

ACTIVITY PERTEMUAN 1

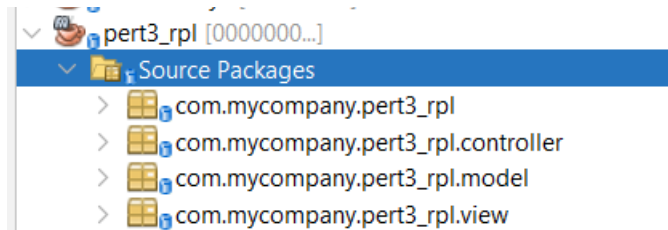
NAMA : Yogi Setiawan

NPM : 5142648

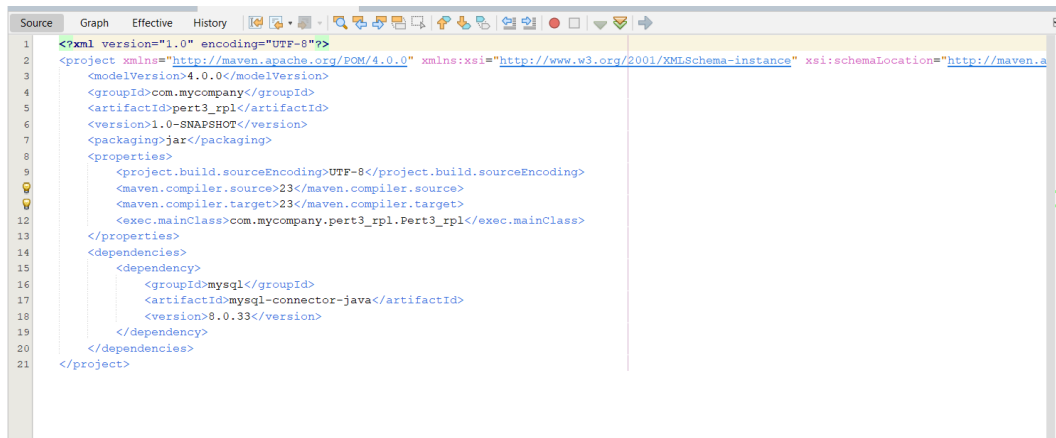
KELAS : 4IA17

MATERI : Konsep Model View Controller (MCV)

MATA PRAKTIKUM : RPL 2



Pom.xml



MahasiswaView

```
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.mycompany.pert3_rpl.view;
6
7  import com.mycompany.pert3_rpl.controller.MahasiswaController;
8  import com.mycompany.pert3_rpl.model.MahasiswaDAO;
9  import java.util.Scanner;
10
11  /**
12   *
13   * @author ASUS
14   */
15  public class MahasiswaView {
16      public static void main(String[] args){
17          MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
18          MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);
19
20          Scanner scanner = new Scanner(System.in);
21          int pilihan;
22
23          while(true){
24              System.out.println("Menu:");
25              System.out.println("1. Tampilkan Semua Mahasiswa");
26              System.out.println("2. Tambah Mahasiswa");
27              System.out.println("3. Update Mahasiswa");
28              System.out.println("4. Hapus Mahasiswa");
29              System.out.println("5. Cek Koneksi Database");
30              System.out.println("6. Keluar");
31
32              System.out.print("PILIH OPSI: ");
33              pilihan = scanner.nextInt();
34              scanner.nextLine();
35
36              switch (pilihan){
37                  case 1:
38                      mahasiswaController.displayAllMahasiswa();
39                      break;
40                  case 2:
41                      // tambah mhs
42                      System.out.println("Masukkan NPM: ");
43                      String npm = scanner.next();
44                      System.out.println("Masukkan Nama: ");
45                      String nama = scanner.next();
46                      System.out.println("Masukkan Semester: ");
47                      int semester = scanner.nextInt();
48                      System.out.println("Masukkan IPK: ");
49                      float ipk = scanner.nextFloat();
50                      System.out.println(npm + nama + semester + ipk);
51
52                      mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
53                      break;
54                  case 3:
55                      System.out.print("Masukkan ID mahasiswa: ");
56                      int id = scanner.nextInt();
57                      scanner.nextLine();
58
59                      System.out.println("Masukkan NPM: ");
60
61                      String npmBaru = scanner.next();
62                      System.out.println("Masukkan Nama: ");
63                      String namaBaru = scanner.next();
64                      System.out.println("Masukkan Semester: ");
65                      int semesterBaru = scanner.nextInt();
66                      System.out.println("Masukkan IPK: ");
67                      float ipkBaru = scanner.nextFloat();
68
69                      mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
70                      break;
71                  case 4:
72                      System.out.print("Masukkan ID Mahasiswa: ");
73                      int idHapus = scanner.nextInt();
74                      mahasiswaController.deleteMahasiswa(idHapus);
75                  case 5:
76                      mahasiswaController.checkDatabaseConnection();
77                      break;
78                  case 6:
79                      // Keluar
80                      mahasiswaController.closeConnection();
81                      System.out.println("Program selesai.");
82                      return;
83                  default:
84                      System.out.println("Input Tidak valid");
85
86              }
87          }
88      }
89  }
90
```

