

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.println("");
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.getNama());
32                 System.out.println("SEMESTER     : " + m.getSemester());
33                 System.out.println("IPK          : " + m.getIpk());
34                 System.out.println("=====");
35             }
36         }
37     }
38
39     public void displayMessage(String message) {
40         System.out.println(message);
41     }
42
43     public void checkDatabaseConnection() {
44         boolean isConnected = mahasiswaDAO.checkConnection();
45         if (isConnected) {
46             displayMessage("Koneksi ke db berhasil");
47         } else {
48             displayMessage("Koneksi DB Gagal");
49         }
50     }
51
52     // READ ALL (Menampilkan semua mahasiswa)
53     public void displayAllMahasiswa() {
54         List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
55         displayMahasiswaList(mahasiswaList);
56     }
57
58     public void addMahasiswa(String npm, String nama, int semester, float ipk) {
59         ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
60         System.out.println("Controller Data: " + npm + nama + semester + ipk);
61         System.out.println(mahasiswaBaru);
62         mahasiswaDAO.addMahasiswa(mahasiswaBaru);
63         displayMessage("Mahasiswa berhasil ditambahkan!");
64     }
65
66     public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk) {
67         ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
68         mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
69         displayMessage("Mahasiswa berhasil diperbarui!");
70     }
71
72 }
```

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
58     public List<ModelMahasiswa> getAllMahasiswa() {
59         List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
60         String sql = "SELECT * FROM mahasiswa";
61         try {
62             Statement stmt = connection.createStatement();
63             ResultSet rs = stmt.executeQuery(sql);
64             while (rs.next()) {
65                 mahasiswaList.add(new ModelMahasiswa(
66                     rs.getInt("id"),
67                     rs.getString("npm"),
68                     rs.getString("nama"),
69                     rs.getInt("semester"),
70                     rs.getFloat("ipk")
71                 ));
72             }
73         } catch (SQLException e) {
74             e.printStackTrace();
75         }
76     }
77 }
```

```

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 getName() {
30         return nama;
31     }
32
33     public void setName(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 }
67
```

MahasiswaView.java

```

16 public class MahasiswaView {
17     public static void main(String[] args){
18         MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
19         MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);
20
21         Scanner scanner = new Scanner(System.in);
22         int pilihan;
23
24         while(true){
25             System.out.println("Menu:");
26             System.out.println("1. Tampilkan Semua Mahasiswa");
27             System.out.println("2. Tambah Mahasiswa");
28             System.out.println("3. Update Mahasiswa");
29             System.out.println("4. Hapus Mahasiswa");
30             System.out.println("5. Cek Koneksi Database");
31             System.out.println("6. Keluar");
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
41                 case 2:
42                     // tambah mhs
43                     System.out.println("Masukkan NPM: ");
44                     String npm = scanner.next();
45                     System.out.println("Masukkan Nama: ");
46                     String nama = scanner.next();
47                     System.out.println("Masukkan Semester: ");
48                     int semester = scanner.nextInt();
49                     System.out.println("Masukkan IPK: ");
50                     float ipk = scanner.nextFloat();
51                     System.out.println(npm + nama + semester + ipk);
52
53                     mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
54                     break;
55
56                 case 3:
57                     System.out.print("Masukkan ID mahasiswa: ");
58                     int id = scanner.nextInt();
59                     scanner.nextLine();
60
61                     System.out.println("Masukkan NPM: ");
62                     String npmBaru = scanner.next();
63                     System.out.println("Masukkan Nama: ");
64                     String namaBaru = scanner.next();
65                     System.out.println("Masukkan Semester: ");
66                     int semesterBaru = scanner.nextInt();
67                     System.out.println("Masukkan IPK: ");
68                     float ipkBaru = scanner.nextFloat();
69
70                     mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
71                     break;
72                 case 4:
73                     System.out.print("Masukkan ID Mahasiswa: ");
74                     int idHapus = scanner.nextInt();
75                     mahasiswaController.deleteMahasiswa(idHapus);
76                 case 5:
77                     mahasiswaController.checkDatabaseConnection();
78                     break;
79                 case 6:
80                     // Keluar
81                     mahasiswaController.closeConnection();
82                     System.out.println("Program selesai.");
83                     return;
84             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/maven-v4_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) x
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

Server: 127.0.0.1 » Database: pert3_rpl2 » Table: mahasiswa

Browse

Structure

SQL

Search

Insert

Export

Import

✓ Showing rows 0 - 0 (1 total, Query took 0.0002 seconds.)

SELECT * FROM `mahasiswa`

☐ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 ▾ Filter rows:

Extra options

↔ T ↔

▼ id nama npm semester ipk

☐ Edit Copy Delete 2 Izzuddin 51422084 7 3.85

⬆

☐ Check all With selected: Edit Copy Delete Export

☐ Show all | Number of rows: 25 ▾ Filter rows: