

MODUL PRAKTIKUM 11

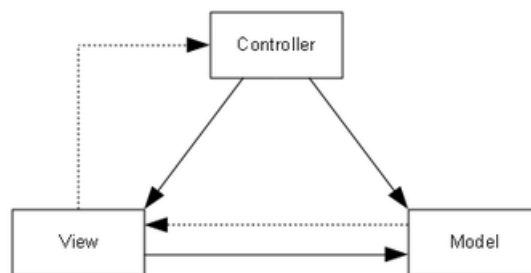
IMPLEMENTASI MVC (Model-View-Controller) PADA JAVA

A. Dasar Teori

➤ Pengertian MVC

MVC merupakan sebuah konsep dalam membangun sebuah aplikasi dengan memisahkan antara data dari tampilan dan aksi pemrosesannya. Model berfungsi sebagai sumber data. View berfungsi sebagai desain interface yang berinteraksi langsung dengan user. Controller berguna sebagai “otak” yang memproses data yang dilakukan oleh user.

Sekilas MVC kelihatannya merepotkan karena harus menulis kode lebih banyak tapi keuntungannya adalah kode kita lebih maintainable karena bisa mengubah salah satu bagian tanpa harus mengubah bagian yang lain. Arsitektur MVC ini memungkinkan adanya perubahan dalam domain model tanpa harus mengubah code untuk menampilkan domain model tersebut. Hal ini sangat bermanfaat ketika aplikasi mempunyai domain model dan view komponen sangat besar dan kompleks.



Gambar : Alur kerja MVC

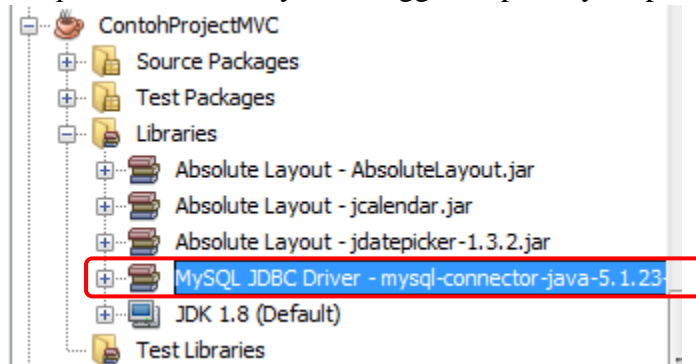
➤ Pengertian DAO

Data Access Object (DAO) merupakan sebuah *object* yang menyediakan sebuah *abstract interface* terhadap beberapa database atau mekanisme persistence, menyediakan beberapa operasi tertentu tanpa mengekspos detail database. Penerapan konsep ini sering disebut dengan *separation of concern* dimana setiap kode dipisahkan berdasarkan fungsinya sehingga kode di atasnya hanya perlu mengetahui secara abstrak cara mengakses data tanpa perlu mengetahui bagaimana akses ke sumber data diimplementasikan.

