

Modul : **IK-290 Desain dan Pemrograman Berorientasi Objek**
Topik : **Contoh Implementasi MVVM (*Model View ViewModel*) pada Java**
Dosen : **Rosa Ariani Sukamto**
Website : **<https://rosa-as.id>**
Blog : **<https://hariiniadalahhadiah.wordpress.com>**

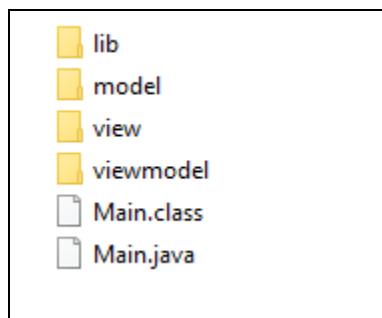
Yang diperlukan:

plugin:

mysql-connector-java-5.1.46-bin.jar dimasukkan ke dalam direktori lib pada penataan *file*

- kegunaan *plugin* biasanya untuk menghubungkan aplikasi diluar Java dengan program Java.

Susunan penataan *file*:



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

misal:

```
javac model\*.java
```

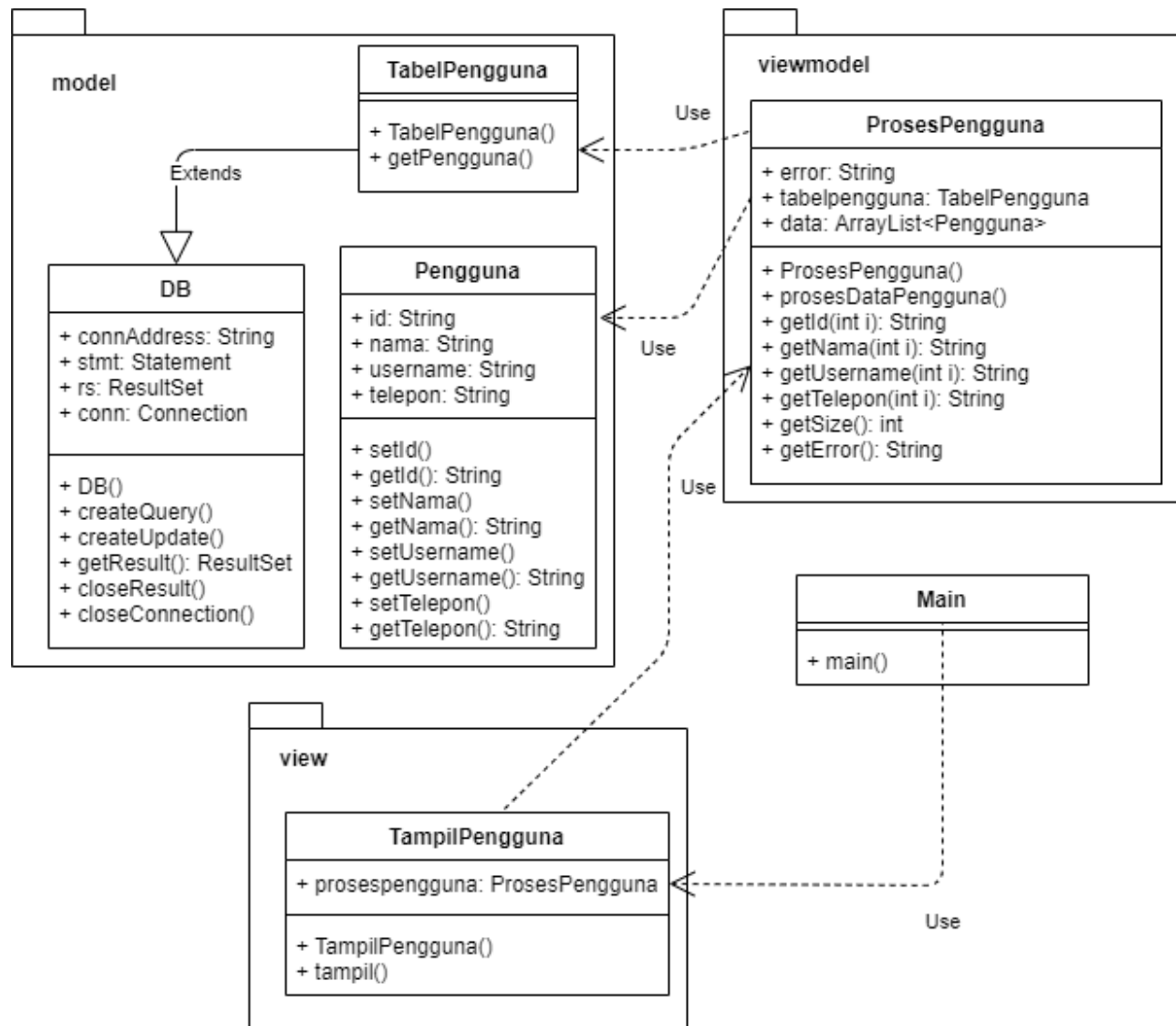
```
javac viewmodel\*.java
```

```
javac view\*.java
```

