----+-----+-----+-----+-----+
| 111222 | Gazali | Laki-laki     | 21   | Jl. Bonasut I | Sulawesi Tenggara | menyanyi,berenang |
+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.015 sec)
```