

## ACTIVITY PERTEMUAN 4

**NAMA** : Mohamad Afif Juniawan

**NPM** 50421831

**KELAS** : 4IA13

**MATERI** : Konsep Dasar Object Relational Mapping (ORM) Dan Framework Hibernate

**MATA PRAKTIKUM** : RPL2

---

(Screenshoot langkah-langkah sesuai video pembelajaran dan jelaskan dengan ringkas)



1. Pada komponen yang ada pada JFrame mahasiswa ada 4

- Label : berfungsi sebagai text judul
- Textfield : berfungsi sebagai sebuah area untuk menuliskan text yang

diinputkan

• Button : untuk memberikan perintah untuk memasukkan text yang sudah diisi  
kedalam database

- Table : berfungsi sebagai tempat untuk menampung inputan

1. 2. Screenshot codingan

MahasiswaController.java

```

1 package com.mahasiswa.controller;
2
3 import com.mahasiswa.model.hibernateUtil;
4 import com.mahasiswa.model.modelMahasiswa;
5 import java.util.List;
6 import org.hibernate.*;
7 import org.hibernate.query.Query;
8
9 public class mahasiswaController {
10     public void addMhs(modelMahasiswa mhs) {
11         Transaction trx = null;
12         try (Session session = hibernateUtil.getSessionFactory().openSession()) {
13             trx = session.beginTransaction();
14             session.save(mhs);
15             trx.commit();
16         } catch (Exception e) {}

```

```

17         if (trx != null) {
18             trx.rollback();
19         }
20         e.printStackTrace();
21     }
22 }
23
24 public void updateMhs(modelMahasiswa mhs) {
25     Transaction trx = null;
26     try (Session session = hibernateUtil.getSessionFactory().openSession()) {
27         trx = session.beginTransaction();
28         session.update(mhs);
29         trx.commit();
30     } catch (Exception e) {
31         if (trx != null) {
32             trx.rollback();
33         }
34         e.printStackTrace();
35     }
36 }
37
38 public void deleteMhs(int id) {
39     Transaction trx = null;
40
41     try (Session session = hibernateUtil.getSessionFactory().openSession()) {
42         trx = session.beginTransaction();
43
44         modelMahasiswa mhs = session.get(modelMahasiswa.class, id);
45         if (mhs != null) {
46             session.delete(mhs);
47             System.out.println("Berhasil dihapus");
48         }
49         trx.commit();
50     } catch (Exception e) {
51         if (trx != null) {
52             trx.rollback();
53         }
54         e.printStackTrace();
55     }
56 }
57
58 public List<modelMahasiswa> getAllMahasiswa() {
59     Transaction trx = null;
60     List<modelMahasiswa> listMhs = null;
61
62     try (Session session = hibernateUtil.getSessionFactory().openSession()) {
63         trx = session.beginTransaction();
64

```

```

65         Query<modelMahasiswa> query = session.createQuery("from modelMahasiswa", modelMahasiswa.class);
66         listMhs = query.list();
67
68         trx.commit();
69     } catch (Exception e) {
70         if (trx != null) {
71             trx.rollback();
72         }
73         e.printStackTrace();
74     }
75
76     return listMhs;
77 }
78 }

```

## modelMahasiswa.java

```

1  package com.mahasiswa.model;
2
3  import jakarta.persistence.*;
4
5  @Entity
6  @Table(name = "mahasiswa")
7  public class modelMahasiswa {
8      @Id
9      @GeneratedValue(strategy = GenerationType.IDENTITY)
10     private int id;
11
12     @Column(name = "npm", nullable = false, length = 8)
13     private String npm;
14
15     @Column(name = "nama", nullable = false, length = 50)
16     private String nama;
17
18     @Column(name = "semester")
19     private int semester;
20
21     @Column(name = "ipk")
22     private float ipk;
23
24     // Constructor kosong (default)
25     public modelMahasiswa() {}
26
27     // Constructor dengan semua parameter
28     public modelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
29         this.id = id;
30         this.npm = npm;
31         this.nama = nama;
32         this.semester = semester;
33         this.ipk = ipk;
34     }
35
36     public int getId() {
37         return id;
38     }
39
40     public void setId(int id) {
41         this.id = id;
42     }
43
44     public String getNpm() {
45         return npm;
46     }
47

