

ACTIVITY PERTEMUAN 3

NAMA : Arya Wahyu Wijaya

NPM : 50421223

KELAS : 4IA14

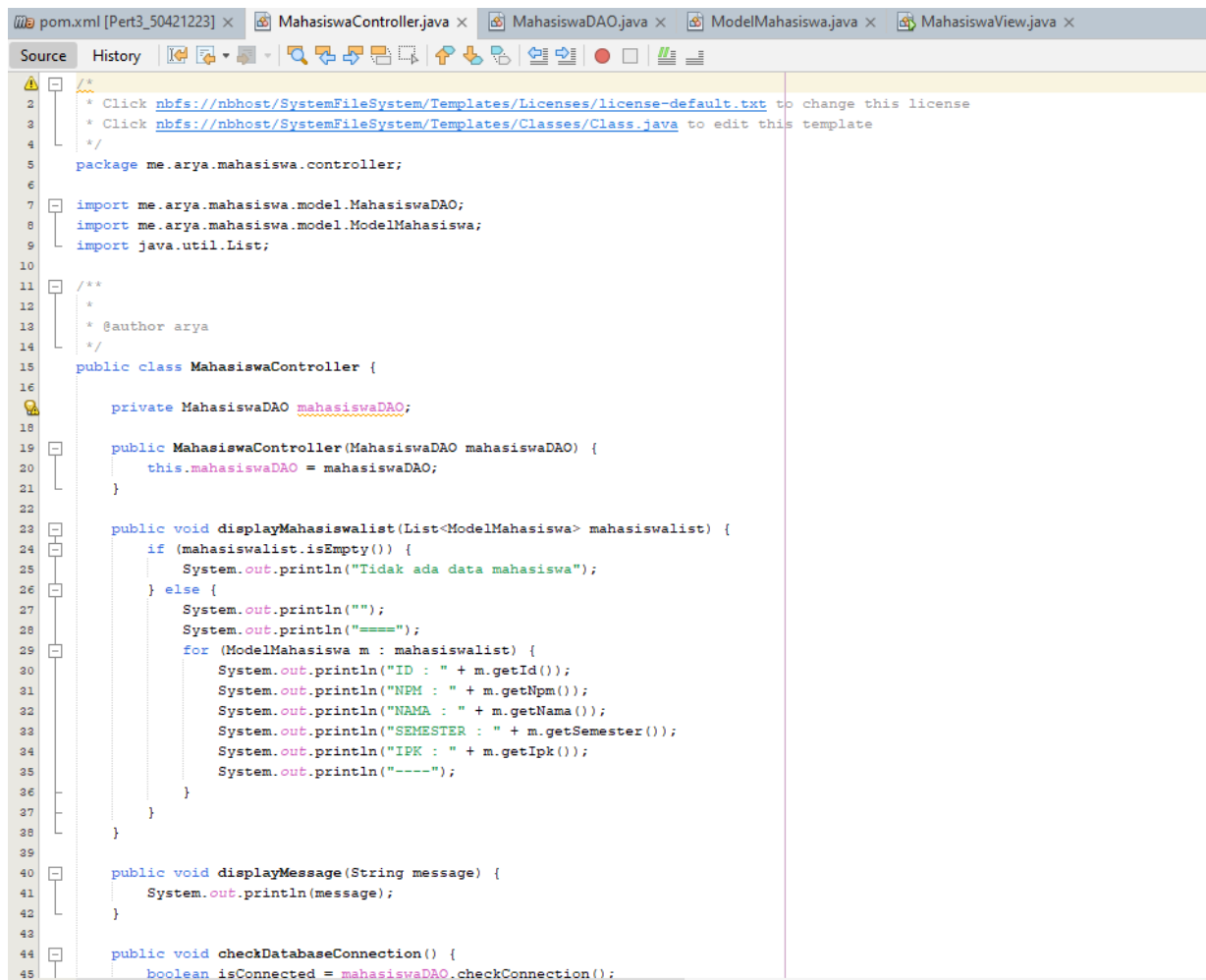
MATERI : MVC

MATA PRAKTIKUM : RPL2

1. Screenshot Code dan Output!

CODE:

