

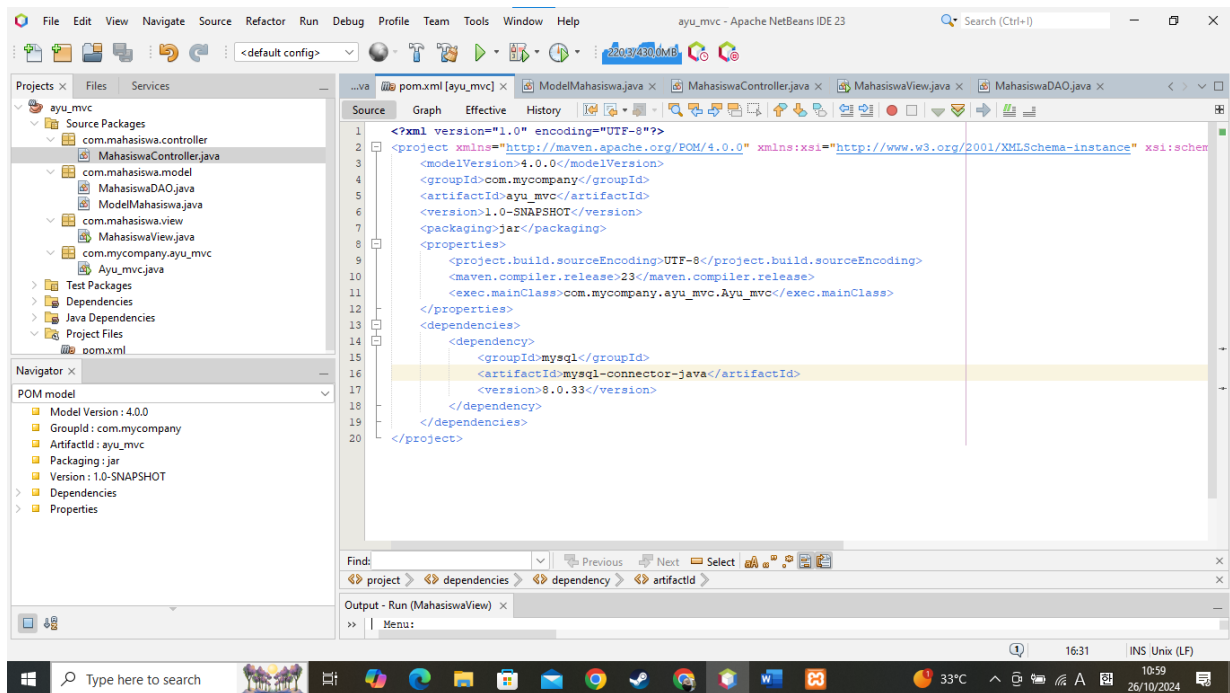
ACTIVITY PERTEMUAN 3

NAMA : AYU RETNONINGRUM SUSILO
NPM : 50421244
KELAS : 4IA28
MATERI : MVC
MATA PRAKTIKUM : REKAYASA PERANGKAT LUNAK 2

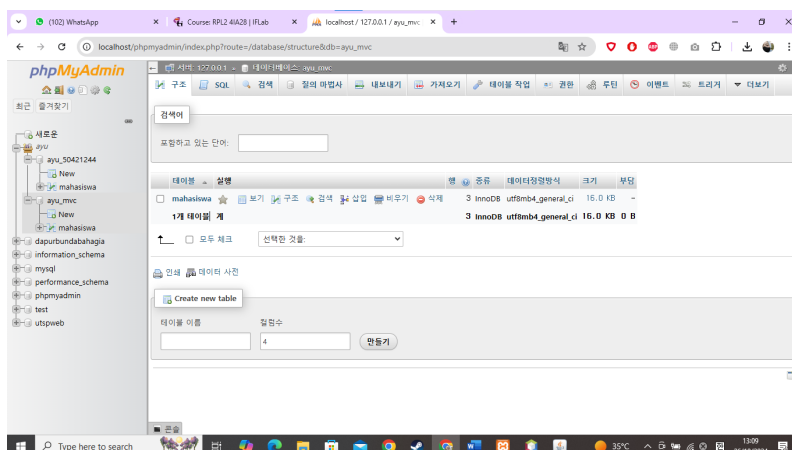
Screenshot source code dan output

Jawab:

Source Code pom



XAMPP



Source Code Controller

```
...va pom.xml | ayu_mv | ModelMahasiswa.java | MahasiswaController.java | MahasiswaView.java | MahasiswaDAO.java |
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.mahasiswa.controller;
6  import com.mahasiswa.model.MahasiswaDAO;
7  import com.mahasiswa.model.ModelMahasiswa;
8  import java.util.List;
9
10 /**
11  *
12  * @author enoss1
13  */
14 public class MahasiswaController {
15     private MahasiswaDAO mahasiswaDAO;
16
17     public MahasiswaController(MahasiswaDAO mahasiswaDAO) {
18         this.mahasiswaDAO = mahasiswaDAO;
19     }
20
21     public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
22         if (mahasiswaList.isEmpty()) {
23             System.out.println("Tidak ada data mahasiswa");
24         } else {
25             System.out.println("");
26             System.out.println("=====");
27
28             System.out.println("ID           : " + m.getId());
29             System.out.println("NPM          : " + m.getNpm());
30             System.out.println("NAMA         : " + m.getName());
31             System.out.println("SEMESTER     : " + m.getSemester());
32             System.out.println("IPK          : " + m.getIpk());
33             System.out.println("=====");
34         }
35     }
36
37     public void displayMessage(String message) {
38         System.out.println(message);
39     }
40
41     public void checkDatabaseConnection() {
42         boolean isConnected = mahasiswaDAO.checkConnection();
43         if (isConnected) {
44             displayMessage("Koneksi ke db berhasil");
45         } else {
46             displayMessage("Koneksi DB Gagal");
47         }
48     }
49
50     // READ ALL (Menampilkan semua mahasiswa)
51     public void displayAllMahasiswa() {
52         List<ModelMahasiswa> mahasiswaList = mahasiswaDAO.getAllMahasiswa();
53         displayMahasiswaList(mahasiswaList);
54     }
55
56     public void addMahasiswa(String npm, String nama, int semester, float ipk) {
57         ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(0, npm, nama, semester, ipk);
58         System.out.println("Controller Data: " + npm + nama + semester + ipk);
59         System.out.println(mahasiswaBaru);
60         mahasiswaDAO.addMahasiswa(mahasiswaBaru);
61         displayMessage("Mahasiswa berhasil ditambahkan!");
62     }
63
64     public void updateMahasiswa(int id, String npm, String nama, int semester, float ipk) {
65         ModelMahasiswa mahasiswaBaru = new ModelMahasiswa(id, npm, nama, semester, ipk);
66         mahasiswaDAO.updateMahasiswa(mahasiswaBaru);
67         displayMessage("Mahasiswa berhasil diperbarui!");
68     }
69
70     public void deleteMahasiswa(int id) {
71         mahasiswaDAO.deleteMahasiswa(id);
72         displayMessage("Mahasiswa Berhasil Dihapus!");
73     }
74
75     public void closeConnection() {
76         mahasiswaDAO.closeConnection();
77     }
78
79 }
```

Source Code DAO

```
...va pom.xml [ayu_mvc] x ModelMahasiswa.java x MahasiswaController.java x MahasiswaView.java x MahasiswaDAO.java 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.mahasiswa.model;
6  import java.sql.*;
7  import java.util.List;
8  import java.util.ArrayList;
9  /**
10   *
11   * @author enoss1
12   */
13  public class MahasiswaDAO {
14      private Connection connection;
15
16      public MahasiswaDAO() {
17          try{
18              Class.forName("com.mysql.cj.jdbc.Driver");
19              connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/ayu_mvc", "root", "");
20          } catch (Exception e) {
21              e.printStackTrace();
22          }
23      }
24
25      public boolean checkConnection() {
26
27          if(connection != null && !connection.isClosed()) {
28              return true;
29          }
30          } catch (SQLException e) {
31              e.printStackTrace();
32          }
33          return false;
34      }
35
36      public void addMahasiswa(ModelMahasiswa mahasiswa){
37          String sql = "INSERT INTO mahasiswa (npm, nama, semester, ipk) VALUES (?, ?, ?, ?)";
38          try{
39              PreparedStatement pstmt = connection.prepareStatement(sql);
40              pstmt.setString(1, mahasiswa.getNpm());
41              pstmt.setString(2, mahasiswa.getNama());
42              pstmt.setInt(3, mahasiswa.getSemester());
43              pstmt.setFloat(4, mahasiswa.getIpk());
44              pstmt.executeUpdate();
45          } catch (SQLException e) {
46              e.printStackTrace();
47          }
48      }
49
50      public List<ModelMahasiswa> getAllMahasiswa(){
51          List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
52          String sql = "SELECT * FROM mahasiswa";
53          try{
54              Statement stmt = connection.createStatement();
55
56              ResultSet rs = stmt.executeQuery(sql);
57              while(rs.next()){
58                  mahasiswaList.add(new ModelMahasiswa(
59                      rs.getInt("id"),
60                      rs.getString("npm"),
61                      rs.getString("nama"),
62                      rs.getInt("semester"),
63                      rs.getFloat("ipk")
64                  ));
65              }
66          } catch (SQLException e) {
67              e.printStackTrace();
68          }
69          return mahasiswaList;
70      }
71
72      public void updateMahasiswa(ModelMahasiswa mahasiswa){
73          String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?";
74          try{
75              PreparedStatement pstmt = connection.prepareStatement(sql);
76              pstmt.setString(1, mahasiswa.getNpm());
77              pstmt.setString(2, mahasiswa.getNama());
78              pstmt.setInt(3, mahasiswa.getSemester());
79              pstmt.setFloat(4, mahasiswa.getIpk());
80              pstmt.setInt(5, mahasiswa.getId());
81              pstmt.executeUpdate();
82          }
```

```

79         } catch (SQLException e) {
80             e.printStackTrace();
81         }
82     }
83     public void deleteMahasiswa(int id) {
84         String sql = "DELETE from mahasiswa where id = ?";
85         try {
86             PreparedStatement pstmt = connection.prepareStatement(sql);
87             pstmt.setInt(1, id);
88             pstmt.executeUpdate();
89         } catch (SQLException e) {
90             e.printStackTrace();
91         }
92     }
93     public void closeConnection() {
94         try {
95             if (connection != null) {
96                 connection.close();
97             }
98         } catch (SQLException e) {
99             e.printStackTrace();
100         }
101     }
102 }

```

Source Code Model

```

...va pom.xml [ayu_mvc] x ModelMahasiswa.java x MahasiswaController.java x MahasiswaView.java x MahasiswaDAO.java 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.mahasiswa.model;
6
7   /**
8    *
9    * @author enoss1
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
28           return npm;
29       }
30
31       public void setNpm(String npm) {
32           this.npm = npm;
33       }
34
35       public String getNama() {
36           return nama;
37       }
38
39       public void setNama(String nama) {
40           this.nama = nama;
41       }
42
43       public int getSemester() {
44           return semester;
45       }
46
47       public void setSemester(int semester) {
48           this.semester = semester;
49       }
50
51       public float getIpk() {
52           return ipk;

```

```

53
54     public void setIpk(float ipk) {
55         this.ipk = ipk;
56     }
57
58
59
60     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk){
61         this.id = id;
62         this.npm = npm;
63         this.nama = nama;
64         this.semester = semester;
65         this.ipk = ipk ;
66     }
67
68
69
70 }

```

Source Code View

```

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.mahasiswa.view;
6  import com.mahasiswa.controller.MahasiswaController;
7  import com.mahasiswa.model.MahasiswaDAO;
8  import java.util.Scanner;
9
10 /**
11  * @author enoss1
12  */
13 public class MahasiswaView {
14     public static void main(String[] args){
15         MahasiswaDAO mahasiswaDAO = new MahasiswaDAO();
16         MahasiswaController mahasiswaController = new MahasiswaController(mahasiswaDAO);
17
18         Scanner scanner = new Scanner(System.in);
19         int pilihan;
20
21         while(true){
22             System.out.println("Menu:");
23             System.out.println("1. Tampilkan Semua Mahasiswa");
24             System.out.println("2. Tambah Mahasiswa");
25             System.out.println("3. Update Mahasiswa");
26             System.out.println("4. Hapus Mahasiswa");
27
28             System.out.println("5. Cek Koneksi Database");
29             System.out.println("6. Keluar");
30             System.out.print("PILIH OPSI: ");
31             pilihan = scanner.nextInt();
32             scanner.nextLine();
33
34             switch (pilihan){
35                 case 1:
36                     mahasiswaController.displayAllMahasiswa();
37                     break;
38                 case 2:
39                     // tambah mhs
40                     System.out.println("Masukkan NPM: ");
41                     String npm = scanner.next();
42                     System.out.println("Masukkan Nama: ");
43                     String nama = scanner.next();
44                     System.out.println("Masukkan Semester: ");
45                     int semester = scanner.nextInt();
46                     System.out.println("Masukkan IPK: ");
47                     float ipk = scanner.nextFloat();
48                     System.out.println(npm + nama + semester + ipk);
49
50                     mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
51                     break;
52

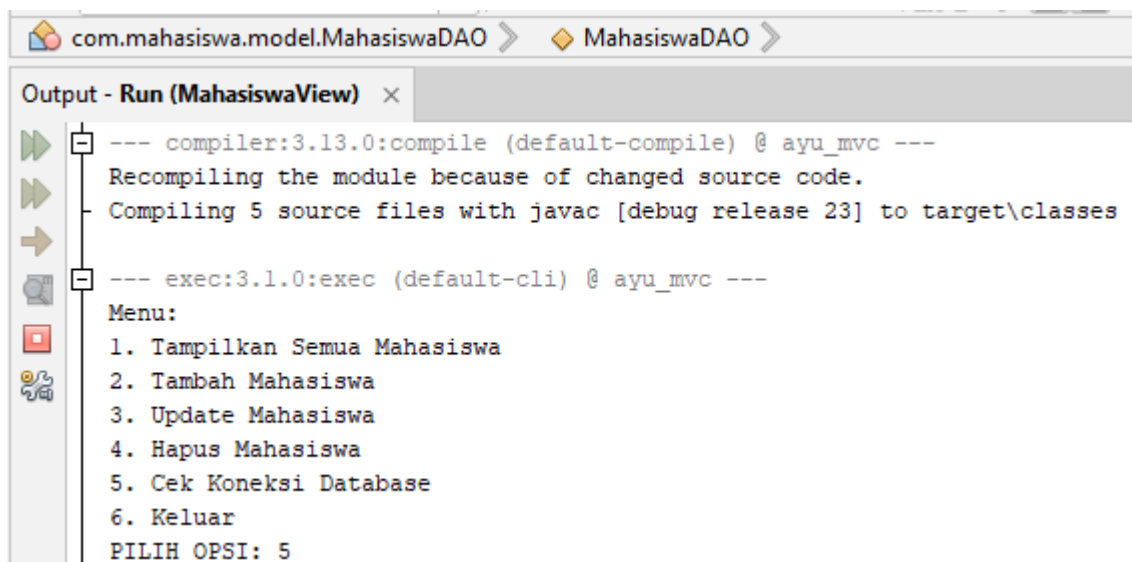
```

```

53:         case 3:
54:             System.out.print("Masukkan ID mahasiswa: ");
55:             int id = scanner.nextInt();
56:             scanner.nextLine();
57:
58:             System.out.println("Masukkan NPM: ");
59:             String npmBaru = scanner.next();
60:             System.out.println("Masukkan Nama: ");
61:             String namaBaru = scanner.next();
62:             System.out.println("Masukkan Semester: ");
63:             int semesterBaru = scanner.nextInt();
64:             System.out.println("Masukkan IPK: ");
65:             float ipkBaru = scanner.nextFloat();
66:
67:             mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
68:             break;
69:         case 4:
70:             System.out.print("Masukkan ID Mahasiswa: ");
71:             int idHapus = scanner.nextInt();
72:             mahasiswaController.deleteMahasiswa(idHapus);
73:         case 5:
74:             mahasiswaController.checkDatabaseConnection();
75:             break;
76:         case 6:
77:             // Keluar
78:             mahasiswaController.closeConnection();
79:
80:             System.out.println("Program selesai.");
81:             return;
82:         default:
83:             System.out.println("Input Tidak valid");
84:     }
85: }
86:

```

Output



The screenshot shows the 'Output - Run (MahasiswaView)' window of an IDE. The top bar indicates the package 'com.mahasiswa.model.MahasiswaDAO' and the class 'MahasiswaDAO'. The output log contains the following text:

```

--- compiler:3.13.0:compile (default-compile) @ ayu_mvc ---
Recompiling the module because of changed source code.
Compiling 5 source files with javac [debug release 23] to target\classes

--- exec:3.1.0:exec (default-cli) @ ayu_mvc ---
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: 5

```

Koneksi Database

```
com.mahasiswa.model.MahasiswaDAO > MahasiswaDAO >
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:
```

Tambah Data

```
com.mahasiswa.view.MahasiswaView > main >
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: 2
Masukkan NPM:
50421244
Masukkan Nama:
AyuRetnoningrumSusilo
Masukkan Semester:
7
Masukkan IPK:
3,8
50421244AyuRetnoningrumSusilo73.8
Controller Data: 50421244AyuRetnoningrumSusilo73.8
com.mahasiswa.model.ModelMahasiswa@1990a65e
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:
```

Tampilkan Data

```
Output - Run (MahasiswaView) x
PILIH OPSI: 1

=====
ID      : 2
NPM     : 50421244
NAMA    : AyuRetnoningrumSusilo
SEMESTER : 7
IPK     : 3.8
=====
ID      : 3
NPM     : 50421212
NAMA    : KimJongyun
SEMESTER : 2
IPK     : 3.9
=====
ID      : 4
NPM     : 50421243
NAMA    : KimSori
SEMESTER : 8
IPK     : 3.7
=====
Menu:
1. Tampilkan Semua Mahasiswa
2. Tambah Mahasiswa
3. Update Mahasiswa
4. Hapus Mahasiswa
5. Cek Koneksi Database
6. Keluar
PILIH OPSI: |
```

Update Data

```
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: 3
Masukkan ID mahasiswa: 2
Masukkan NPM:
50421244
Masukkan Nama:
Ayuuuuuuu
Masukkan Semester:
8
Masukkan IPK:
3,9
Mahasiswa berhasil diperbarui!
```


Hapus Data

```
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: 4
Masukkan ID Mahasiswa: 1
Mahasiswa Berhasil Dihapus!
Koneksi ke db berhasil
```

Tampilan Data Akhir

```
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: 1

=====
ID      : 2
NPM     : 50421244
NAMA    : Ayuuuuuuuu
SEMESTER : 8
IPK     : 3.9
=====
ID      : 3
NPM     : 50421212
NAMA    : KimJongyun
SEMESTER : 2
IPK     : 3.9
=====
ID      : 4
NPM     : 50421243
NAMA    : KimSori
SEMESTER : 8
IPK     : 3.7
=====
```

Keluar

```
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: 6
Program selesai.

-----
BUILD SUCCESS
-----

Total time: 05:22 min
Finished at: 2024-10-26T10:58:25+07:00
-----
```

ACTIVITY PERTEMUAN 4

NAMA : AYU RETNONINGRUM SUSILO

NPM : 50421244

KELAS : 4IA28

MATERI : ORM

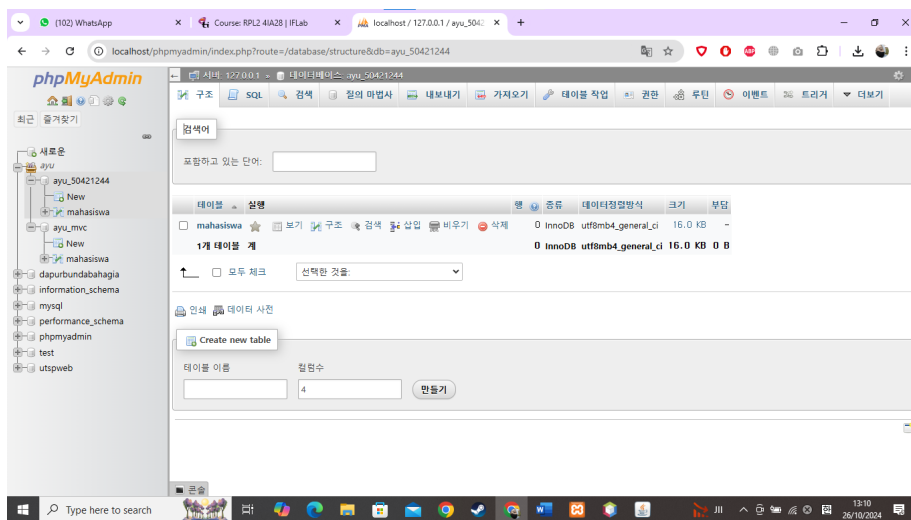
MATA PRAKTIKUM : REKAYASA PERANGKAT LUNAK 2

