

MODUL 3

IMPLEMENTASI DAO

Teori Singkat

DAO merupakan design pattern berkaitan dengan database access. DAO memisahkan persistence logic dari business atau presentation logic. DAO merupakan representasi fisik dan relasi antara object dari database.

DAO design pattern diimplementasi dengan beberapa langkah, yaitu :

- Mendefinisikan interface DAO
- Menulis implementasi dari interface DAO
- Menuliskan business logic dari aplikasi untuk mengakses data source

Latihan 3

1. Buka kembali project yang telah dibuat pada latihan 2 / Modul 2 (project akademik)
2. Pada latihan ini akan di tambahkan fungsi insert, update dan delete data

TAHAP 1

3. Pada tahap satu ini akan dilengkapi source code pada Class SiswaDAOImpl dengan menambahkan metode insertSiswa, updateSiswa dan deleteSiswa

Ubah Clas SiswaDAOImpl menjadi seperti berikut :

```
package akademik.siswa.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

public class SiswaDAOImpl implements SiswaDAO {

    private Connection connection;

    public SiswaDAOImpl() {
        try {
            // 1 - tentukan driver yang digunakan
            Class.forName("org.apache.derby.jdbc.ClientDriver");

            // 2 - tentukan url koneksi
            String url = "jdbc:derby://localhost:1527/akademik";
```

```

        // 3 - buat object connection
        connection = DriverManager.getConnection(url, "root", "root");

    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

public Connection getConnection() {
    return connection;
}

public Siswa getSiswa(String nim) {
    Siswa siswa = new Siswa();
    Connection conn;
    Statement sttmt;

    try {
        //siapkan koneksi yang digunakan
        conn = getConnection();
        //siapkan statement untuk mengeksekusi query
        sttmt = conn.createStatement();
        //ambil data dan letakkan di Resultset
        String query = "select * from siswa where nim='"+ nim + "'";
        ResultSet rsSiswa = sttmt.executeQuery(query);
        rsSiswa.next();
        //bentuk object siswa dari data di Resultset
        siswa.setNim(rsSiswa.getString(1));
        siswa.setNama(rsSiswa.getString(2));
        siswa.setJk(rsSiswa.getInt(3));
        siswa.setKelas(rsSiswa.getString(4));
        siswa.setJurusan(rsSiswa.getString(5));

    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return siswa;
}

```

```

//metode untuk menampilkan semua data siswa
public List getAllSiswa() {
    List listSiswa = new ArrayList<Siswa>();
    Connection conn;
    Statement sttmt;
    try {
        //siapkan koneksi yang digunakan
        conn = getConnection();
        //siapkan statement untuk mengeksekusi query
        sttmt = conn.createStatement();
        //ambil data dan letakkan di Resultset
        String query = "select * from siswa";
        ResultSet rsSiswa = sttmt.executeQuery(query);
        //bentuk object list yang terdiri banyak object siswa
        //dari data di Resultset
        while (rsSiswa.next()) {
            Siswa siswa = new Siswa();
            siswa.setNim(rsSiswa.getString(1));

```

```

        siswa.setNama(rsSiswa.getString(2));
        siswa.setJk(rsSiswa.getInt(3));
        siswa.setKelas(rsSiswa.getString(4));
        siswa.setJurusan(rsSiswa.getString(5));

        listSiswa.add(siswa);
    }
} catch (Exception e) {
    System.out.println(e.getMessage());
}
return listSiswa;
}

```

```

//metode untuk menambah data siswa ke dbms
public void insertSiswa(Siswa siswa) {
    Connection conn;
    Statement sttmt;

    try {
        //siapkan koneksi yang digunakan
        conn = getConnection();
        //siapkan statement untuk mengeksekusi query
        sttmt = conn.createStatement();

        String query =
            "insert into siswa(nim, nama, jk, kelas, jurusan) ";

        query += "values ('"+ siswa.getNim().trim() + "', ";
        query += "'" + siswa.getNama().trim() + "', ";
        query += "'" + siswa.getJk() + "', ";
        query += "'" + siswa.getKelas().trim() + "', ";
        query += "'" + siswa.getJurusan().trim() + "')";

        sttmt.execute(query);
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