- MahasiswaController.java



```
1  pom.xml [Pert3_50421223] x  MahasiswaController.java x  MahasiswaDAO.java x  ModelMahasiswa.java x  MahasiswaView.java x
2  Source History
3  /**
4   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
5   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
6   */
7  package me.arya.mahasiswa.controller;
8
9  import me.arya.mahasiswa.model.MahasiswaDAO;
10 import me.arya.mahasiswa.model.ModelMahasiswa;
11 import java.util.List;
12
13 /**
14  *
15  * @author arya
16  */
17 public class MahasiswaController {
18
19     private MahasiswaDAO mahasiswaDAO;
20
21     public MahasiswaController(MahasiswaDAO mahasiswaDAO) {
22         this.mahasiswaDAO = mahasiswaDAO;
23     }
24
25     public void displayMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
26         if (mahasiswaList.isEmpty()) {
27             System.out.println("Tidak ada data mahasiswa");
28         } else {
29             System.out.println("");
30             System.out.println("====");
31             for (ModelMahasiswa m : mahasiswaList) {
32                 System.out.println("ID : " + m.getId());
33                 System.out.println("NPM : " + m.getNpm());
34                 System.out.println("NAMA : " + m.getNama());
35                 System.out.println("SEMESTER : " + m.getSemester());
36                 System.out.println("IPK : " + m.getIpk());
37                 System.out.println("----");
38             }
39         }
40     }
41
42     public void displayMessage(String message) {
43         System.out.println(message);
44     }
45
46     public void checkDatabaseConnection() {
47         boolean isConnected = mahasiswaDAO.checkConnection();
48     }
49 }
```

```

44 public void checkDatabaseConnection() {
45     boolean isConnected = mahasiswaDAO.checkConnection();
46     if (isConnected) {
47         displayMessage("Koneksi ke db berhasil");
48     } else {
49         displayMessage("Koneksi DB Gagal");
50     }
51 }
52
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     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 }

```

- MahasiswaDAO.java

pom.xml [Pert3_50421223] x MahasiswaController.java x MahasiswaDAO.java x ModelMahasiswa.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 me.arya.mahasiswa.model;
6
7   import java.sql.*;
8   import java.util.ArrayList;
9   import java.util.List;
10
11  /**
12   *
13   * @author arya
14   */
15  public class MahasiswaDAO {
16
17      private Connection connection;
18
19      public MahasiswaDAO() {
20          try {
21              Class.forName("com.mysql.cj.jdbc.Driver");
22              connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/mvc_50421223", "root", ""); // URL typo diperbaiki
23          } catch (Exception e) {
24              e.printStackTrace();
25          }
26      }
27
28      public boolean checkConnection() {
29          try {
30              if (connection != null && !connection.isClosed()) {
31                  return true;
32              }
33          } catch (SQLException e) {
34              e.printStackTrace();
35          }
36          return false;
37      }
38
39      public void closeConnection() {
40          try {
41              if (connection != null) {
42                  connection.close();
43              }
44          } catch (SQLException e) {
45              e.printStackTrace();
46          }
47      }
48  }
```