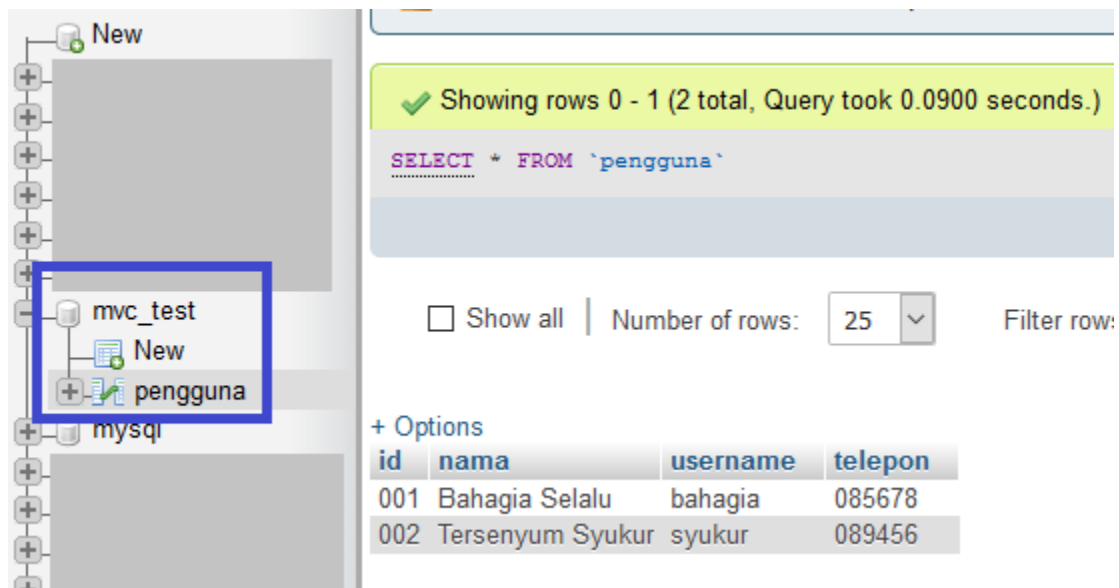
```
javac Main.java
```

```
java -cp lib/mysql-connector-java-5.1.46-bin.jar; Main
```

Diagram Kelas:



Basis Data dan Tabel MySQL:



The screenshot shows a MySQL database management tool interface. On the left, a tree view displays the database structure, with 'mvc_test' selected and 'pengguna' highlighted. The main area shows a query result for the 'pengguna' table. The query is `SELECT * FROM `pengguna``. The result shows 2 rows: (001, Bahagia Selalu, bahagia, 085678) and (002, Tersenyum Syukur, syukur, 089456). The interface also includes a status bar indicating 'Showing rows 0 - 1 (2 total, Query took 0.0900 seconds.)' and a 'Number of rows' dropdown set to 25.