```

```

//metode untuk mengubah data siswa
public void updateSiswa(Siswa siswa) {
    Connection conn;
    Statement sttmt;
    try {
        //siapkan koneksi yang digunakan
        conn = getConnection();
        //siapkan statement untuk mengeksekusi query
        sttmt = conn.createStatement();

        String query =
            "update siswa set nama='"+ siswa.getNama().trim() + "', ";
        query += "jk='"+ siswa.getJk() + "', ";
        query += "kelas='"+ siswa.getKelas().trim() + "', ";
        query += "jurusan='"+ siswa.getJurusan().trim() + "' ";
        query += "where nim='"+ siswa.getNim().trim() + "'";

        sttmt.execute(query);
    }
}

```

```
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}

//metode untuk menghapus data siswa
public void deleteSiswa(Siswa siswa) {
    Connection conn;
    Statement sttmt;

    try {
        //siapkan koneksi yang digunakan
        conn = getConnection();
        //siapkan statement untuk mengeksekusi query
        sttmt = conn.createStatement();

        String query = "delete from siswa ";
        query += "where nim='"+ siswa.getNim().trim() +"'";

        System.out.println(query);
        sttmt.execute(query);
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
}
}
```

TAHAP 2

Pada tahap 2 ini akan diimplementasi method insertSiswa, updateSiswa dan deleteSiswa dari SiswaDAOImpl untuk membuat fungsi penambahan, perubahan dan penghapusan data siswa.

Pada Class SiswaModel

1. Ubah Class SiswaModel menjadi seperti berikut :

```
package akademik.siswa;

import akademik.siswa.dao.Siswa;
import akademik.siswa.dao.SiswaDAOImpl;
import java.util.ArrayList;
import java.util.List;

public class SiswaModel {
    private String nim;
    private String nama;
    private int jenisKelamin;
    private String kelas;
    private String jurusan;
    private ArrayList<Object> listeners = new ArrayList<Object>(10);
}
```

```
// deklarasikan object siswaDAO
private SiswaDAOImpl siswaDAO;

public SiswaModel() {
    // instansiasi object
    siswaDAO = new SiswaDAOImpl();
    List lstSiswa = siswaDAO.getAllSiswa();
    int jml = lstSiswa.size();
    if (jml > 0) {
        Siswa siswa = (Siswa) lstSiswa.get(0);
        setNim(siswa.getNim().trim());
        setNama(siswa.getNama().trim());
        setJenisKelamin(siswa.getJk());
        setKelas(siswa.getKelas().trim());
        setJurusan(siswa.getJurusan().trim());
    }
}

public String getNim() {
    return nim;
}

public String getNama() {
    return nama;
}

public int getJenisKelamin() {
    return jenisKelamin;
}

public String getKelas() {
    return kelas;
}

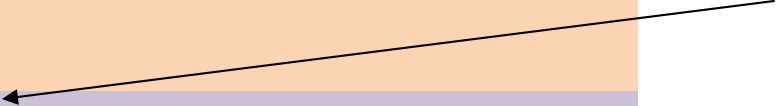
public String getJurusan() {
    return jurusan;
}

public void setNim(String nim) {
    this.nim = nim;
}

public void setNama(String nama) {
    this.nama = nama;
}

public void setJenisKelamin(int jenisKelamin) {
    this.jenisKelamin = jenisKelamin;
}

public void setKelas(String kelas) {
    this.kelas = kelas;
}
}
```



```
public void setJurusan(String jurusan) {
    this.jurusan = jurusan;
}

public void addModelChangeListener(Object view) {
    listeners.add(view);
}

public void removeModelChangeListener(Object view) {
    listeners.remove(view);
}
```

```
public void insertSiswa() {

    Siswa siswa = new Siswa();
    siswa.setNim(getNim());
    siswa.setNama(getNama());
    siswa.setJk(getJenisKelamin());
    siswa.setKelas(getKelas());
    siswa.setJurusan(getJurusan());

    siswaDAO.insertSiswa(siswa);
    fireModelChangeEvent(siswa);
}
```

```
public void updateSiswa() {

    Siswa siswa = new Siswa();
    siswa.setNim(getNim());
    siswa.setNama(getNama());
    siswa.setJk(getJenisKelamin());
    siswa.setKelas(getKelas());
    siswa.setJurusan(getJurusan());

    siswaDAO.updateSiswa(siswa);
    fireModelChangeEvent(siswa);
}
```

```
public void deleteSiswa() {

    Siswa siswa = new Siswa();
    siswa.setNim(getNim());
    siswa.setNama(getNama());
    siswa.setJk(getJenisKelamin());
    siswa.setKelas(getKelas());
    siswa.setJurusan(getJurusan());

    siswaDAO.deleteSiswa(siswa);
    fireModelChangeEvent(siswa);
}
```

```

private void fireModelChangeEvent(Object o) {
    for (Object v : listeners) {
        if
(v.getClass().getName().equals("akademik.siswa.SiswaFormView") == true) {
            SiswaFormView s = (SiswaFormView) v;
            s.processModelChange(v);
        }
        if
(v.getClass().getName().equals("akademik.siswa.SiswaDokumenView") == true)
{
            SiswaDokumenView s = (SiswaDokumenView) v;
            s.processModelChange(v);
        }
    }
}
}
}