```
pom.xml [Pert3_50421223] x MahasiswaController.java x MahasiswaDAO.java x ModelMahasiswa.java x MahasiswaView.java x
Source History
53 pstmt.setString(1, mahasiswa.getNpm());
54 pstmt.setString(2, mahasiswa.getNama());
55 pstmt.setInt(3, mahasiswa.getSemester());
56 pstmt.setFloat(4, mahasiswa.getIpk());
57 pstmt.executeUpdate();
58 } catch (SQLException e) {
59     e.printStackTrace();
60 }
61 }
62
63 public List<ModelMahasiswa> getAllMahasiswa() {
64     List<ModelMahasiswa> mahasiswaList = new ArrayList<>();
65     String sql = "SELECT * FROM mahasiswa";
66     try {
67         Statement stmt = connection.createStatement();
68         ResultSet rs = stmt.executeQuery(sql);
69         while (rs.next()) {
70             mahasiswaList.add(new ModelMahasiswa(
71                 rs.getInt("id"),
72                 rs.getString("nama"),
73                 rs.getString("npm"),
74                 rs.getInt("semester"),
75                 rs.getFloat("ipk")
76             ));
77         }
78     } catch (SQLException e) {
79         e.printStackTrace();
80     }
81     return mahasiswaList;
82 }
83
84 public void updateMahasiswa(ModelMahasiswa mahasiswa) {
85     String sql = "UPDATE mahasiswa SET npm = ?, nama = ?, semester = ?, ipk = ? WHERE id = ?"; // Nama berlebih dihapus
86     try {
87         PreparedStatement pstmt = connection.prepareStatement(sql);
88         pstmt.setString(1, mahasiswa.getNpm());
89         pstmt.setString(2, mahasiswa.getNama());
90         pstmt.setInt(3, mahasiswa.getSemester());
91         pstmt.setFloat(4, mahasiswa.getIpk());
92         pstmt.setInt(5, mahasiswa.getId()); // Parameter kelima dihapus
93         pstmt.executeUpdate();
94     } catch (SQLException e) {
95         e.printStackTrace();
96     }
97 }
98
99 public void deleteMahasiswa(int id) {
100     String sql = "DELETE FROM mahasiswa WHERE id = ?";
101     try {
102         PreparedStatement pstmt = connection.prepareStatement(sql);
103         pstmt.setInt(1, id);
104         pstmt.executeUpdate();
105     } catch (SQLException e) {
106         e.printStackTrace();
107     }
}
```



- ModelMahasiswa.java

pom.xml [Pert3_50421223] x MahasiswaController.java x MahasiswaDAO.java x ModelMahasiswa.java x MahasiswaView.java x

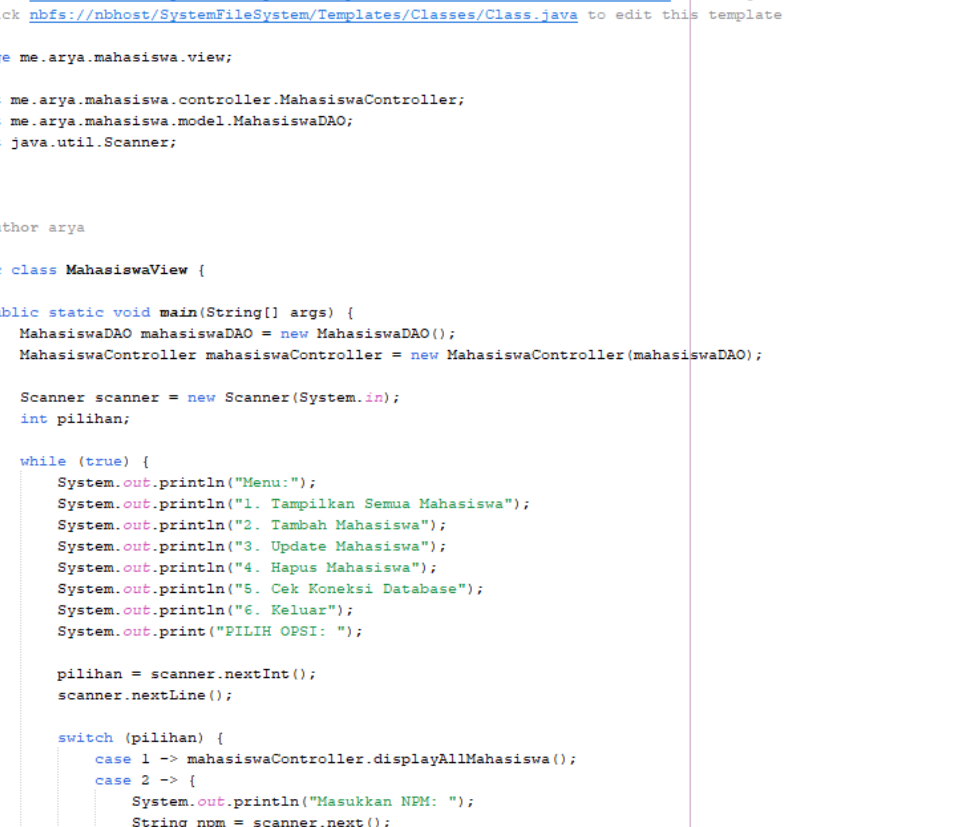
Source History