Screenshot source code dan output

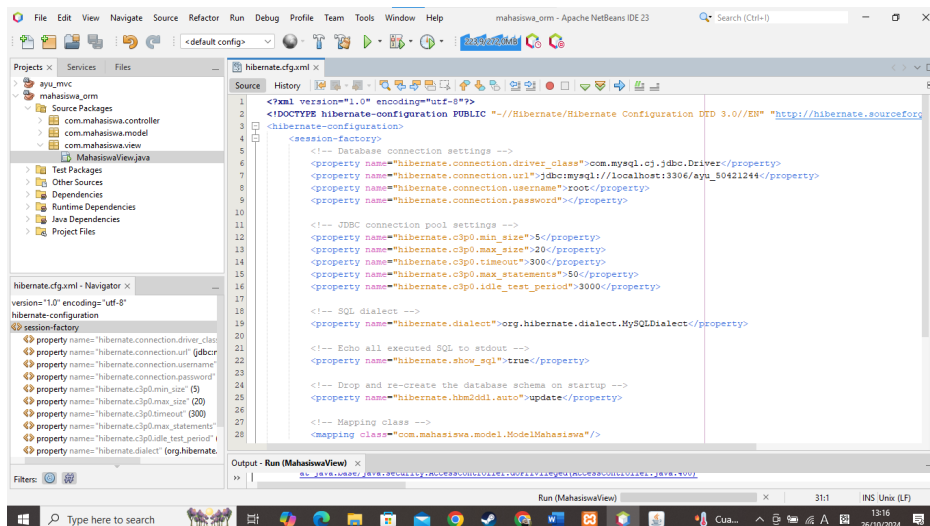
Jawab:

Source Code

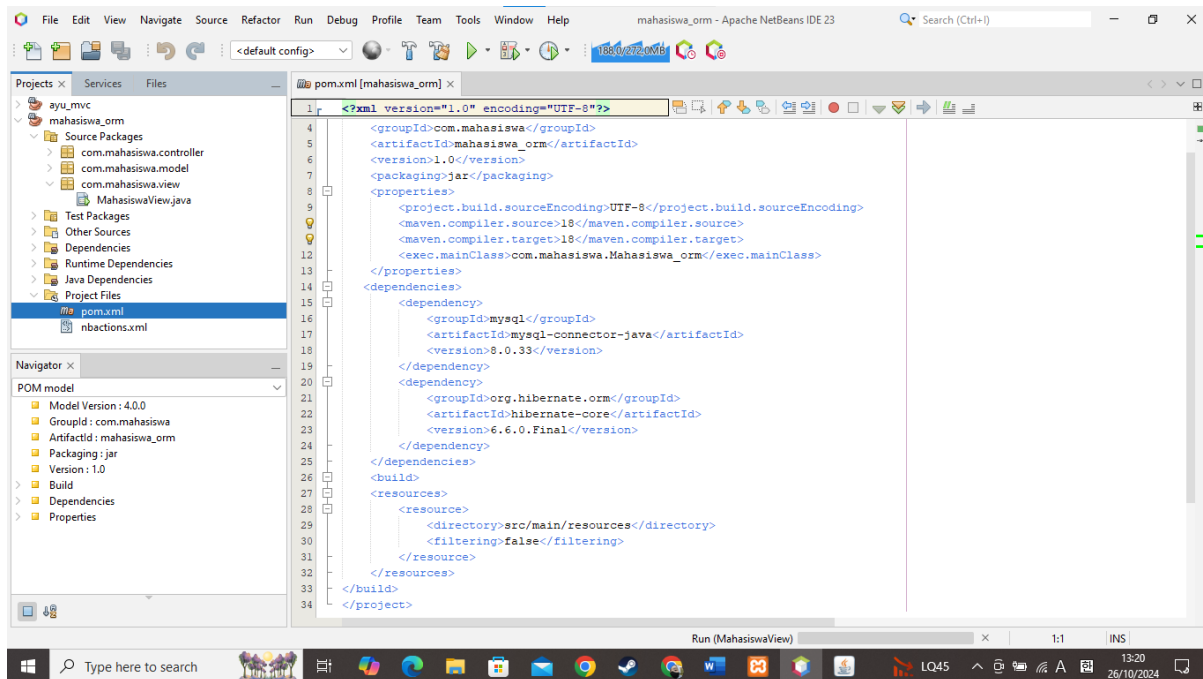
XAMPP



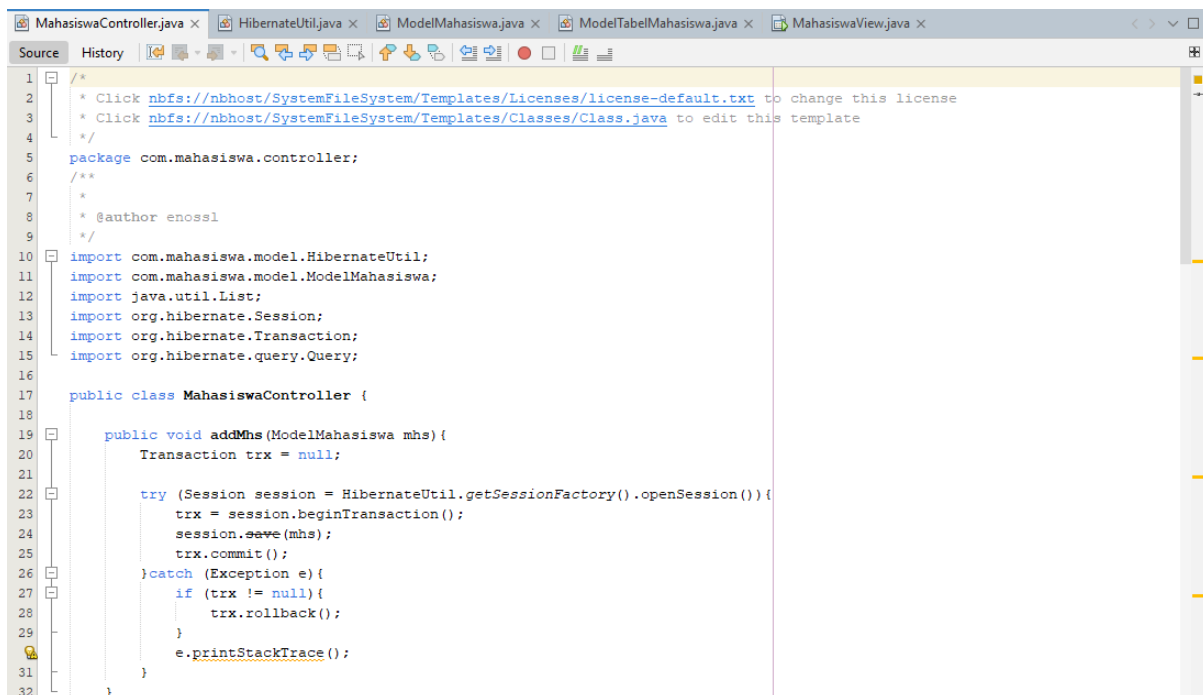
Hibernate.cfg.xml



Pom



Mahasiswa Controller



```

33
34
35 public void updateMhs(ModelMahasiswa mhs) {
36     Transaction trx = null;
37
38     try (Session session = HibernateUtil.getSessionFactory().openSession()) {
39         trx = session.beginTransaction();
40         session.update(mhs);
41         trx.commit();
42     } catch (Exception e) {
43         if (trx != null) {
44             trx.rollback();
45         }
46         e.printStackTrace();
47     }
48
49 }
50
51 public void deleteMhs(int id) {
52     Transaction trx = null;
53
54     try (Session session = HibernateUtil.getSessionFactory().openSession()) {
55         trx = session.beginTransaction();
56         ModelMahasiswa mhs = session.get(ModelMahasiswa.class, id);
57         if (mhs != null) {
58             session.delete(mhs);
59             System.out.println("Berhasil hapus");
60         }
61         trx.commit();
62     } catch (Exception e) {
63         if (trx != null) {
64             trx.rollback();

```

```

65         e.printStackTrace();
66     }
67 }
68
69 }
70
71 public List<ModelMahasiswa> getAllMahasiswa() {
72     Transaction trx = null;
73     List<ModelMahasiswa> listMhs = null;
74
75     try (Session session = HibernateUtil.getSessionFactory().openSession()) {
76         trx = session.beginTransaction();
77         // Using HQL (Hibernate Query Language) to fetch all records
78         Query<ModelMahasiswa> query = session.createQuery("from ModelMahasiswa", ModelMahasiswa.class);
79         listMhs = query.list(); // Fetch all results
80
81         trx.commit(); // Commit transaction
82     } catch (Exception e) {
83         if (trx != null) {
84             trx.rollback(); // Rollback transaction in case of error
85         }
86         e.printStackTrace();
87     }
88
89     // Return the fetched list
90     return listMhs;
91 }
92

```

Hibernate Util

```
MahasiswaController.java x HibernateUtil.java x ModelMahasiswa.java x ModelTabelMahasiswa.java x MahasiswaView.java x
Source History
1 package com.mahasiswa.model;
2
3 import org.hibernate.Session;
4 import org.hibernate.SessionFactory;
5 import org.hibernate.cfg.Configuration;
6
7 /**
8  *
9  * @author enoss1
10 */
11 public class HibernateUtil {
12     private static SessionFactory sessionFactory;
13
14     static {
15         try {
16             // Create the SessionFactory from hibernate.cfg.xml
17             sessionFactory = new Configuration().configure().buildSessionFactory();
18         } catch (Throwable ex) {
19             // Make sure you log the exception, as it might be swallowed
20             System.err.println("Initial SessionFactory creation failed." + ex);
21             throw new ExceptionInInitializerError(ex);
22         }
23     }
24
25     public static SessionFactory getSessionFactory() {
26         return sessionFactory;
27     }
28
29     public static void testConnection() {
30         try (Session session = sessionFactory.openSession()) {
31             System.out.println("Connection to the database was successful!");
32         } catch (Exception e) {
33             System.err.println("Failed to connect to the database.");
34             e.printStackTrace();
35         }
36     }
37 }
```