```

2. Ubah Class SiswaController menjadi seperti berikut :

```

package akademik.siswa;

import akademik.siswa.dao.Siswa;

public class SiswaController {

    private SiswaModel model;
    private SiswaFormView view1;
    private SiswaDokumenView view2;

    public SiswaController(SiswaModel model, SiswaFormView view) {
        this.model = model;
        this.view1 = view;
        view1.addUserGestureListener(this);
    }

    public SiswaController(SiswaModel model, SiswaDokumenView view) {
        this.model = model;
        this.view2 = view;
        view2.addUserGestureListener(this);
    }

    public void processGetNim() {
        model.getNim();
    }

    public void processGetNama() {
        model.getNama();
    }

    public void processGetJenisKelamin() {
        model.getJenisKelamin();
    }
}

```

```
public void processGetKelas() {
    model.getKelas();
}

public void processUpdateNim(String nim) {
    model.setNim(nim);
}

public void processUpdateNama(String nama) {
    model.setNama(nama);
}

public void processUpdateJenisKelamin(int jenisKelamin) {
    model.setJenisKelamin(jenisKelamin);
}

public void processUpdateKelas(String kelas) {
    model.setKelas(kelas);
}

public void processUpdateJurusan(String jurusan) {
    model.setJurusan(jurusan);
}
```

```
public void processUpdateSiswa(Siswa siswa) {
    processUpdateNim(siswa.getNim().trim());
    processUpdateNama(siswa.getNama().trim());
    processUpdateJenisKelamin(siswa.getJk());
    processUpdateKelas(siswa.getKelas().trim());
    processUpdateJurusan(siswa.getJurusan().trim());

    model.updateSiswa();
}
```

```
public void processInsertSiswa(Siswa siswa) {
    processUpdateNim(siswa.getNim().trim());
    processUpdateNama(siswa.getNama().trim());
    processUpdateJenisKelamin(siswa.getJk());
    processUpdateKelas(siswa.getKelas().trim());
    processUpdateJurusan(siswa.getJurusan().trim());

    model.insertSiswa();
}
```

```
public void processDeleteSiswa(Siswa siswa) {
    processUpdateNim(siswa.getNim().trim());
    processUpdateNama(siswa.getNama().trim());
    processUpdateJenisKelamin(siswa.getJk());
    processUpdateKelas(siswa.getKelas().trim());
    processUpdateJurusan(siswa.getJurusan().trim());

    model.deleteSiswa();
}
```

```
}
```


3. Ubah Class SiswaFormView menjadi seperti berikut :

The screenshot shows a Java Swing form with the following fields and controls:

- NIM: Text input field
- Nama: Text input field
- Jenis Kelamin: Radio buttons for "Laki-Laki" and "Perempuan"
- Kelas: Dropdown menu with "Satu" selected
- Jurusan: List box containing "IPA", "IPS", and "BAHASA"
- Buttons: "jButton1", "Update", "jButton2", and "jButton3"
- Table: A table with 4 columns (Title 1, Title 2, Title 3, Title 4) and 4 rows.

