

Nama : Muhammad Izzuddin Almansyur
NPM : 51422084
Kelas : 4IA07
Praktikum : Rekayasa Perangkat Lunak 2, ACT3

1. SS kode dan output

MahasiswaController.java

```
5  package com.mycompany.rpl2_pert3.controller;
6
7  import com.mycompany.rpl2_pert3.model.MahasiswaDAO;
8  import com.mycompany.rpl2_pert3.model.ModelMahasiswa;
9  import java.util.List;
10
11 /**
12 * 
13 * @author Izza
14 */
15 public class MahasiswaController {
16     private MahasiswaDAO mahasiswaDAO;
17
18     public MahasiswaController(MahasiswaDAO mahasiswaDAO){
19         this.mahasiswaDAO = mahasiswaDAO;
20     }
21
22     public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList){
23         if(mahasiswaList.isEmpty()){
24             System.out.println("Tidak ada data mahasiswa");
25         } else {
26             System.out.print("");
27             System.out.println("=====");
28             for(ModelMahasiswa m: mahasiswaList){
29                 System.out.println("ID      : " + m.getId());
30                 System.out.println("NPM    : " + m.getNpm());
31                 System.out.println("NAMA   : " + m.getName());
32                 System.out.println("SEMESTER : " + m.getSemester());
33                 System.out.println("IPK     : " + m.getIpk());
34                 System.out.println("=====");
35             }
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
```

MahasiswaDAO.java

```
5  package com.mycompany.rpl2_pert3.model;
6
7  import java.sql.Connection;
8  import java.sql.DriverManager;
9  import java.sql.PreparedStatement;
10 import java.sql.ResultSet;
11 import java.sql.SQLException;
12 import java.sql.Statement;
13 import java.util.ArrayList;
14 import java.util.List;
15
16
17 /**
18  *
19  * @author Izza
20  */
21 public class MahasiswaDAO {
22     private Connection connection;
23
24     public MahasiswaDAO(){
25         try{
26             Class.forName("com.mysql.cj.jdbc.Driver");
27             connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/pert3_rpl2", "root", "");
28         } catch (Exception e){
29             e.printStackTrace();
30         }
31     }
32
33     public boolean checkConnection(){
34         try{
35             if(connection != null && !connection.isClosed()){
36                 return true; //koneksi berhasil
37             }
38         } catch (SQLException e){
39             e.printStackTrace();
40         }
41         return false;
42     }
43
44     public void addMahasiswa(ModelMahasiswa mahasiswa){
45         String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
46         try{
47             PreparedStatement pstmt = connection.prepareStatement(sql);
48             pstmt.setString(1, mahasiswa.getNpm());
49             pstmt.setString(2, mahasiswa.getNama());
50             pstmt.setInt(3, mahasiswa.getSemester());
51             pstmt.setFloat(4, mahasiswa.getIpk());
52             pstmt.executeUpdate();
53         } catch (SQLException e){
54             e.printStackTrace();
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             ResultSet rs = stmt.executeQuery(sql);
63             while(rs.next()){
64                 mahasiswaList.add(new ModelMahasiswa(
65                     rs.getInt("id"),
66                     rs.getString("npm"),
67                     rs.getString("nama"),
68                     rs.getInt("semester"),
69                     rs.getFloat("ipk")
70                 )
71             );
72         }
73     }
74 }
```

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

ModelMahasiswa.java

```
5  package com.mycompany.rpl2_pert3.model;
6
7  /**
8   * 
9   * @author Izza
10  */
11 public class ModelMahasiswa {
12
13     public int getId() {
14         return id;
15     }
16
17     public void setId(int id) {
18         this.id = id;
19     }
20
21     public String getNpm() {
22         return npm;
23     }
24
25     public void setNpm(String npm) {
26         this.npm = npm;
27     }
28
29     public String getNama() {
30         return nama;
31     }
32
33     public void setNama(String nama) {
34         this.nama = nama;
35     }
36
37     public int getSemester() {
38         return semester;
39     }
40
41     public void setSemester(int semester) {
42         this.semester = semester;
43     }
44
45     public float getIpk() {
46         return ipk;
47     }
48
49     public void setIpk(float ipk) {
50         this.ipk = ipk;
51     }
52     private int id;
53     private String npm;
54     private String nama;
55     private int semester;
56     private float ipk;
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 }
```