ModelMahasiswa

```
ModelMahasiswa.java [-/A] X
Source History
1
2 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4 */
5 package com.mycompany.pert3_rpl.model;
6
7 /**
8  *
9  * @author ASUS
10 */
11 public class ModelMahasiswa {
12     private int id;
13     private String npm;
14     private String nama;
15     private int semester;
16     private float ipk;
17
18     public int getId() {
19         return id;
20     }
21
22     public void setId(int id) {
23         this.id = id;
24     }
25
26     public String getNpm() {
27         return npm;
28     }
29
30     public void setNpm(String npm) {
31         this.npm = npm;
32     }
33
34     public String getName() {
35         return nama;
36     }
37
38     public void setName(String nama) {
39         this.nama = nama;
40     }
41
42     public int getSemester() {
43         return semester;
44     }
45
46     public void setSemester(int semester) {
47         this.semester = semester;
48     }
49
50     public float getIpk() {
51         return ipk;
52     }
53
54     public void setIpk(float ipk) {
55         this.ipk = ipk;
56     }
57
58     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
59         this.id = id;
60         this.npm = npm;
61         this.nama = nama;
62         this.semester = semester;
63         this.ipk = ipk;
64     }
65
66 }
```

MahasiswaDAO

```
Source History
MahasiswaDAO.java [~/A] X
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package com.mycompany.pert3_rpl.model;
6  import java.sql.Connection;
7  import java.sql.DriverManager;
8  import java.sql.PreparedStatement;
9  import java.sql.ResultSet;
10 import java.sql.SQLException;
11 import java.sql.Statement;
12 import java.util.ArrayList;
13 import java.util.List;
14
15 /**
16  *
17  * @author ASUS
18  */
19
20 public class MahasiswaDAO {
21     private Connection connection;
22
23     public MahasiswaDAO() {
24         try {
25             Class.forName("com.mysql.cj.jdbc.Driver");
26             connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mvc_db", "root", "");
27         } catch (Exception e) {
28             e.printStackTrace();
29         }
30     }
31
32     public boolean checkConnection() {
33         try {
34             if (connection != null && !connection.isClosed()) {
35                 return true; // koneksi berhasil
36             }
37         } catch (SQLException e) {
38             e.printStackTrace();
39         }
40         return false;
41     }
42
43     public void addMahasiswa(ModelMahasiswa mahasiswa) {
44         String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
45         try {
46             PreparedStatement pstmt = connection.prepareStatement(sql);
47             pstmt.setString(1, mahasiswa.getNpm());
48             pstmt.setString(2, mahasiswa.getNama());
49             pstmt.setInt(3, mahasiswa.getSemester());
50             pstmt.setFloat(4, mahasiswa.getIpk());
51             pstmt.executeUpdate();
52         } catch (SQLException e) {
53             e.printStackTrace();
54         }
55     }
56
57     public List<ModelMahasiswa> getAllMahasiswa() {
58         List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
59         String sql = "SELECT * FROM mahasiswa";
60         try {
61             Statement stmt = connection.createStatement();
62         }
63     }
64 }
```

```

61         ResultSet rs = stmt.executeQuery(sql);
62         while(rs.next()){
63             mahasiswaList.add(new ModelMahasiswa(
64                 rs.getInt("id"),
65                 rs.getString("npm"),
66                 rs.getString("nama"),
67                 rs.getInt("semester"),
68                 rs.getFloat("ipk")
69             ));
70         }
71     } catch (SQLException e) {
72         e.printStackTrace();
73     }
74     return mahasiswaList;
75 }
76
77 public void updateMahasiswa(ModelMahasiswa mahasiswa) {
78     String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
79     try {
80         PreparedStatement pstmt = connection.prepareStatement(sql);
81         pstmt.setString(1, mahasiswa.getNpm());
82         pstmt.setString(2, mahasiswa.getNama());
83         pstmt.setInt(3, mahasiswa.getSemester());
84         pstmt.setFloat(4, mahasiswa.getIpk());
85         pstmt.setInt(5, mahasiswa.getId());
86         pstmt.executeUpdate();
87     } catch (SQLException e) {
88         e.printStackTrace();
89     }
90 }
91
92 public void deleteMahasiswa(int id) {
93     String sql = "DELETE FROM mahasiswa WHERE id = ?";
94     try {
95         PreparedStatement pstmt = connection.prepareStatement(sql);
96         pstmt.setInt(1, id);
97         pstmt.executeUpdate();
98     } catch (SQLException e) {
99         e.printStackTrace();
100     }
101 }
102
103 // Method untuk menutup koneksi database
104 public void closeConnection() {
105     try {
106         if (connection != null) {
107             connection.close();
108         }
109     } catch (SQLException e) {
110         e.printStackTrace();
111     }
112 }
113
114 }
115