```
1 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 package me.arya.mahasiswa.model;
6
7 /**
8  *
9
10 @author arya*/
11 public class ModelMahasiswa {
12     private int id;
13     private String nama;
14     private String npm;
15     private int semester;
16     private float ipk;
17
18     public ModelMahasiswa(int id, String nama, String npm, int semester, float ipk) {
19         this.id = id;
20         this.nama = nama;
21         this.npm = npm;
22         this.semester = semester;
23         this.ipk = ipk;
24     }
25
26     public int getId() {
27         return id;
28     }
29
30     public void setId(int id) {
31         this.id = id;
32     }
33
34     public String getNama() {
35         return nama;
36     }
37
38     public void setNama(String nama) {
39         this.nama = nama;
40     }
41
42     public String getNpm() {
43         return npm;
```

```
41
42 [-] public String getNpm() {
43     return npm;
44 }
45
46 [-] public void setNpm(String npm) {
47     this.npm = npm;
48 }
49
50 [-] public int getSemester() {
51     return semester;
52 }
53
54 [-] public void setSemester(int semester) {
55     this.semester = semester;
56 }
57
58 [-] public float getIpk() {
59     return ipk;
60 }
61
62 [-] public void setIpk(float ipk) {
63     this.ipk = ipk;
64 }
65 }
```

 me.arya.mahasiswa.model.ModelMahasiswa >  setNpm >

- MahasiswaView.java



```
pom.xml [Pert3_50421223] x MahasiswaController.java x MahasiswaDAO.java x ModelMahasiswa.java x MahasiswaView.java x
Source History
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 me.arya.mahasiswa.view;
6
7 import me.arya.mahasiswa.controller.MahasiswaController;
8 import me.arya.mahasiswa.model.MahasiswaDAO;
9 import java.util.Scanner;
10
11 /**
12  *
13  * @author arya
14  */
15 public class MahasiswaView {
16
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
34             pilihan = scanner.nextInt();
35             scanner.nextLine();
36
37             switch (pilihan) {
38                 case 1 -> mahasiswaController.displayAllMahasiswa();
39                 case 2 -> {
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                     mahasiswaController.tambahMahasiswa(npm, nama, semester);
47                 }
48                 case 3 -> {
49                     System.out.println("Masukkan NPM: ");
50                     String npm = scanner.next();
51                     System.out.println("Masukkan Nama: ");
52                     String nama = scanner.next();
53                     System.out.println("Masukkan Semester: ");
54                     int semester = scanner.nextInt();
55                     mahasiswaController.updateMahasiswa(npm, nama, semester);
56                 }
57                 case 4 -> {
58                     System.out.println("Masukkan NPM: ");
59                     String npm = scanner.next();
60                     mahasiswaController.hapusMahasiswa(npm);
61                 }
62                 case 5 -> mahasiswaController.cekKoneksiDatabase();
63                 case 6 -> System.exit(0);
64             }
65         }
66     }
67 }
```

```

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     mahasiswaController.addMahasiswa(npm, nama, semester, ipk);
49 }
50
51 case 3 -> {
52     System.out.print("Masukkan ID Mahasiswa: ");
53     int id = scanner.nextInt();
54     scanner.nextLine();
55     System.out.println("Masukkan NPM: ");
56     String npmBaru = scanner.next();
57     System.out.println("Masukkan Nama: ");
58     String namaBaru = scanner.next();
59     System.out.println("Masukkan Semester: ");
60     int semesterBaru = scanner.nextInt();
61     System.out.println("Masukkan IPK: ");
62     float ipkBaru = scanner.nextFloat();
63     mahasiswaController.updateMahasiswa(id, npmBaru, namaBaru, semesterBaru, ipkBaru);
64 }
65
66 case 4 -> {
67     System.out.print("Masukkan ID Mahasiswa: ");
68     int idHapus = scanner.nextInt();
69     mahasiswaController.deleteMahasiswa(idHapus);
70     // Break ditambahkan
71 }
72
73 case 5 -> mahasiswaController.checkDatabaseConnection();
74 // Break ditambahkan
75
76 case 6 -> {
77     mahasiswaController.closeConnection();
78     System.out.println("Program selesai.");
79     return;
80 }
81
82 default -> System.out.println("Pilihan tidak valid.");

```


OUTPUT:

- Saat di run