MahasiswaView.java

```
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              System.out.print("PILIH OPSI: ");
32              pilihan = scanner.nextInt();
33              scanner.nextLine();
34
35              switch (pilihan){
36                  case 1:
37                      mahasiswaController.displayAllMahasiswa();
38                      break;
39
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
55                  case 3:
56                      System.out.print("Masukkan ID mahasiswa: ");
57                      int id = scanner.nextInt();
58                      scanner.nextLine();
59
60                      System.out.println("Masukkan NPM: ");
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:
```

Pom.xml

```
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.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
3      <modelVersion>4.0.0</modelVersion>
4      <groupId>com.mycompany</groupId>
5      <artifactId>rpl2_pert3</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.release>24</maven.compiler.release>
11     <exec.mainClass>com.mycompany.rpl2_pert3.Rpl2_pert3</exec.mainClass>
12   </properties>
13   <dependencies>
14     <dependency>
15         <groupId>mysql</groupId>
16         <artifactId>mysql-connector-java</artifactId>
17         <version>8.0.33</version>
18     </dependency>
19   </dependencies>
20 </project>
```

Output

```
Output - Run (MahasiswaView) ×
▶ Menu:
▶   1. Tampilkan Semua Mahasiswa
▶   2. Tambah Mahasiswa
▶   3. Update Mahasiswa
▶   4. Hapus Mahasiswa
▶   5. Cek Koneksi Database
▶   6. Keluar
▶ PILIH OPSI: 5
Koneksi ke db berhasil
Menu:
  1. Tampilkan Semua Mahasiswa
  2. Tambah Mahasiswa
  3. Update Mahasiswa
  4. Hapus Mahasiswa
  5. Cek Koneksi Database
  6. Keluar
PILIH OPSI: 2
Masukkan NPM:
51422084
Masukkan Nama:
Izzuddin
Masukkan Semester:
7
Masukkan IPK:
3,85
51422084Izzuddin73.85
Controller Data: 51422084Izzuddin73.85
com.mycompany.rpl2_pert3.model.ModelMahasiswa@10bbd20a
Mahasiswa berhasil ditambahkan!
Menu:
  1. Tampilkan Semua Mahasiswa
  2. Tambah Mahasiswa
  3. Update Mahasiswa
  4. Hapus Mahasiswa
  5. Cek Koneksi Database
  6. Keluar
```

```
Menu:  
1. Tampilkan Semua Mahasiswa  
2. Tambah Mahasiswa  
3. Update Mahasiswa  
4. Hapus Mahasiswa  
5. Cek Koneksi Database  
6. Keluar  
PILIH OPSI: 1
```

```
=====  
ID      : 2  
NPM     : 51422084  
NAMA    : Izzuddin  
SEMESTER : 7  
IPK     : 3.85  
=====
```

```
Menu:  
1. Tampilkan Semua Mahasiswa  
2. Tambah Mahasiswa  
3. Update Mahasiswa  
4. Hapus Mahasiswa  
5. Cek Koneksi Database  
6. Keluar  
PILIH OPSI: 6  
Program selesai.
```

```
BUILD SUCCESS
```

```
Total time: 18:25 min  
Finished at: 2025-11-01T13:24:29+07:00
```

Database

The screenshot shows the phpMyAdmin interface for a MySQL database named 'pert3_rpl2'. The current table is 'mahasiswa'. The top navigation bar includes links for Server (127.0.0.1), Database (pert3_rpl2), and Table (mahasiswa). Below the navigation are tabs for Browse, Structure, SQL, Search, Insert, Export, Import, and Help. A success message 'Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)' is displayed in a green bar. The SQL query 'SELECT * FROM `mahasiswa`' is shown in the query editor. Below the query are buttons for Profiling, Edit inline, Edit, Explain SQL, Create PHP code, and Refresh. There are also buttons for Show all, Number of rows (set to 25), Filter rows, and a search input field. The main content area displays the 'mahasiswa' table with one row: id 2, nama Izzuddin, npm 51422084, semester 7, and ipk 3.85. Below the table are buttons for Edit, Copy, Delete, Check all, With selected, and Export. At the bottom are additional buttons for Show all, Number of rows (set to 25), Filter rows, and a search input field.

	id	nama	npm	semester	ipk
<input type="checkbox"/>	2	Izzuddin	51422084	7	3.85