

Modul : **IK-481 Pemrograman Berorientasi Objek**
Topik : **Contoh Implementasi MVC (*Model View Controller*) pada Java**
Dosen : **Rosa Ariani Sukamto**
Website : **<http://www.gangsir.com>**
Blog : **<http://udinrosa.wordpress.com>**

Yang diperlukan:

plugin:

mysql-connector-java-3.1.8-bin.jar

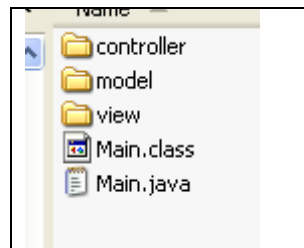
Masukkan *connector* tersebut ke dalam direktori pada tempat jdk di-*instal* dengan *path*

<INSTALL_jdk>/jre/lib/ext

dan

<INSTALL_jre>/lib/ext

Susunan penataan *file*:

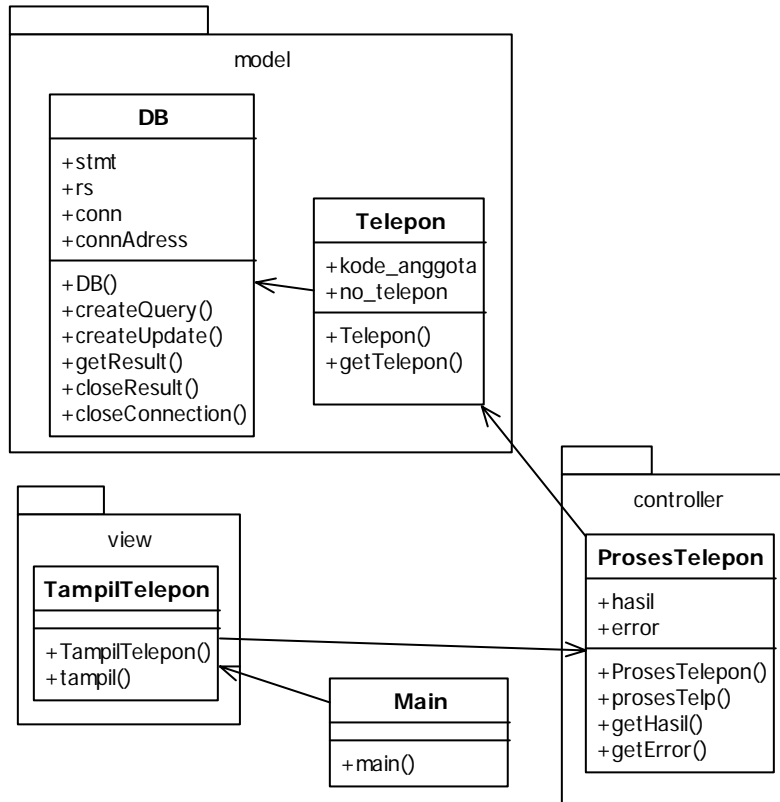


Setiap mengganti kode, lakukan kompilasi kembali pada setiap *package*:

misal:

```
javac model\*.java
javac controller\*.java
javac view\*.java
javac Main.java
java Main
```

Diagram Kelas:



```

/*****
 * Filename : DB.java
 * Programmer : Rosa Ariani Sukamto
 * Date : 2009-01-13
 * Email : rosa_if_itb_01@yahoo.com
 * Website : http://www.gangsir.com
 * Deskripsi : package model untuk mengakses basis data
 *
 *****/

package model;//package model/kelas yang mengakses basis data

//import konektor
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.ResultSet;

//kelas DB
public class DB {
    private String ConAddress = "jdbc:mysql://localhost:3306/penyewaan_vcd?user=root&password=";
    /*konfigurasi koneksi==>jdbc:mysql://host:port/nama_basis
data?user=nama_user&password=isi_password*/
    private Statement stmt = null; // koneksi query
    private ResultSet rs = null; // hasil query
    private Connection conn = null; // koneksi MySQL dan basis data

    public DB() throws Exception, SQLException {
        /**
         * Method DB
         * Konstruktor : melakukan koneksi ke MySQL dan basis data
         * Menerima masukan berupa string alamat koneksi ke MySQL dan basis data
         */
        try {
            // membuat driver MySQL
            Class.forName("com.mysql.jdbc.Driver").newInstance();
            // membuat koneksi MySQL dan basis data
            conn = DriverManager.getConnection(ConAddress);
            conn.setTransactionIsolation(conn.TRANSACTION_READ_UNCOMMITTED);
        }
        catch(SQLException es) {
            // mengeluarkan pesan error jika koneksi gagal
            throw es;
        }
    }

    public void createQuery(String Query)throws Exception, SQLException {
        /**
         * Method createQuery
         * Mengeksekusi query tanpa mengubah isi data
         * Menerima masukan berupa string query
         */
        try {