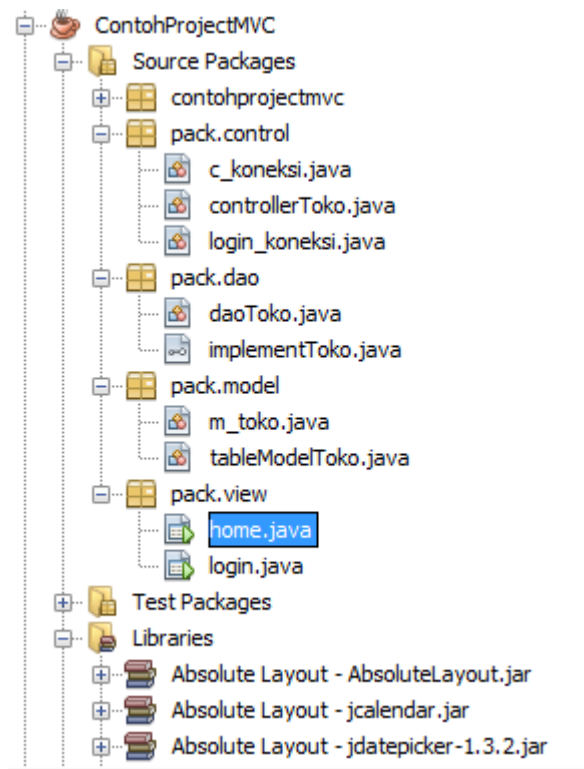
B. Praktikum

Konsep MVC ini menggunakan studi kasus pembuatan data rekap barang di sebuah toko dengan database MySQL. Langkah-langkah pembuatannya dengan konsep MVC dengan fungsi CRUD (Create-Read-Update-Delete) dan fungsi pencarian, sebagai berikut.

1. Import library MySQL, Klik kanan pada Libraries dari project anda, kemudian pilih “Add Library”. Sehingga tampilannya seperti ini.



2. Buatlah package-package untuk model, view, controller, dao, koneksi, dan main. Yang bertujuan agar lebih terstruktur dalam peng-konsep-an MVC, lihat Gambar berikut.



3. Membuat class **control** untuk koneksi, Buatlah Java class dengan nama “**c_koneksi.java**” dan tempatkan pada package “**control**”, kemudian isikan dengan code berikut.

File **c_koneksi.java**

```

package pack.control;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;

import javax.swing.JOptionPane;
/**
 *
 * @author ARIFIN
 */
public class c_koneksi {

    public static ResultSet executeQuery(String SQL) {
        ResultSet rs = null;
        Connection koneksi = setKoneksi();
        try {
            Statement st = koneksi.createStatement();
            rs = st.executeQuery(SQL);
        } catch (SQLException ex) {

            Logger.getLogger(c_koneksi.class.getName()).log(Level.SEVERE, null, ex);
        }
        return rs;
    }

    public static int execute(String SQL) {
        int status = 0;
        Connection koneksi = setKoneksi();
        try {
            Statement st = koneksi.createStatement();

            status = st.executeUpdate(SQL);
        } catch (SQLException ex) {

            Logger.getLogger(c_koneksi.class.getName()).log(Level.SEVERE, null, ex);
        }
        return status;
    }

    public static Connection setKoneksi() {
        String konString =
            "jdbc:mysql://localhost:3306/toko";
        Connection koneksi = null;
        try {
            Class.forName("com.mysql.jdbc.Driver");

```

```

        koneksi = (Connection)
DriverManager.getConnection(konString,"root","");
        System.out.println("Koneksi Berhasil");
    } catch (ClassNotFoundException ex) {

Logger.getLogger(c_koneksi.class.getName()).log(Level.SEVERE, null, ex);
        System.out.println("Koneksi Gagal");
    } catch (SQLException ex) {

Logger.getLogger(c_koneksi.class.getName()).log(Level.SEVERE, null, ex);
        System.out.println("Koneksi Gagal");
    }
    return koneksi;
}

    public static Object getConnection() {
        return null;
    }

Connection con = null;
Statement st = null;
ResultSet rs = null;
String sql = null;

public c_koneksi(){
    try
    {
        con
=DriverManager.getConnection("jdbc:mysql://localhost:3306
/toko","root","");
        st = con.createStatement();
    }
    catch (Exception a)
    {
        JOptionPane.showMessageDialog(null,"Koneksi Database
Gagal"+ a);
    }
}
}

```

File login_koneksi.java

```

package pack.control;

import com.mysql.jdbc.Connection;
import com.mysql.jdbc.Driver;

```

```

import java.sql.DriverManager;
import java.sql.SQLException;

/**
 * @author ARIFIN
 */
public class login_koneksi {
    private static Connection koneksi;
    public static Connection GetConnection() throws
    SQLException
    {
        if (koneksi == null) {
            Driver driver = new Driver();
            koneksi = (Connection)
            DriverManager.getConnection("jdbc:mysql://localhost/toko?
            zeroDateTimeBehavior=convertToNull", "root", "");
        }
        return koneksi;
    }
}

```

Nama database: “toko”. nama user MySQL yaitu “**root**”, sedangkan dikosongkan karena MySQL tidak menggunakan password sehingga hanya ditandai dengan string kosong (“”).