```

```

48     public void setNpm(String npm) {
49         this.npm = npm;
50     }
51
52     public String getName() {
53         return nama;
54     }
55
56     public void setName(String nama) {
57         this.nama = nama;
58     }
59
60     public int getSemester() {
61         return semester;
62     }
63
64     public void setSemester(int semester) {
65         this.semester = semester;
66     }
67
68     public float getIpk() {
69         return ipk;
70     }
71
72     public void setIpk(float ipk) {
73         this.ipk = ipk;
74     }
75 }
76

```

#### modelTableMahasiswa.java

```

1  package com.mahasiswa.model;
2
3  import java.util.List;
4  import javax.swing.table.AbstractTableModel;
5
6  public class modelTableMahasiswa extends AbstractTableModel {
7
8      private List<modelMahasiswa> mahasiswaList;
9      private String[] columnNames = {"ID", "NPM", "NAMA", "SEMESTER", "IPK"};
10
11     public modelTableMahasiswa(List<modelMahasiswa> mahasiswaList) {
12         this.mahasiswaList = mahasiswaList;
13     }
14
15     @Override
16     public int getRowCount() {
17         return mahasiswaList.size();
18     }
19
20     @Override
21     public int getColumnCount() {
22         return columnNames.length;
23     }
24
25     @Override
26     public Object getValueAt(int rowIndex, int columnIndex) {
27         modelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);

```

```

29         switch (columnIndex) {
30             case 0:
31                 return mahasiswa.getId();
32             case 1:
33                 return mahasiswa.getNpm();
34             case 2:
35                 return mahasiswa.getNama();
36             case 3:
37                 return mahasiswa.getSemester();
38             case 4:
39                 return mahasiswa.getIpk();
40             default:
41                 return null;
42         }
43     }
44     @Override
45     public String getColumnName(int column) {
46         return columnNames[column];
47     }
48     @Override
49     public boolean isCellEditable(int rowIndex, int columnIndex) {
50         return false;
51     }
52 }
53
54 public void setMahasiswaList(List<modelMahasiswa> mahasiswaList) {
55     this.mahasiswaList = mahasiswaList;
56     fireTableDataChanged();
57 }
58 }

```

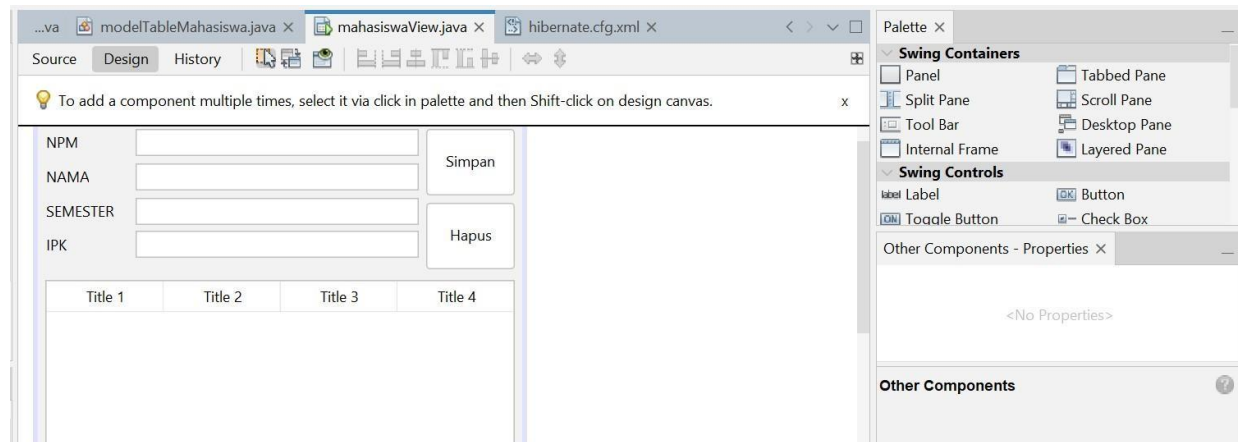
## hibernateUtil.java

```

1  package com.mahasiswa.model;
2
3  import org.hibernate.Session;
4  import org.hibernate.SessionFactory;
5  import org.hibernate.cfg.Configuration;
6
7  public class hibernateUtil {
8      private static SessionFactory sessionFactory;
9
10     static {
11         try {
12             sessionFactory = new Configuration().configure().buildSessionFactory();
13         } catch (Throwable ex) {
14             System.err.println("Initial SessionFactory creation failed" + ex);
15             throw new ExceptionInInitializerError(ex);
16         }
17     }
18
19     public static SessionFactory getSessionFactory() {
20         return sessionFactory;
21     }
22
23     public static void testConnection() {
24         try (Session session = sessionFactory.openSession()) {
25             System.out.println("Connection to the database was successful!");
26         } catch (Exception e) {
27             System.err.println("Failed to connect to the database");
28             e.printStackTrace();
29         }
30     }
31 }
32