id	nama	username	telepon
001	Bahagia Selalu	bahagia	085678
002	Tersenyum Syukur	syukur	089456

Hasil Eksekusi:

```
*****
id: 001
nama: Bahagia Selalu
username: bahagia
telepon: 085678
*****
id: 002
nama: Tersenyum Syukur
username: syukur
telepon: 089456
*****
```

Perhatikan bagaimana menulis komentar yang baik dalam sebuah kode program.

```

/*****
 * Filename : DB.java
 * Programmer : Rosa Ariani Sukanto
 * Date : 2018-03-30
 * Email : rosa_if_itb_01@yahoo.com
 * Website : https://rosa-as.id
 * 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/mvc_test?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 : TabelPengguna.java  
 * Programmer : Rosa Ariani Sukanto  
 * Date : 2020-05-10  
 * Email : rosa.ariani@upi.edu  
 * Website : https://rosa-as.id  
 * Deskripsi : package model untuk mengakses tabel pengguna  
 */
```

```
 */
```

```
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import java.sql.Statement;  
import java.sql.ResultSet;
```

```
public class TabelPengguna extends DB {
```

```
    public TabelPengguna() throws Exception, SQLException{  
        //konstruktor  
        super();  
    }
```

```
    public void getPengguna(){  
        //mengeksekusi query untuk mengambil semua data pada tabel pengguna  
        try{  
            String query = "SELECT * FROM pengguna ";  
            createQuery(query);  
        }catch(Exception e){  
            //tampilkan kesalahan jika terjadi kesalahan  
            System.out.println(e.toString());  
        }  
    }
```

```
}
```



```
package model;
```

```
/*  
*****  
*/
```

```
* Filename : Pengguna.java  
* Programmer : Rosa Ariani Sukanto  
* Date : 2020-05-10  
* Email : rosa.ariani@upi.edu  
* Website : https://rosa-as.id  
* Deskripsi : package model untuk menampung setiap data pengguna  
*  
*****/  
*/
```

```
public class Pengguna{
```

```
    private String id;//id pengguna  
    private String nama;//nama pengguna  
    private String username;//username pengguna  
    private String telepon;//nomor telepon pengguna
```

```
    public Pengguna(){  
        //konstruktor  
    }
```

```
    public void setId(String id){  
        //mengeset id pengguna  
        this.id = id;  
    }
```

```
    public String getId(){  
        //mengembalikan id pengguna  
        return this.id;  
    }
```

```
    public void setName(String nama){  
        //mengeset nama pengguna  
        this.nama = nama;  
    }
```



```
public String getNama(){
//mengembalikan nama pengguna
    return this.nama;
}

public void setUsername(String username){
//mengeset username pengguna
    this.username = username;
}

public String getUsername(){
//mengembalikan username pengguna
    return this.username;
}

public void setTelepon(String telepon){
//mengeset telepon pengguna
    this.telepon = telepon;
}

public String getTelepon(){
//mengembalikan telepon pengguna
    return this.telepon;
}
```

```

package viewmodel;

/*****
 * Filename : ProsesPengguna.java
 * Programmer : Rosa Ariani Sukamto
 * Date : 2020-05-10
 * Email : rosa.ariani@upi.edu
 * Website : https://rosa-as.id
 * Deskripsi : package viewmodel untuk penengah antara View dan Model
 *             tanpa terlibat hal-hal yang terkait format tampilan pada View
 *             dan tanpa adanya kontrak
 *****/

import java.util.ArrayList;
import model.TabelPengguna;
import model.Pengguna;

public class ProsesPengguna{

    private String error;//error yang terjadi
    private TabelPengguna tabelpengguna;//kelas untuk mengakses query tabel pengguna
    private ArrayList<Pengguna> data;//tempat menyimpan hasil query

    public ProsesPengguna(){
        //konstruktor
        try{
            tabelpengguna = new TabelPengguna();//instansiasi TabelPengguna
            data = new ArrayList<Pengguna>();//instansiasi list untuk data Pengguna
        }catch(Exception e){
            error = e.toString();
        }
    }
}