Kemudian membuat class toko di **package control** dengan nama **controllerToko.java**, kemudian isikan code berikut

```

package pack.control;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableModel;
import pack.dao.daoToko;
import pack.dao.implementToko;
import pack.model.m_toko;
import pack.view.home;
import pack.model.tableModelToko;

/**
 *
 * @author ARIFIN
 */
public class controllerToko {
    home hm;
    implementToko impToko;
    List<m_toko> lt;
}

```

```

    public controllerToko(home hm){
        this.hm = hm;
        impToko = new daoToko();
        lt = impToko.getAll();
    }

//mengkosongkan isian field
    public void Reset(){
        hm.getTxtKode().setText("");
        hm.getTxtNama().setText("");
        hm.getTxtHarga().setText("");
        hm.getCbKategori().setSelectedItem(null);
        hm.getCbJenis().setSelectedItem(null);
    }

//menghapus data yang dipilih
    public void Hapus(){
        if (!hm.getTxtKode().getText().trim().isEmpty()){

        }else{
            String kode = (hm.getTxtKode().getText());
            impToko.HapusData(kode);
            JOptionPane.showMessageDialog(hm, "Data berhasil
dihapus");
        }
    }
    //menyimpan data
    public void SimpanData(){
        m_toko toko = new m_toko();
        toko.setkode(hm.getTxtKode().getText());
        toko.setnama(hm.getTxtNama().getText());
        toko.setharga(hm.getTxtHarga().getText());

        toko.setkategori(hm.getCbKategori().getSelectedItem().toString());
        toko.setjenis(hm.getCbJenis().getSelectedItem().toString());
        impToko.SimpanData(toko);
    }
    //mengubah data
    public void Ubah(){
        m_toko toko = new m_toko();
        toko.setkode(hm.getTxtKode().getText());
        toko.setnama(hm.getTxtNama().getText());

        toko.setkategori(hm.getCbKategori().getSelectedItem().toString());
        toko.setjenis(hm.getCbJenis().getSelectedItem().toString());
        toko.setharga(hm.getTxtHarga().getText());
        impToko.UbahData(toko);
    }

    public void isiTable() {
        lt =impToko.getAll();
        tableModelToko tmt = new tableModelToko(lt);
    }

```

```

        hm.getTableData().setModel(tmt);

    }
    public void isiField(int row){
        hm.getTxtKode().setText(lt.get(row).getkode().toString());
        hm.getTxtNama().setText(lt.get(row).getnama().toString());

        hm.getCbKategori().setSelectedItem(lt.get(row).getkategori().toString());

        hm.getCbJenis().setSelectedItem(lt.get(row).getjenis().toString());
        hm.getTxtHarga().setText(lt.get(row).getharga().toString());

    }
    public void CariKategori(){
        if
(hm.getCbCariKategori().getSelectedItem().toString().isEmpty()){

        impToko.getCariKategori((String)hm.getCbCariKategori().getSelectedItem());

            isiTableCariKategori();
        }else{
            JOptionPane.showMessageDialog(hm, "Silahkan Pilih Kategori");
        }
    }

    private void isiTableCariKategori() {
        lt = impToko.getCariKategori((String)
hm.getCbKategori().getSelectedItem());
        tableModelToko tmt = new tableModelToko(lt);
        hm.getTableData().setModel(tmt);

    }
}

```

4. Kemudian setelah selesai, buatlah file “**m_toko.java**” pada package “**model**” yang berisi method set dan get.

```

package pack.model;

/**
 *
 * @author ARIFIN
 */
public class m_toko {
    public String Nama_Barang, Jenis, Harga, Kategori;
    public String Kode_Barang;

    public String getkode()