```

## mahasiswaView.java



```
1 package com.mahasiswa.view;
2
3 import com.mahasiswa.model.modelTableMahasiswa;
4 import com.mahasiswa.model.modelMahasiswa;
5 import com.mahasiswa.controller.mahasiswaController;
6 import com.mahasiswa.model.hibernateUtil;
7 import java.util.List;
8 import javax.swing.*;
9
10 public class mahasiswaView extends JFrame {
11     private mahasiswaController controller;
12
13     public void loadMahasiswaTable() {
14         List<modelMahasiswa> listMahasiswa = controller.getAllMahasiswa();
15
16         modelTableMahasiswa tableModel = new modelTableMahasiswa(listMahasiswa);
17
18         dataTable.setModel(tableModel);
19     }
20
21     public mahasiswaView() {
22         initComponents();
23         controller = new mahasiswaController();
24         hibernateUtil.testConnection();
25         loadMahasiswaTable();
26     }
27
28     public void clearTextField() {
29         jTextField1.setText("");
30         jTextField2.setText("");
31         jTextField3.setText("");
32         jTextField4.setText("");
33     }
34 }
```



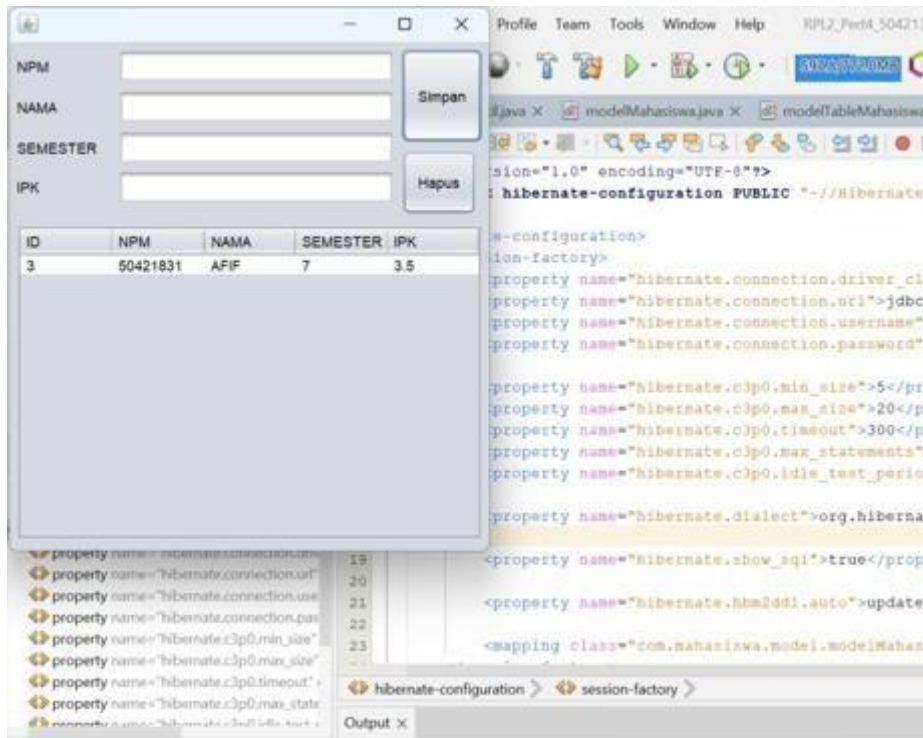
## Codingan button simpan

```
153 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
154     String npm = jTextField1.getText();  
155     String nama = jTextField2.getText();  
156     int semester = Integer.parseInt(jTextField3.getText());  
157     float ipk = Float.parseFloat(jTextField4.getText());  
158  
159     modelMahasiswa mahasiswa = new modelMahasiswa(0, npm, nama, semester, ipk);  
160  
161     System.out.println(mahasiswa.getIpk());  
162     System.out.println(mahasiswa.getNama());  
163     System.out.println(mahasiswa.getSemester());  
164     System.out.println(mahasiswa.getNpm());  
165  
166     controller.addMhs(mahasiswa);  
167     loadMahasiswaTable();  
168     clearTextField();  
169 }  
170  
171
```