```

```

public void prosesDataPengguna(){
    try{
        //mengambil data di tabel pengguna
        tabelpengguna.getPengguna();
        while(tabelpengguna.getResult().next()){
            // ambil hasil query
            Pengguna pengguna = new Pengguna();//instansiasi objek pengguna untuk setiap data pengguna

            pengguna.setId(tabelpengguna.getResult().getString(1));//mengisi id
            pengguna.setNama(tabelpengguna.getResult().getString(2));// mengisi nama
            pengguna.setUsername(tabelpengguna.getResult().getString(3));//mengisi username
            pengguna.setTelepon(tabelpengguna.getResult().getString(4));//mengisi telepon

            data.add(pengguna);//tambahkan data pengguna ke dalam list
        }
        //tutup koneksi
        tabelpengguna.closeResult();
        tabelpengguna.closeConnection();
    }catch(Exception e){
        //memproses error
        error = e.toString();
    }
}

public String getId(int i){
    //mengembalikan id pengguna dengan indeks ke i
    return data.get(i).getId();
}

public String getNama(int i){
    //mengembalikan nama pengguna dengan indeks ke i
    return data.get(i).getNama();
}

```

```

public String getUsername(int i){
    //mengembalikan username pengguna dengan indeks ke i
    return data.get(i).getUsername();
}

public String getTelepon(int i){
    //mengembalikan telepon pengguna dengan indeks ke i
    return data.get(i).getTelepon();
}

public int getSize(){
    //mengembalikan banyaknya data pengguna yang masuk ke dalam list
    return data.size();
}

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

```

```

package view;

/*****
 * Filename : TampilPengguna.java
 * Programmer : Rosa Ariani Sukamto
 * Date : 2020-05-10
 * Email : rosa.ariani@upi.edu
 * Website : https://rosa-as.id
 * Deskripsi : package view untuk menampilkan tampilan
 *
 *****/

import viewmodel.ProsesPengguna;
import java.util.ArrayList;

public class TampilPengguna{

    private ProsesPengguna prosespengguna; // viewmodel yang dapat diakses view

    public TampilPengguna(){
        //konstruktor
        prosespengguna = new ProsesPengguna();
    }

```

```

        public void tampil(){
            //menampilkan tampilan
            try{
                prosespengguna.prosesDataPengguna();

                String hasil = "";
                //semua terkait tampilan adalah tanggung jawab View
                for(int i=0;i<prosespengguna.getSize();i++){
                    hasil = hasil + "id: " + prosespengguna.getId(i) + "\n";
                    hasil = hasil + "nama: " + prosespengguna.getNama(i) + "\n";
                    hasil = hasil + "username: " + prosespengguna.getUsername(i) + "\n";
                    hasil = hasil + "telepon: " + prosespengguna.getTelepon(i) + "\n";
                    hasil = hasil + "*****\n";
                }

                System.out.println("*****");
                System.out.println(hasil);

            }catch(Exception e){
                //menampilkan error
                System.out.println(prosespengguna.getError());
            }
        }
    }
}

```

```
import view.TampilPengguna;
```

```
/******
```

```
* Filename : Main.java
```

```
* Programmer : Rosa Ariani Sukanto
```

```
* Date : 2018-03-30
```

```
* Email : rosa_if_itb_01@yahoo.com
```

```
* Website : https://rosa-as.id
```

```
* Deskripsi : program utama
```

```
*
```

```
*****/
```

```
public class Main{
```

```
    public static void main(String[] args){
```

```
        //memanggil modul view
```

```
        TampilPengguna tampilpengguna = new TampilPengguna();
```

```
        tampilpengguna.tampil();
```

```
    }
```

```
}
```