

ACTIVITY PERTEMUAN 3

NAMA : Fizri Rosdiansyah

NPM : 50421535

KELAS : 4IA06

MATERI : Konsep MVC

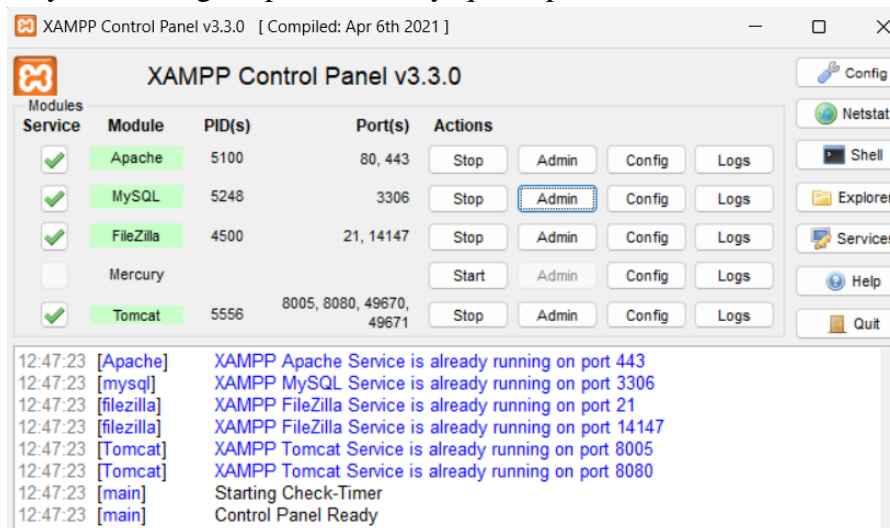
MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

Soal

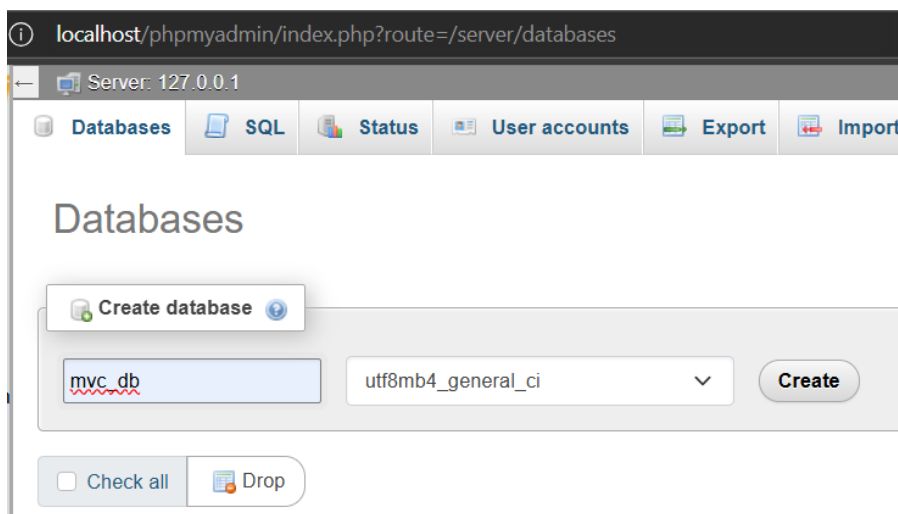
1. Jelaskan bagaimana cara membuat database “mvc_db” menggunakan XAAMP?
2. Screenshot hasil codingan kalian lengkap dengan windows projects,navigator, dan outputnya?

Jawab

- 1.nyalakan fungsi Apache dan Mysql dan pencet admin



Create Database



Lalu Buat masukan kodingan berikut

```
1 CREATE TABLE `mahasiswa` (  
2   `id` int(11) NOT NULL,  
3   `ipk` float NOT NULL,  
4   `nama` varchar(55) DEFAULT NULL,  
5   `npm` varchar(10) DEFAULT NULL,  
6   `semester` int(3) NOT NULL  
7 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;  
8  
9 ALTER TABLE `mahasiswa`  
10  ADD PRIMARY KEY (`id`);  
11  
12 ALTER TABLE `mahasiswa`  
13  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
```

Setelah itu pencet go

Tabel Mahasiswa sudah dibuat

Server: 127.0.0.1 » Database: mvc_db » Table: mahasiswa

Browse

Structure

SQL

Search

Insert

Export

Import

Privileges

Operations

Table structure

Relation view

	#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2	ipk	float			No	None			Change Drop More
<input type="checkbox"/>	3	nama	varchar(55)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	4	npm	varchar(10)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5	semester	int(3)			No	None			Change Drop More

☐ Check all

With selected:

Browse

Change

Drop

Primary

Unique

Index

Spatial

Fulltext

Add to central columns

Remove from central columns

2.

MahasiswaController

The screenshot displays the `MahasiswaController.java` file in an IDE. The code is as follows:

```

1 package com.mahasiswa.controller;
2
3 import java.sql.*;
4 import com.mahasiswa.model.MahasiswaDAO;
5 import com.mahasiswa.model.ModelMahasiswa;
6 import java.util.List;
7
8 public class MahasiswaController {
9     private MahasiswaDAO mahasiswaDAO;
10
11     public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
12         if (mahasiswaList.isEmpty()) {
13             System.out.println("Tidak ada data mahasiswa");
14         } else {
15             System.out.println("");
16             System.out.println("=====");
17             for (ModelMahasiswa m : mahasiswaList) {
18                 System.out.println("ID          : " + m.getId());
19                 System.out.println("NPM         : " + m.getNpm());
20                 System.out.println("NAMA        : " + m.getNama());
21                 System.out.println("SEMESTER    : " + m.getSemester());
22                 System.out.println("IPK         : " + m.getIpk());
23                 System.out.println("=====");
24             }
25             displayMessage("Mahasiswa berhasil ditampilkan");
26         }
27     }
28
29     public void displayMessage(String message) {
30         System.out.println(message);
31     }
32
33     public MahasiswaController(MahasiswaDAO mahasiswaDAO) {
34         this.mahasiswaDAO = mahasiswaDAO;
35     }
36
37     public void checkDatabaseConnection() {
38         boolean isConnected = mahasiswaDAO.cekKoneksi();
39         if (isConnected) {
40             displayMessage("Koneksi ke db berhasil");
41         } else {
42             displayMessage("Koneksi DB gagal");
43         }
44     }
45
46     public void displayAllMahasiswa() {
47         List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
48         displayMahasiswaList(mahasiswaList);
49     }
50
51     public void addMahasiswa(String npm, String nama, int semester, float ipk) {
52         ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
53     }
54 }

```

The output window shows the following results:

```

4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
1
=====
ID          : 2
NPM         : 50421535
NAMA        : fizri
SEMESTER    : 7
=====
Mahasiswa berhasil ditampilkan

```

Projects

Services

RPL2_Pert1

RPL2_Pert2_50421535

Source Packages

rpl2_pert2_50421535

Mahasiswa.java

RPL2_Pert2_50421535.java

Student.java

Test Packages

Libraries

Test Libraries

RPL2_Pert3_50421535

Source Packages

com.mahasiswa.controller

MahasiswaController.java

com.mahasiswa.model

MahasiswaDAO.java

ModelMahasiswa.java

com.mahasiswa.view

MahasiswaController - Navigator

Members

<empty>

MahasiswaController

MahasiswaController(MahasiswaDAO mahasiswaDAO)

addMahasiswa(String npm, String nama, int semester, float ipk)

checkDatabaseConnection()

closeConnection()

MahasiswaController.java

Source

History

```

48     displayMahasiswaList(mahasiswaList);
49
50
51
52     public void addMahasiswa(String npm, String nama, int semester, float ipk){
53         ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
54         mahasiswaDAO.addMahasiswa(mahasiswaBaru);
55         displayMessage("Mahasiswa berhasil ditambahkan");
56     }
57
58     public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk){
59         ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
60         mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
61         displayMessage("Mahasiswa berhasil diperbaharui");
62     }
63
64     public void deleteMahasiswa(int id){
65         mahasiswaDAO.deleteMahasiswa(id);
66         displayMessage("Mahasiswa Berhasil Dihapus!");
67     }
68
69     public void closeConnection(){
70         mahasiswaDAO.closeConnection();
71     }

```

Output - Run (RPL2_Pert3_50421535)

4. Hapus Mahasiswa

5. Cek Koneksi Database

6. Keluar

PILIH OPSI:

1

=====

ID

:

2

NPM

:

50421535

NAMA

:

fizri

SEMESTER

:

7

MahasiswaDAO

Projects

Services

RPL2_Pert1

RPL2_Pert2_50421535

Source Packages

rpl2_pert2_50421535

Mahasiswa.java

RPL2_Pert2_50421535.java

Student.java

Test Packages

Libraries

Test Libraries

RPL2_Pert3_50421535

Source Packages

com.mahasiswa.controller

MahasiswaController.java

com.mahasiswa.model

MahasiswaDAO.java

ModelMahasiswa.java

com.mahasiswa.view

MahasiswaDAO.java - Navigator

Members

<empty>

MahasiswaDAO

MahasiswaDAO()

addMahasiswa(ModelMahasiswa mahasiswa)

cekKoneksi(): boolean

closeConnection()

MahasiswaController.java

MahasiswaDAO.java

Source

History

```

1 package com.mahasiswa.model;
2
3 import java.sql.*;
4 import java.util.ArrayList;
5 import java.util.List;
6
7 public class MahasiswaDAO {
8     private Connection connection;
9
10    public MahasiswaDAO(){
11        try{
12            Class.forName("com.mysql.cj.jdbc.Driver");
13            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mvc_db", "root", "");
14        }catch (Exception e){
15            e.printStackTrace();
16        }
17    }
18
19    public boolean cekKoneksi(){
20        try{
21            if(connection != null && !connection.isClosed()){
22                return true;
23            }
24        }catch (SQLException e){
25            e.printStackTrace();
26        }
27    }

```

Output - Run (RPL2_Pert3_50421535)

4. Hapus Mahasiswa

5. Cek Koneksi Database

6. Keluar

PILIH OPSI:

1

=====

ID

:

2

NPM

:

50421535

NAMA

:

fizri

SEMESTER

:

7

Projects | Services

- RPL2_Pert1
 - RPL2_Pert2_50421535
 - Source Packages
 - rpl2_pert2_50421535
 - Mahasiswa.java
 - RPL2_Pert2_50421535.java
 - Student.java
 - Test Packages
 - Libraries
 - Test Libraries
 - RPL2_Pert3_50421535
 - Source Packages
 - com.mahasiswa.controller
 - MahasiswaController.java
 - com.mahasiswa.model
 - ModelMahasiswa.java
 - com.mahasiswa.view
 - MahasiswaDAO.java

MahasiswaDAO.java - Navigator

Members

- MahasiswaDAO
 - MahasiswaDAO()
 - addMahasiswa(ModelMahasiswa mahasiswa)
 - cekKoneksi(): boolean
 - closeConnection()

Source | History

```

19 public boolean cekKoneksi() {
20     try {
21         if (connection != null && !connection.isClosed()) {
22             return true;
23         }
24     } catch (SQLException e) {
25         e.printStackTrace();
26     }
27     return false;
28 }
29
30 public void addMahasiswa(ModelMahasiswa mahasiswa) {
31     String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
32     try {
33         PreparedStatement pstmt = connection.prepareStatement(sql);
34         pstmt.setString(1, mahasiswa.getNpm());
35         pstmt.setString(2, mahasiswa.getNama());
36         pstmt.setInt(3, mahasiswa.getSemester());
37         pstmt.setFloat(4, mahasiswa.getIpk());
38         pstmt.executeUpdate();
39     } catch (SQLException e) {
40         e.printStackTrace();
41     }
42 }
43

```

Output - Run (RPL2_Pert3_50421535)

```

4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
1

=====
ID          : 2
NPM         : 50421535
NAMA        : fizri
SEMESTER    : 7

```

Projects | Services

- RPL2_Pert1
 - RPL2_Pert2_50421535
 - Source Packages
 - rpl2_pert2_50421535
 - Mahasiswa.java
 - RPL2_Pert2_50421535.java
 - Student.java
 - Test Packages
 - Libraries
 - Test Libraries
 - RPL2_Pert3_50421535
 - Source Packages
 - com.mahasiswa.controller
 - MahasiswaController.java
 - com.mahasiswa.model
 - ModelMahasiswa.java
 - com.mahasiswa.view
 - MahasiswaDAO.java