```
me.arya.mahasiswa.view.MahasiswaView >
Output - Run (MahasiswaView) x
-----< me.arya.mahasiswa:Pert3_50421223 >---
[ ] Building Pert3_50421223 1.0-SNAPSHOT
    from pom.xml
    -----[ jar ]-----
    The artifact mysql:mysql-connector-java:jar:8.0.33 has b
[ ] --- resources:3.3.1:resources (default-resources) @ Pert
    skip non existing resourceDirectory C:\Users\Arya\Docume
[ ] --- compiler:3.11.0:compile (default-compile) @ Pert3_50
    Nothing to compile - all classes are up to date
[ ] --- exec:3.1.0:exec (default-cli) @ Pert3_50421223 ---
    Menu:
    1. Tampilkan Semua Mahasiswa
    2. Tambah Mahasiswa
    3. Update Mahasiswa
    4. Hapus Mahasiswa
    5. Cek Koneksi Database
    6. Keluar
    PILIH OPSI:
```

- Menambah Mahasiswa

```
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:
50421223
Masukkan Nama:
ARAP
Masukkan Semester:
7
Masukkan IPK:
4,00
Mahasiswa berhasil ditambahkan!
```

- Menampilkan Mahasiswa

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 : ARAP
NAMA : 50421223
SEMESTER : 7
IPK : 4.0

- Update Mahasiswa

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:
50421223
Masukkan Nama:
ARYA
Masukkan Semester:
5
Masukkan IPK:
1,00
Mahasiswa berhasil diperbarui!

- Hapus Mahasiswa

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: 2
Mahasiswa Berhasil Dihapus

- Cek Koneksi

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

- Keluar

Menu:

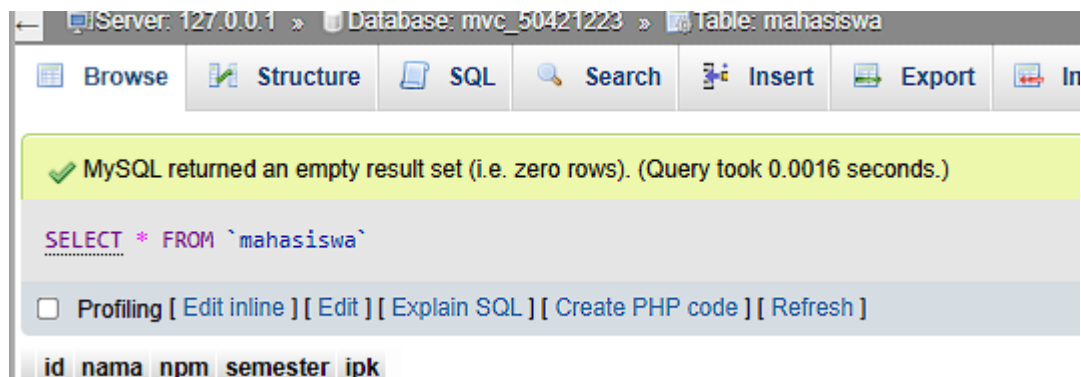
1. Tampilkan Semua Mahasiswa
 2. Tambah Mahasiswa
 3. Update Mahasiswa
 4. Hapus Mahasiswa
 5. Cek Koneksi Database
 6. Keluar
- PILIH OPSI: 6
Program selesai.

BUILD SUCCESS

Total time: 03:00 min
Finished at: 2024-10-28T14:39:33+07:00

- PHPmyadmin

- Saat baru dibuat



- Saat setelah dimasukkan data

```
SELECT * FROM `mahasiswa`
```

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

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

Options

			id	nama	npm	semester	ipk	
<input type="checkbox"/>	 Edit	 Copy	 Delete	3	50421223	ARYA	7	4

- Setelah data dihapus

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0007 seconds.)

```
SELECT * FROM `mahasiswa`
```

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

id	nama	npm	semester	ipk
----	------	-----	----------	-----