Model Mahasiswa

```
MahasiswaController.java x HibernateUtil.java x ModelMahasiswa.java x ModelTabelMahasiswa.java x MahasiswaView.java 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.mahasiswa.model;
6
7 import jakarta.persistence.*;
8
9 /**
10  *
11  * @author enoss1
12  */
13 @Entity
14 @Table(name = "mahasiswa")
15 public class ModelMahasiswa {
16
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     @Column(name = "id")
20     private int id;
21
22     @Column(name = "npm", nullable = false, length = 8)
23     private String npm;
24
25     @Column(name = "nama", nullable = false, length = 50)
26     private String nama;
27
28     @Column(name = "semester")
29     private int semester;
30
31     @Column(name = "ipk")
32     private float ipk;
33 }
```

```

33
34     public ModelMahasiswa() {
35     }
36
37
38     public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk){
39         this.id = id;
40         this.npm = npm;
41         this.nama = nama;
42         this.semester = semester;
43         this.ipk = ipk;
44     }
45
46     public int getId() {
47         return id;
48     }
49
50     public void setId(int id) {
51         this.id = id;
52     }
53
54     public String getNpm() {
55         return npm;
56     }
57
58     public void setNpm(String npm) {
59         this.npm = npm;
60     }
61
62     public String getNama() {
63         return nama;
64     }

```

```

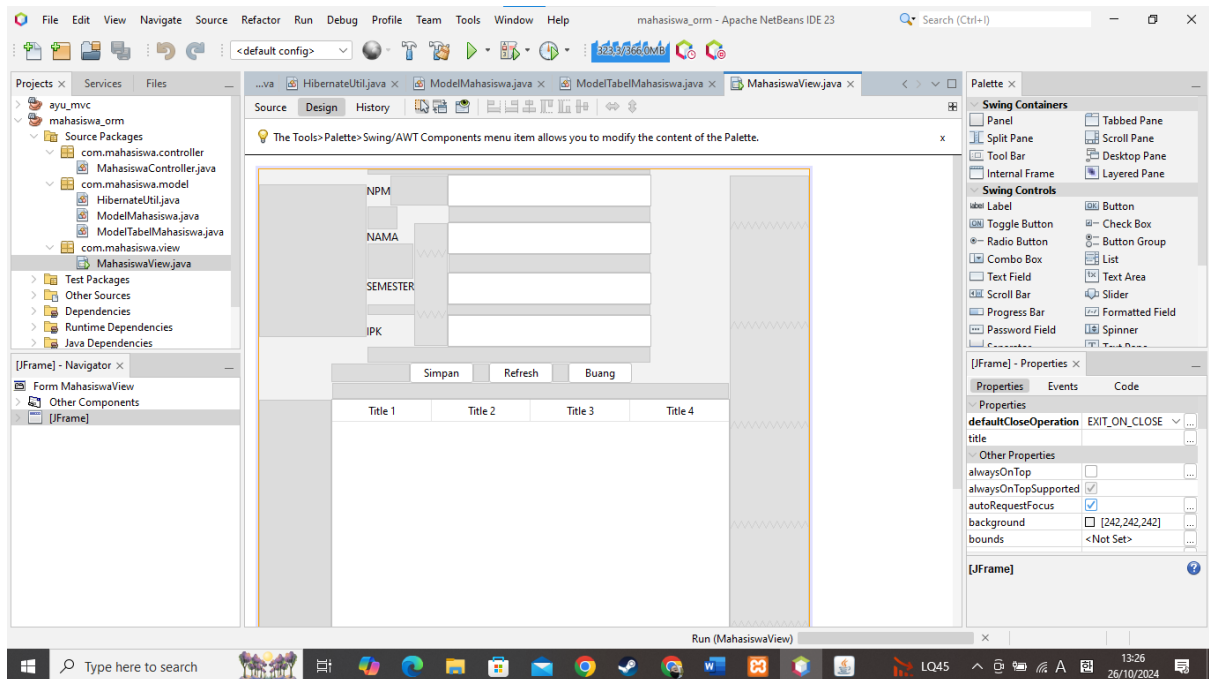
65
66     public void setName(String nama) {
67         this.nama = nama;
68     }
69
70     public int getSemester() {
71         return semester;
72     }
73
74     public void setSemester(int semester) {
75         this.semester = semester;
76     }
77
78     public float getIpk() {
79         return ipk;
80     }
81
82     public void setIpk(float ipk) {
83         this.ipk = ipk;
84     }
85
86
87
88 }

```

Model Tabel Mahasiswa

```
MahasiswaController.java x HibernateUtil.java x ModelMahasiswa.java x ModelTabelMahasiswa.java x MahasiswaView.java 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.mahasiswa.model;
6  import javax.swing.table.AbstractTableModel;
7  import java.util.List;
8  /**
9   *
10  * @author enoss1
11  */
12  public class ModelTabelMahasiswa extends AbstractTableModel{
13      private List<ModelMahasiswa> mahasiswaList;
14      private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPK"};
15
16      public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
17          this.mahasiswaList = mahasiswaList;
18      }
19
20      @Override
21      public int getRowCount() {
22          return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data mahasiswa
23      }
24
25      @Override
26      public int getColumnCount() {
27          return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen dalam columnNames
28      }
29
30      @Override
31      public Object getValueAt(int rowIndex, int columnIndex) {
32          ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
33
34          switch (columnIndex) {
35              case 0:
36                  return mahasiswa.getId();
37              case 1:
38                  return mahasiswa.getNpm();
39              case 2:
40                  return mahasiswa.getNama();
41              case 3:
42                  return mahasiswa.getSemester();
43              case 4:
44                  return mahasiswa.getIpk();
45              default:
46                  return null;
47          }
48      }
49
50      @Override
51      public String getColumnName(int column) {
52          return columnNames[column]; // Mengatur nama kolom
53      }
54
55      @Override
56      public boolean isCellEditable(int rowIndex, int columnIndex) {
57          return false; // Semua sel tidak dapat diedit
58      }
59
60      // Method untuk menambahkan atau memodifikasi data, jika dibutuhkan
61      public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
62          this.mahasiswaList = mahasiswaList;
63          fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
64      }
65  }
```