MahasiswaDAO.java - Navigator

Members

 - MahasiswaDAO
 - MahasiswaDAO()
 - addMahasiswa(ModelMahasiswa mahasiswa)
 - cekKoneksi(): boolean
 - closeConnection()

Source | History

```

43 public List<ModelMahasiswa> getAllMahasiswa() {
44     List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
45     String sql = "SELECT * FROM mahasiswa";
46     try {
47         Statement stmt = connection.createStatement();
48         ResultSet rs = stmt.executeQuery(sql);
49         while (rs.next()) {
50             mahasiswaList.add(new ModelMahasiswa(
51                 rs.getInt("id"),
52                 rs.getString("npm"),
53                 rs.getString("nama"),
54                 rs.getInt("semester"),
55                 rs.getFloat("ipk")
56             ));
57         }
58     } catch (SQLException e) {
59         e.printStackTrace();
60     }
61     return mahasiswaList;
62 }
63
64 public void updateMahasiswa(ModelMahasiswa mahasiswa) {
65     String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
66     try {
67

```

Output - Run (RPL2_Pert3_50421535)

```

4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
1

=====
ID          : 2
NPM         : 50421535
NAMA        : fizri
SEMESTER    : 7

```

Projects x Services

- RPL2_Pert1
- RPL2_Pert2_50421535
 - Source Packages
 - rpl2_pert2_50421535
 - Mahasiswa.java
 - RPL2_Pert2_50421535.java
 - Student.java
 - Test Packages
 - Libraries
 - Test Libraries
 - RPL2_Pert3_50421535
 - Source Packages
 - com.mahasiswa.controller
 - MahasiswaController.java
 - com.mahasiswa.model
 - MahasiswaDAO.java
 - ModelMahasiswa.java
 - com.mahasiswa.view

MahasiswaDAO.java - Navigator x

Members

 - MahasiswaDAO
 - MahasiswaDAO()
 - addMahasiswa(ModelMahasiswa mahasiswa)
 - cekKoneksi(): boolean
 - closeConnection()

Output - Run (RPL2_Pert3_50421535)

```
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
1
=====
ID          : 2
NPM         : 50421535
NAMA        : fizrri
SEMESTER    : 7
```

MahasiswaDAO.java x

```
64
65 public void updateMahasiswa(ModelMahasiswa mahasiswa){
66     String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
67     try{
68         PreparedStatement pstmt = connection.prepareStatement(sql);
69         pstmt.setString(1, mahasiswa.getNpm());
70         pstmt.setString(2, mahasiswa.getNama());
71         pstmt.setInt(3, mahasiswa.getSemester());
72         pstmt.setFloat(4, mahasiswa.getIpk());
73         pstmt.setInt(5, mahasiswa.getId());
74     } catch (SQLException e){
75         e.printStackTrace();
76     }
77 }
78
79 public void deleteMahasiswa(int id){
80     String sql = "DELETE FROM mahasiswa WHERE id = ?";
81     try{
82         PreparedStatement pstmt = connection.prepareStatement(sql);
83         pstmt.setInt(1, id);
84         pstmt.executeUpdate();
85     } catch (SQLException e){
86         e.printStackTrace();
87     }
88 }
```

Projects x Services

- RPL2_Pert1
- RPL2_Pert2_50421535
 - Source Packages
 - rpl2_pert2_50421535
 - Mahasiswa.java
 - RPL2_Pert2_50421535.java
 - Student.java
 - Test Packages
 - Libraries
 - Test Libraries
 - RPL2_Pert3_50421535
 - Source Packages
 - com.mahasiswa.controller
 - MahasiswaController.java
 - com.mahasiswa.model
 - MahasiswaDAO.java
 - ModelMahasiswa.java
 - com.mahasiswa.view

MahasiswaDAO.java - Navigator x

Members

 - MahasiswaDAO
 - MahasiswaDAO()
 - addMahasiswa(ModelMahasiswa mahasiswa)
 - cekKoneksi(): boolean
 - closeConnection()

Output - Run (RPL2_Pert3_50421535)

```
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
1
=====
ID          : 2
NPM         : 50421535
NAMA        : fizrri
SEMESTER    : 7
```