```

```

{
    return Kode_Barang;
}
public String getnama()
{
    return Nama_Barang;
}
public String getkategori()
{
    return Kategori;
}

public String getjenis()
{
    return Jenis;
}
public String getharga()
{
    return Harga;
}

//method set
public void setkode(String kode)
{
    this.Kode_Barang= kode;
}
public void setnama(String nama)
{
    this>Nama_Barang= nama;
}
public void setkategori(String kategori)
{
    this.Kategori=kategori;
}
public void setjenis(String jenis)
{
    this.Jenis= jenis;
}
public void setharga(String harga)
{
    this.Harga= harga;
}
}

```

kode, nama, kategori, jenis dan harga merupakan nama dari attribute yang ada di database. Fungsi **setter** (method set) and **getter** (method get) tersebut untuk

mengambil data dari database dan memanggilnya, sebagai perantara penyaluran data.

Kemudian buatlah file “tableModelToko.java” pada package “model” dan isikan dengan code berikut:

```
package pack.model;

import java.util.List;
import javax.swing.table.AbstractTableModel;

/**
 * @author ARIFIN
 */
public class tableModelToko extends AbstractTableModel{
    List<m_toko>lt;
    public tableModelToko(List<m_toko> lt) {
        this.lt=lt;
    }

    @Override
    public int getRowCount() {
        return lt.size();
    }

    @Override
    public int getColumnCount() {
        return 5;
    }
    public String getColumnName(int column) {
        switch (column){
            case 0:
                return"Kode_Barang";
            case 1:
                return>Nama_Barang";
            case 2:
                return"Kategori";
            case 3:
                return"Jenis";
            case 4:
                return"Harga";
            default:
                return null;
        }
    }
    @Override
    public Object getValueAt(int row, int column) {
        switch (column){
            case 0:
                return lt.get(row).getkode();
            case 1:
                return lt.get(row).getnama();
            case 2:
                return lt.get(row).getkategori();
            case 3:
                return lt.get(row).getjenis();
            case 4:
                return lt.get(row).getharga();
        }
    }
}
```

```

        return lt.get(row).getharga();
    default:
        return null;
    }
}
}

```

Tabel model berguna untuk mengambil data dari database yang akan ditampilkan pada JTable yang ada di Frame. Data disimpan dalam List. Yang kemudian dapat dipanggil untuk ditampilkan di JTable.

5. Membuat Desain Interface, Buatlah frame desain aplikasi yang ditempatkan pada package “view”
Frame pertama dengan nama login dan kedua dengan nama home.



Code untuk **Button SignIn**

```

private void SignInActionPerformed(java.awt.event.ActionEvent
evt) {
    Connection connection;
    PreparedStatement ps;
    try {
        connection
        DriverManager.getConnection("jdbc:mysql://localhost/toko?zeroDate
TimeBehavior=convertToNull", "root", "");
        ps = connection.prepareStatement("SELECT * FROM
`tb_akun` WHERE `username` = ? AND `password` = ?");
        ps.setString(1, txtnama.getText());
        ps.setString(2, txtpass.getText());
        ResultSet result =ps.executeQuery();
        if(result.next()){
            new home().show();
            user = txtnama.getText();//perlu deklarasi user
diclass utama.
            this.dispose();
        }
        else{
            JOptionPane.showMessageDialog(rootPane,
"Salah!");
            txtpass.setText("");
        }
    } catch (SQLException ex) {
        Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);
    }
}

```

```

        txtnama.requestFocus();
    }
} catch (SQLException ex) {
    JOptionPane.showMessageDialog(rootPane, "Gagal!");
}
}

```

Frame Home

Admin Toko :

Data Barang di Toko

Kode Barang

Nama Barang

Kategori Barang

Makanan

Jenis Packaging

Pack

Harga

Simpan

Ubah

Bersihkan

Hapus

Keluar

Pencarian Berdasarkan Kategori Makanan

Makanan

Cari

Kode Barang	Nama Barang	Kategori	Packaging	Harga

Code untuk penambahan import java

```

package pack.view;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JOptionPane;
import javax.swing.JTable;
import javax.swing.JTextField;
import javax.swing.table.DefaultTableModel;
import pack.control.c_koneksi;
import pack.control.controllerToko;
import pack.model.m_toko;