Mahasiswa View



Output

NPM

NAMA

SEMESTER

IPK

ID	NPM	Nama	Semester	IPK

Tambah Data

NPM

50421244

NAMA

Ayuu

SEMESTER

7

IPK

3.8

Simpan

Refresh

Buang

ID	NPM	Nama	Semester	IPK
1	50421244	Ayuu	7	3.8

Tampilan setelah penambahan data

NPM

50421203

NAMA

Yuni

SEMESTER

1

IPK

3.9

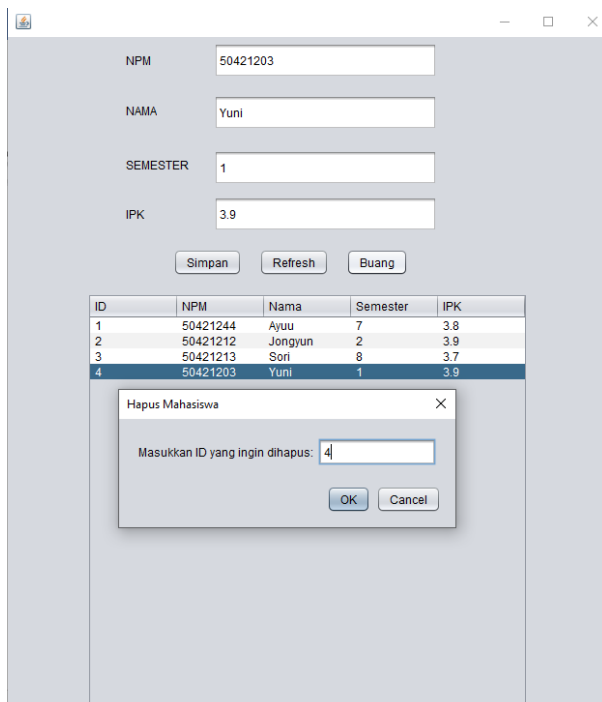
Simpan

Refresh

Buang

ID	NPM	Nama	Semester	IPK
1	50421244	Ayuu	7	3.8
2	50421212	Jongyun	2	3.9
3	50421213	Sori	8	3.7
4	50421203	Yuni	1	3.9

Hapus data



NPM: 50421203

NAMA: Yuni

SEMESTER: 1

IPK: 3.9

Simpan Refresh Buang

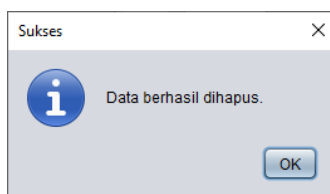
ID	NPM	Nama	Semester	IPK
1	50421244	Ayuu	7	3.8
2	50421212	Jongyun	2	3.9
3	50421213	Sori	8	3.7
4	50421203	Yuni	1	3.9

Hapus Mahasiswa

Masukkan ID yang ingin dihapus: 4

OK Cancel

Berhasil hapus data

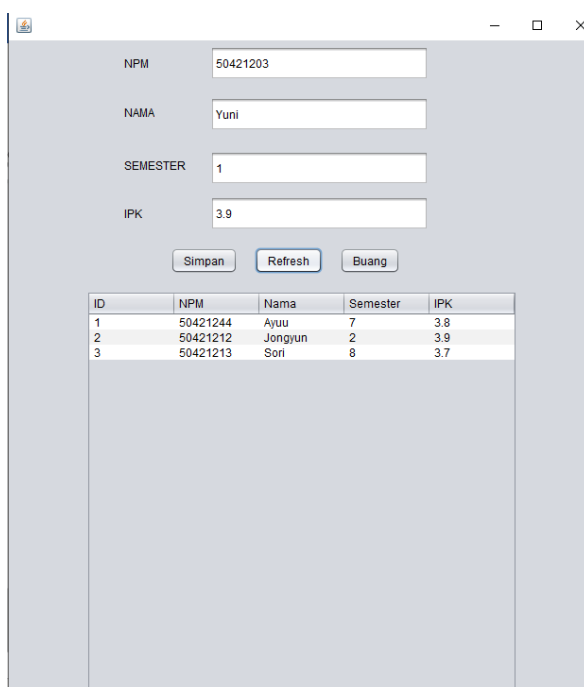


Sukses

Data berhasil dihapus.

OK

Tampilan seteah hapus data



NPM: 50421203

NAMA: Yuni

SEMESTER: 1

IPK: 3.9

Simpan Refresh Buang

ID	NPM	Nama	Semester	IPK
1	50421244	Ayuu	7	3.8
2	50421212	Jongyun	2	3.9
3	50421213	Sori	8	3.7