MahasiswaDAO.java x

```
76
77 }
78
79 public void deleteMahasiswa(int id){
80     String sql = "DELETE FROM mahasiswa WHERE id = ?";
81     try{
82         PreparedStatement pstmt = connection.prepareStatement(sql);
83         pstmt.setInt(1, id);
84         pstmt.executeUpdate();
85     } catch (SQLException e){
86         e.printStackTrace();
87     }
88 }
89
90 public void closeConnection(){
91     try{
92         if(connection != null){
93             connection.close();
94         }
95     } catch (SQLException e){
96         e.printStackTrace();
97     }
98 }
99 }
```

ModelMahasiswa

The screenshot shows an IDE with the following components:

- Projects:** RPL2_Pert1, RPL2_Pert2_50421535, RPL2_Pert3_50421535.
- Source Packages:** rpl2_pert2_50421535, rpl2_pert2_50421535, RPL2_Pert2_50421535.java, Student.java, Test Packages, Libraries, Test Libraries.
- Source Packages:** com.mahasiswa.controller, MahasiswaController.java, com.mahasiswa.model, MahasiswaDAO.java, ModelMahasiswa.java, com.mahasiswa.view.
- Navigator:** ModelMahasiswa, ModelMahasiswa(int id, String npm, String nama, int semester, float ipk), getId(): int, getIpk(): float, getNama(): String.
- Source:**

```
package com.mahasiswa.model;

public class ModelMahasiswa {
    private int id;
    private String npm;
    private String nama;
    private int semester;
    private float ipk;

    public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
        this.id = id;
        this.npm = npm;
        this.nama = nama;
        this.semester = semester;
        this.ipk = ipk;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }
}
```
- Output - Run (RPL2_Pert3_50421535):** 4. Hapus Mahasiswa, 5. Cek Koneksi Database, 6. Keluar, PILIH OPSI: 1, ID : 2, NPM : 50421535, NAMA : firzi, SEMESTER : 7.

The screenshot shows an IDE with the following components:

- Projects:** RPL2_Pert1, RPL2_Pert2_50421535, RPL2_Pert3_50421535.
- Source Packages:** rpl2_pert2_50421535, Mahasiswa.java, RPL2_Pert2_50421535.java, Student.java, Test Packages, Libraries, Test Libraries.
- Source Packages:** com.mahasiswa.controller, MahasiswaController.java, com.mahasiswa.model, MahasiswaDAO.java, ModelMahasiswa.java, com.mahasiswa.view.
- Navigator:** ModelMahasiswa, ModelMahasiswa(int id, String npm, String nama, int semester, float ipk), getId(): int, getIpk(): float, getNama(): String.
- Source:**

```
public String getNama() {
    return nama;
}

public void setNama(String nama) {
    this.nama = nama;
}

public int getSemester() {
    return semester;
}

public void setSemester(int semester) {
    this.semester = semester;
}

public float getIpk() {
    return ipk;
}

public void setIpk(float ipk) {
    this.ipk = ipk;
}
```
- Output - Run (RPL2_Pert3_50421535):** 4. Hapus Mahasiswa, 5. Cek Koneksi Database, 6. Keluar, PILIH OPSI: 1, ID : 2, NPM : 50421535, NAMA : firzi, SEMESTER : 7.

pom.xml

The screenshot shows an IDE with the `pom.xml` file open. The `Source` tab displays the XML content, which defines a Maven project with the following details:

- Group ID: `com.mahasiswa`
- Artifact ID: `RPL2_Pert3_50421535`
- Version: `1.0-SNAPSHOT`
- Packaging: `jar`
- Dependencies: `mysql:mysql-connector-java:8.0.33`

The `Output` tab shows the execution of the `Run (RPL2_Pert3_50421535)` command, displaying the following output:

```
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
1

=====
ID      : 2
NPM     : 50421535
NAMA    : fizri
SEMESTER : 7
```