```

```

    stmt = conn.createStatement();
    // eksekusi query
    rs = stmt.executeQuery(Query);
    if (stmt.execute(Query)) {
        // ambil hasil query
        rs = stmt.getResultSet();
    }
}
catch(SQLException es) {
    // eksepsi jika query gagal dieksekusi
    throw es;
}
}

public void createUpdate(String Query)throws Exception, SQLException {
/**
 * Method createQuery
 * Mengeksekusi query yang mengubah isi data (update, insert, delete)
 * Menerima masukan berupa string query
 */
try {
    stmt = conn.createStatement();
    // eksekusi query
    int hasil = stmt.executeUpdate(Query);
}
catch(SQLException es) {
    // eksepsi jika query gagal dieksekusi
    throw es;
}
}

public ResultSet getResult()throws Exception {
/**
 * Method getResult
 * Memberikan hasil query
 */
ResultSet Temp = null;
try{
    return rs;
}
catch (Exception ex) {
    // eksepsi jika hasil tidak dapat dikembalikan
    return Temp;
}
}

public void closeResult()throws SQLException, Exception {
/**
 * Method closeResult
 * Menutup hubungan dari eksekusi query
 */
if (rs != null) {
    try {
        rs.close();

```

```
    }
    catch (SQLException sqlEx) {
        rs = null;
        throw sqlEx;
    }
}
if (stmt != null) {
    try {
        stmt.close();
    }
    catch (SQLException sqlEx) {
        stmt = null;
        throw sqlEx;
    }
}
}
```

```
public void closeConnection()throws SQLException, Exception {
    /**
     * Method closeConnection
     * Menutup hubungan dengan MySQL dan basis data
     */
    if (conn != null) {
        try {
            conn.close();
        }
        catch(SQLException sqlEx) {
            conn = null;
        }
    }
}
}
```

```
package model;
```

```
/*  
 * Filename : Telepon.java  
 * Programmer : Rosa Ariani Sukamto  
 * Date : 2009-01-13  
 * Email : rosa_if_itb_01@yahoo.com  
 * Website : http://www.gangsir.com  
 * Deskripsi : package model untuk mengakses tabel telepon  
 */  
*****/
```

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.sql.ResultSet;  
  
public class Telepon extends DB {  
  
    private String kode_anggota;//kode anggota  
    private String no_telepon;//nomor telepon  
  
    public Telepon() throws Exception, SQLException{  
        //konstruktor  
        super();  
    }  
  
    public void getTelepon(){  
        //mengeksekusi query untuk mengambil semua data pada tabel telepon  
        try{  
            String query = "SELECT * FROM telepon ";  
            createQuery(query);  
        }catch(Exception e){  
            //tampilkan kesalahan jika terjadi kesalahan  
            System.out.println(e.toString());  
        }  
    }  
}
```

```

package controller;

/*****
 * Filename : ProsesTelepon.java
 * Programmer : Rosa Ariani Sukamto
 * Date : 2009-01-13
 * Email : rosa_if_itb_01@yahoo.com
 * Website : http://www.gangsir.com
 * Deskripsi : package controller untuk Proses yang terkait dengan telepon
 *
 *****/

import java.io.*;
import model.DB;
import model.Telepon;

public class ProsesTelepon {

    private String hasil;//hasil mengambil data
    private String error;//error yang terjadi

    public ProsesTelepon(){
        //konstruktor
    }

    public void prosesTelp(){
        try{
            //mengambil data di tabel telepon dengan mengakses kelas model telepon
            Telepon telepon = new Telepon();
            telepon.getTelepon();
            hasil = "*****\n";

            while(telepon.getResult().next()){
                // ambil hasil query
                String kode_anggota = telepon.getResult().getString(1);
                String no_telepon = telepon.getResult().getString(2);
                //proses hasil
                hasil = hasil + "kode anggota: " + kode_anggota + "\n";
                hasil = hasil + "nomor telepon" + no_telepon + "\n";
                hasil = hasil + "*****\n";
            }
            //tutup koneksi
            telepon.closeResult();
            telepon.closeConnection();

        }catch(Exception e){
            //memproses error
            error = e.toString();
        }
    }

    public String getHasil(){
        //mengembalikan hasil
        return this.hasil;
    }
}

```

```
}
```

```
public String getError(){  
    //mengembalikan error  
    return this.error;  
}
```

```
}
```



```
package view;

/*****
 * Filename : TampilTelepon.java
 * Programmer : Rosa Ariani Sukamto
 * Date : 2009-01-13
 * Email : rosa_if_itb_01@yahoo.com
 * Website : http://www.gangsir.com
 * Deskripsi : package view untuk menampilkan tampilan
 *
 *****/

import controller.ProsesTelepon;

public class TampilTelepon {

    public TampilTelepon(){
        //konstruktor
    }

    public void tampil(){
        //menampilkan hasil tampilan
        ProsesTelepon ptelepon = new ProsesTelepon();
        try{
            ptelepon.prosesTelp();
            System.out.println(ptelepon.getHasil());
        }catch(Exception e){
            //menampilkan error
            System.out.println(ptelepon.getError());
        }
    }
}
```

```
import view.TampilTelepon;

/*****
 * Filename : Main.java
 * Programmer : Rosa Ariani Sukamto
 * Date : 2009-01-13
 * Email : rosa_if_itb_01@yahoo.com
 * Website : http://www.gangsir.com
 * Deskripsi : program utama
 *
 *****/

public class Main{
    public static void main(String[] args){
        //memanggil modul view
        TampilTelepon tampiltelepon = new TampilTelepon();
        tampiltelepon.tampil();
    }
}
```