```

Penambahan coding pada class home dan konstruktor home

```

public class home extends javax.swing.JFrame {
    controllerToko ctoko;
    List<m_toko>listdata = new ArrayList<>();

    /**
     * Creates new form home
     */
    public home() {
        initComponents();
        ctoko = new controllerToko (this);
        ctoko.isiTable();
    }
}

```

Coding untuk Button Simpan

```

private void Btn_SimpanActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ctoko.SimpanData();
    ctoko.isiTable();
}

```

Coding untuk Button Ubah

```

private void Btn_UbahActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ctoko.Ubah();
    ctoko.isiTable();
}

```

Coding untuk Button Bersihkan

```

private void Btn_BersihActionPerformed(java.awt.event.ActionEvent evt) {
    ctoko.Reset();
}

```

Coding untuk Button Hapus

```

private void Btn_HapusActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ctoko.Hapus();
    ctoko.isiTable();
}

```

Coding untuk Button Keluar

```

private void Btn_KeluarActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    dispose();
}

```

Coding untuk Button Cari

```

private void Btn_CariActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    ctoko.CariKategori();
    ctoko.isiTable();
}

```

Tambahkan method berikut di frame home

```

public JTextField getTxtKode() {

```

```

        return txtkode;
    }
    public JTextField getTxtNama() {
        return txtnama;
    }
    public JTextField getTxtHarga() {
        return txtharga;
    }
    public JComboBox getCbKategori() {
        return cbKategoriBrg;
    }
    public JComboBox getCbJenis() {
        return cbjenis;
    }
    public JComboBox getCbCariKategori() {
        return CbCariKategori;
    }
    public JButton getButtonHapus() {
        return Btn_Hapus;
    }

    public JButton getButtonBersih() {
        return Btn_Bersih;
    }
    public JButton getButtonSimpan() {
        return Btn_Simpan;
    }
    public JButton getButtonUbah() {
        return Btn_Ubah;
    }
    public JButton getButtonKeluar() {
        return Btn_Keluar;
    }
    public JTable getTableData() {
        return Tabell1;
    }
}

```

6. Membuat Fungsi DAO

Buatlah Class Interface dengan nama “implementToko.java” yang diletakkan pada package “dao”. Kemudian isikan code berikut:

```

package pack.dao;
import java.util.List;
import pack.model.m_toko;

/**
 * @author ARIFIN
 */
public interface implementToko {

    public void TampilData(m_toko a);
    public void UbahData(m_toko a);
    public void SimpanData(m_toko a);
    public void HapusData(String kode);
    public List<m_toko> getCariKategori(String kategori);
}