Callouts point to the following buttons:

- btnNew: Points to jButton1
- btnInsert: Points to jButton2
- btnDelete: Points to jButton3

Sehingga menjadi sbb :

The updated Java Swing form has the following fields and controls:

- NIM: Text input field
- Nama: Text input field
- Jenis Kelamin: Radio buttons for "Laki-Laki" and "Perempuan"
- Kelas: Dropdown menu with "Satu" selected
- Jurusan: List box containing "IPA", "IPS", and "BAHASA"
- Buttons: "New", "Update", "Insert", and "Delete"
- Table: A table with 4 columns (Title 1, Title 2, Title 3, Title 4) and 4 rows.

4. Ubah Listing Code Class SiswaFormView menjadi seperti berikut :

```
package akademik.siswa;

import akademik.siswa.dao.Siswa;
import java.util.ArrayList;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

public class SiswaFormView
    extends javax.swing.JFrame implements ListSelectionListener {

    private SiswaModel model;
    private ArrayList<SiswaController> listeners =
        new ArrayList<SiswaController>();

    //buat object SiswaModelTable menampilkan data dalam bentuk model
    private SiswaModelTable modelTabel;

    public SiswaFormView(SiswaModel model) {
        this.model = model;
        initComponents();

        modelTabel = new SiswaModelTable();
        tabelSiswa.getSelectionModel().addListSelectionListener(this);
        tabelSiswa.setModel(modelTabel);

        refresh();
        setVisible(true);
        setDefaultCloseOperation(DISPOSE_ON_CLOSE);
        model.addModelChangeListener(this);
    }

    public void addUserGestureListener(SiswaController controller) {
        listeners.add(controller);
    }

    public void processModelChange(Object event) {
        refresh();
    }

    private void refresh() {
        String nim = model.getNim();
        String nama = model.getNama();
        int jenisKelamin = model.getJenisKelamin();
        String kelas = model.getKelas();
        String jurusan = model.getJurusan();

        txtNIM.setText(nim);
        txtNama.setText(nama);

        if (jenisKelamin == 0) rdoLaki.setSelected(true);
        else rdoPerempuan.setSelected(true);
    }
}
```

```

        cbKelas.setSelectedItem(kelas);
        lstJurusan.setSelectedValue(jurusan, true);
    }

private Siswa getUIValue() {
    String nim = txtNIM.getText().trim();
    String nama = txtNama.getText().trim();
    int jeniskelamin = (rdoLaki.isSelected()== true ? 0 : 1);
    String kelas = cbKelas.getSelectedItem().toString().trim();
    String jurusan = lstJurusan.getSelectedValue().toString().trim();

    Siswa siswa = new Siswa(nim,nama,jeniskelamin,kelas,jurusan);
    return siswa;
}

private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    Siswa siswa = getUIValue();

    for (SiswaController con : listeners) {
        con.processUpdateSiswa(siswa);
    }
}

private void btnInsertActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    Siswa siswa = getUIValue();

    for (SiswaController con : listeners) {
        con.processInsertSiswa(siswa);
    }
    modelTabel.add(siswa);
}

private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    Siswa siswa = getUIValue();

    for (SiswaController con : listeners) {
        con.processDeleteSiswa(siswa);
    }
    int index = tabelSiswa.getSelectedRow();
    modelTabel.remove(index);
}

public void valueChanged(ListSelectionEvent e) {
    try {
        Siswa siswa = modelTabel.get(tabelSiswa.getSelectedRow());
    } catch (IndexOutOfBoundsException exception) {
    }
}
}

```

```
// Variables declaration - do not modify
private javax.swing.JButton btnDelete;
private javax.swing.JButton btnInsert;
private javax.swing.JButton btnNew;
private javax.swing.JButton btnUpdate;
private javax.swing.ButtonGroup buttonGroup1;
private javax.swing.JComboBox cbKelas;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JList lstJurusan;
private javax.swing.JRadioButton rdoLaki;
private javax.swing.JRadioButton rdoPerempuan;
private javax.swing.JTable tabelSiswa;
private javax.swing.JTextField txtNIM;
private javax.swing.JTextField txtNama;
// End of variables declaration
}
```