MahasiswaView

The screenshot shows an IDE with the `MahasiswaView.java` file open. The `Source` tab displays the Java code, which includes the following imports and class definition:

```
package com.mahasiswa.view;

import com.mahasiswa.controller.MahasiswaController;
import com.mahasiswa.model.MahasiswaDAO;
import java.util.Scanner;

public class MahasiswaView {

    public static void main(String[] args) {
        MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
        MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);

        Scanner scanner = new Scanner(System.in);
        int pilihan;

        while(true){
            System.out.println("Menu :");
            System.out.println("1. Tampilkan Semua Mahasiswa");
            System.out.println("2. Tambah Mahasiswa");
            System.out.println("3. Update Mahasiswa");
            System.out.println("4. Hapus Mahasiswa");
            System.out.println("5. Cek Koneksi Database");
            System.out.println("6. Keluar");
            System.out.println("PILIH OPSI: ");
            pilihan = scanner.nextInt();
        }
    }
}
```

The `Output` tab shows the execution of the `Run (RPL2_Pert3_50421535)` command, displaying the following output:

```
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
1

=====
ID      : 2
NPM     : 50421535
NAMA    : fizri
SEMESTER : 7
```



```
pom.xml [RPL2_Pert3_50421535] x MahasiswaView.java x
Source History
16 while(true) {
17     System.out.println("Menu :");
18     System.out.println("1. Tampilkan Semua Mahasiswa");
19     System.out.println("2. Tambah Mahasiswa");
20     System.out.println("3. Update Mahasiswa");
21     System.out.println("4. Hapus Mahasiswa");
22     System.out.println("5. Cek Koneksi Database");
23     System.out.println("6. Keluar");
24     System.out.println("PILIH OPSI: ");
25     pilihan = scanner.nextInt();
26     scanner.nextLine();
27
28     switch(pilihan) {
29         case 1:
30             mahasiswaController.displayAllMahasiswa();
31             break;
32         case 2:
33             System.out.println("Masukan NPM: ");
34             String npm = scanner.next();
35             System.out.println("Masukan Nama: ");
36             String nama = scanner.next();
37             System.out.println("Masukan Semester: ");
38             int semester = scanner.nextInt();
39             System.out.println("Masukan IPK: ");
40             float ipk = scanner.nextFloat();
```

```
pom.xml [RPL2_Pert3_50421535] x MahasiswaView.java x
Source History
34 String npm = scanner.next();
35 System.out.println("Masukan Nama: ");
36 String nama = scanner.next();
37 System.out.println("Masukan Semester: ");
38 int semester = scanner.nextInt();
39 System.out.println("Masukan IPK: ");
40 float ipk = scanner.nextFloat();
41
42 mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
43 break;
44 case 3:
45     System.out.println("Masukan ID Mahasiswa: ");
46     int id = scanner.nextInt();
47     scanner.nextLine();
48
49     System.out.println("Masukan NPM: ");
50     String npmBaru = scanner.next();
51     System.out.println("Masukan Nama: ");
52     String namaBaru = scanner.next();
53     System.out.println("Masukan Semester");
54     int semesterBaru = scanner.nextInt();
55     System.out.println("Masukan IPK: ");
56     float ipkBaru = scanner.nextFloat();
57
```

```
pom.xml [RPL2_Pert3_50421535] x MahasiswaView.java x
Source History
51      System.out.println("Masukan Nama: ");
52      String namaBaru = scanner.next();
53      System.out.println("Masukan Semester");
54      int semesterBaru = scanner.nextInt();
55      System.out.println("Masukan IPK: ");
56      float ipkBaru = scanner.nextFloat();
57
58      mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semest
59      break;
60      case 4:
61      System.out.println("Masukan ID Mahasiswa yg Ingin Dihapus: ");
62      int idHapus = scanner.nextInt();
63      mahasiswaController.deleteMahasiswa(idHapus);
64      break;
65      case 5:
66      mahasiswaController.checkDatabaseConnection();
67      break;
68      case 6:
69      mahasiswaController.closeConnection();
70      System.out.println("Program selesai");
71      return;
72      default:
73      System.out.println("Input Tidak Valid");
74  }
75 }
```

Output

```
-- exec:3.1.0:exec (default-cli) @ k
Menu :
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI:
2
Masukan NPM:
50421535
Masukan Nama:
fizri
Masukan Semester:
7
Masukan IPK:
4
Mahasiswa berhasil ditambahkan
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         : 50421535
NAMA        : fizri
SEMESTER    : 7
IPK         : 4.0
=====
```