```

Mahasiswa Controller

```
Source History
MahasiswaController.java [-A] X
2  /*
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
4   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
5   */
6
7   package com.mycompany.pert3_rpl.controller;
8
9   import com.mycompany.pert3_rpl.model.MahasiswaDAO;
10  import com.mycompany.pert3_rpl.model.ModelMahasiswa;
11  import java.util.List;
12
13  /**
14   *
15   * @author ASUS
16   */
17  public class MahasiswaController {
18      private MahasiswaDAO mahasiswaDAO;
19
20      public MahasiswaController(MahasiswaDAO mahasiswaDAO) {
21          this.mahasiswaDAO = mahasiswaDAO;
22      }
23  }
```

```
Source History
MahasiswaController.java [-A] X
23
24  if(mahasiswaList.isEmpty()){
25      System.out.println("Tidak ada data mahasiswa");
26  } else {
27      System.out.println("");
28      System.out.println("=====");
29      for(ModelMahasiswa m: mahasiswaList){
30          System.out.println("ID          : " + m.getId());
31          System.out.println("NPM       : " + m.getNpm());
32          System.out.println("NAMA      : " + m.getName());
33          System.out.println("SEMESTER  : " + m.getSemester());
34          System.out.println("IPK       : " + m.getIpk());
35          System.out.println("=====");
36      }
37  }
38
39
40  public void displayMessage(String message){
41      System.out.println(message);
42  }
43
44
45
46  public void checkDatabaseConnection(){
47      boolean isConnected = mahasiswaDAO.checkConnection();
48      if (isConnected){
49          displayMessage("Koneksi ke db berhasil");
50      } else{
51          displayMessage("Koneksi DB Gagal");
52      }
53  }
54  }
```

```
55  // READ ALL (Menampilkan semua mahasiswa)
56  public void displayAllMahasiswa(){
57      List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
58      displayMahasiswaList(mahasiswaList);
59  }
60
61  public void addMahasiswa(String npm, String nama, int semester, float ipk){
62      ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
63      System.out.println("Controller Data: " + npm + nama + semester + ipk);
64      System.out.println(mahasiswaBaru);
65      mahasiswaDAO.addMahasiswa(mahasiswaBaru);
66      displayMessage("Mahasiswa berhasil ditambahkan!");
67  }
68
69  public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk){
70      ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
71      mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
72      displayMessage("Mahasiswa berhasil diperbarui!");
73  }
74
75  public void deleteMahasiswa(int id){
76      mahasiswaDAO.deleteMahasiswa(id);
77      displayMessage("Mahasiswa Berhasil Dihapus!");
78  }
79
80  public void closeConnection() {
81      mahasiswaDAO.closeConnection();
82  }
83
84  }
85  }
```

```
pom.xml [-A] [pert3_rpl] x
Source Graph Effective History
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.a
3 <modelVersion>4.0.0</modelVersion>
4 <groupId>com.mycompany</groupId>
5 <artifactId>pert3_rpl</artifactId>
6 <version>1.0-SNAPSHOT</version>
7 <packaging>jar</packaging>
8 <properties>
9 <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
10 <maven.compiler.source>23</maven.compiler.source>
11 <maven.compiler.target>23</maven.compiler.target>
12 <exec.mainClass>com.mycompany.pert3_rpl.Pert3_rpl</exec.mainClass>
13 </properties>
14 <dependencies>
15 <dependency>
16 <groupId>mysql</groupId>
17 <artifactId>mysql-connector-java</artifactId>
18 <version>8.0.33</version>
19 </dependency>
20 </dependencies>
21 </project>
```