```

```
public List<m_toko> getAll();  
}
```

Pada class `implementToko.java` terdapat method-method yang belum terimplementasi. Method pada class interface digunakan sebagai method inti yang wajib dideklarasikan oleh subclass yang mengimplement class interface tersebut. Kemudian buatlah class “`daoToko.java`” pada package “`dao`” dan isikan dengan code berikut:

```
package pack.dao;  
  
import java.sql.Connection;  
import java.sql.PreparedStatement;  
import java.sql.ResultSet;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.logging.Level;  
import java.util.logging.Logger;  
import pack.control.c_koneksi;  
import pack.model.m_toko;  
  
/**  
 *  
 * @author ARIFIN  
 */  
public class daoToko implements implementToko {  
    Connection connection;  
    public String TampilData = "SELECT * FROM `barang`";  
    public String UbahData = "UPDATE `barang` SET  
`Nama_Barang`=?, `Harga`=?, `Kategori`=?, `Jenis`=? Where  
`Kode_Barang`=?";  
    public String SimpanData = "INSERT INTO `barang` VALUES  
(?, ?, ?, ?, ?)";  
    public String HapusData = "DELETE FROM `barang`";  
    public String CariKategori = "SELECT `Kode_Barang`,  
`Nama_Barang`, `Kategori`, `Jenis`, `Harga` FROM `barang`  
WHERE Kategori like ?";  
  
    public daoToko(){  
        connection = c_koneksi.setKoneksi();  
    }  
  
    @Override  
    public void TampilData(m_toko a) {  
  
    }  
  
    @Override
```

```

public void UbahData(m_toko a) {
    PreparedStatement statement = null;
    try {
        statement = connection.prepareStatement(UbahData);
        statement.setString(1, a.getkode());
        statement.setString(2, a.getnama());
        statement.setString(3, a.getkategori());
        statement.setString(4, a.getjenis());
        statement.setString(5, a.getharga());
        statement.executeUpdate();

    } catch (SQLException ex) {

        Logger.getLogger(daoToko.class.getName()).log(Level.SEVERE,
        null, ex);
    }
}

@Override
public void SimpanData(m_toko a) {
    PreparedStatement statement = null;
    try {
        statement = connection.prepareStatement(SimpanData);
        statement.setString(1, a.getkode());
        statement.setString(2, a.getnama());
        statement.setString(3, a.getkategori());
        statement.setString(4, a.getjenis());
        statement.setString(5, a.getharga());
        statement.executeUpdate();
        ResultSet rs = statement.getGeneratedKeys();
        while (rs.next()){
            a.setkode(rs.getString(1));
        }
    } catch (SQLException ex) {

        Logger.getLogger(daoToko.class.getName()).log(Level.SEVERE,
        null, ex);
    }
}

//menampilkan data ke tabel sesuai pencarian
@Override
public List<m_toko> getCariKategori(String kategori) {
    List<m_toko> lt=null;
    try {
        lt = new ArrayList<m_toko>();
        PreparedStatement
st=connection.prepareStatement(CariKategori);
        st.setString(1, "%" + kategori + "%");
        ResultSet rs = st.executeQuery();

```

```

        while (rs.next()){
            m_toko toko = new m_toko();
            toko.setkode(rs.getString("Kode_Barang"));
            toko.setnama(rs.getString("Nama_Barang"));
            toko.setkategori(rs.getString("Kategori"));
            toko.setjenis(rs.getString("Jenis"));
            toko.setharga(rs.getString("Harga"));
            lt.add(toko);
        }
    } catch (SQLException ex) {

Logger.getLogger(daoToko.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return lt;
}

//menampilkan data ke Tabel
@Override
public List<m_toko> getAll() {
    List<m_toko> lt=null;
    try {
        lt = new ArrayList<m_toko>();
        Statement st = connection.createStatement();
        ResultSet rs = st.executeQuery(TampilData);
        while (rs.next()){
            m_toko toko = new m_toko();
            toko.setkode(rs.getString("Kode_Barang"));
            toko.setnama(rs.getString("Nama_Barang"));
            toko.setkategori(rs.getString("Kategori"));
            toko.setjenis(rs.getString("Jenis"));
            toko.setharga(rs.getString("Harga"));
            lt.add(toko);
        }
    } catch (SQLException ex) {

Logger.getLogger(daoToko.class.getName()).log(Level.SEVERE,
null, ex);
    }
    return lt;
}

//hapus data
@Override
public void HapusData(String kode) {
    PreparedStatement statement = null;
    try {
        statement = connection.prepareStatement(HapusData);
        statement.setString(1, kode);
        statement.executeUpdate();
    } catch (SQLException ex) {

```



```

Logger.getLogger(daoToko.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
//pencarian data
public void CariKategori(int kode) {
    PreparedStatement statement = null;
    try {
        statement = connection.prepareStatement(CariKategori);
        statement.setInt(1, kode);
        statement.executeUpdate();

    } catch (SQLException ex) {

Logger.getLogger(daoToko.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
}

```

- Setelah selesai cek terlebih dahulu sebelum dijalankan programnya.

**Kesuksesan Akan Bersama
Mereka yang Selalu Menjaga Sikap
dan Semangat yang Tinggi**

-- SISWA TELKOM LUAR BIASA --