Output xml dan view

```
Output x
Run (MahasiswaView) x Build (pert3_rpl) x
Changes detected - recompiling the module! :source
Compiling 5 source files with javac [debug target 23] to target\classes
--- resources:3.3.1:testResources (default-testResources) @ pert3_rpl ---
skip non existing resourceDirectory C:\Users\ASUS\Documents\31a17\pert3_rpl\src\test\resources
--- compiler:3.11.0:testCompile (default-testCompile) @ pert3_rpl ---
Changes detected - recompiling the module! :dependency
--- surefire:3.2.2:test (default-test) @ pert3_rpl ---
--- jar:3.3.0:jar (default-jar) @ pert3_rpl ---
Building jar: C:\Users\ASUS\Documents\31a17\pert3_rpl\target\pert3_rpl-1.0-SNAPSHOT.jar
--- install:3.1.1:install (default-install) @ pert3_rpl ---
Installing C:\Users\ASUS\Documents\31a17\pert3_rpl\pom.xml to C:\Users\ASUS\.m2\repository\com\mycompany\pert3_rpl\1.0-SNAPSHOT\pert3_rpl-1.0-SNAPSHOT.pom
Installing C:\Users\ASUS\Documents\31a17\pert3_rpl\target\pert3_rpl-1.0-SNAPSHOT.jar to C:\Users\ASUS\.m2\repository\com\mycompany\pert3_rpl\1.0-SNAPSHOT\pert3_rpl-1.0-SNAPSHOT.jar
BUILD SUCCESS
-----
Total time: 4.215 s
Finished at: 2025-11-01T15:03:13+07:00
-----

cd C:\Users\ASUS\Documents\31a17\pert3_rpl; "JAVA_HOME=C:\Program Files\Java\jdk-23" cmd /c "%C:\Program Files\NetBeans-21\netbeans\java\maven\bin\
Scanning for projects...

-----< com.mycompany:cert3_rpl >-----
Building cert3_rpl 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----
The artifact mysql:mysql-connector-java:jar:8.0.33 has been relocated to com.mysql:mysql-connector-j:jar:8.0.33: MySQL Connector/J artifacts moved to reverse-
--- resources:3.3.1:resources (default-resources) @ pert3_rpl ---
skip non existing resourceDirectory C:\Users\ASUS\Documents\31a17\pert3_rpl\src\main\resources
--- compiler:3.11.0:compile (default-compile) @ pert3_rpl ---
Changes detected - recompiling the module! :source
Compiling 5 source files with javac [debug target 23] to target\classes
--- exec:3.1.0:exec (default-cli) @ pert3_rpl ---
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
```

Phpmyadmin

The screenshot displays the phpMyAdmin web interface. The left sidebar shows a database structure with 'mvc_db' selected. The main panel shows the 'Table structure' view for the 'mahasiswa' table. The table has five columns: 'id' (int(11), AUTO_INCREMENT), 'nama' (varchar(60), utf8mb4_general_ci), 'npm' (varchar(15), utf8mb4_general_ci), 'semester' (int(2)), and 'ipk' (float). Below the table structure, there are options to add, drop, or change columns, and a section for indexes. The 'Indexes' section shows a primary index on the 'id' column.

| # | Name | Type | Collation | Attributes | Null | Default | Comments | Extra | Action |
|---|----------|-------------|--------------------|------------|------|---------|----------|----------------|------------------|
| 1 | id | int(11) | | | No | None | | AUTO_INCREMENT | Change Drop More |
| 2 | nama | varchar(60) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 3 | npm | varchar(15) | utf8mb4_general_ci | | No | None | | | Change Drop More |
| 4 | semester | int(2) | | | No | None | | | Change Drop More |
| 5 | ipk | float | | | No | None | | | Change Drop More |

Indexes

| Action | Keyname | Type | Unique | Packed | Column | Cardinality | Collation | Null | Comment |
|------------------|---------|-------|--------|--------|--------|-------------|-----------|------|---------|
| Edit Rename Drop | PRIMARY | BTREE | Yes | No | id | 0 | A | No | |

Create an index on 1 columns Go