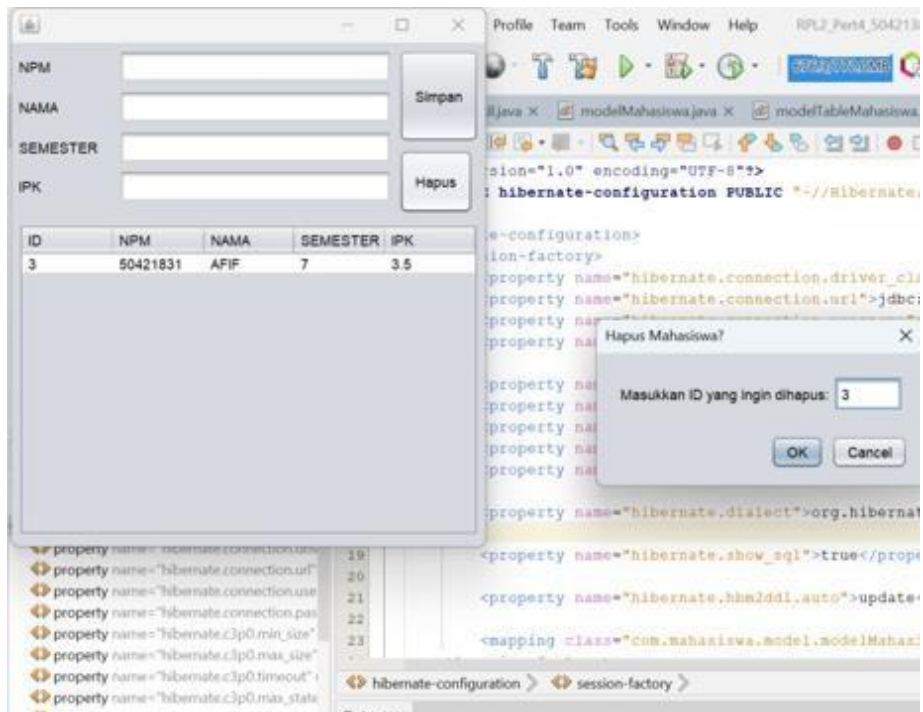
## Codingan Button Hapus

```
172 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
173     JTextField idField = new JTextField(5);  
174  
175     JPanel panel = new JPanel();  
176     panel.add(new JLabel("Masukkan ID yang ingin dihapus:"));  
177     panel.add(idField);  
178  
179     int result = JOptionPane.showConfirmDialog(null, panel, "Hapus Mahasiswa?", JOptionPane.OK_CANCEL_OPTION,  
180  
181     if (result == JOptionPane.OK_OPTION) {  
182         try {  
183             int id = Integer.parseInt(idField.getText());  
184             controller.deleteMhs(id);  
185             JOptionPane.showMessageDialog(null, "Data berhasil dihapus", "Sukses", JOptionPane.INFORMATION_MESSAGE);  
186             loadMahasiswaTable();  
187         } catch (NumberFormatException e) {  
188             JOptionPane.showMessageDialog(null, "ID harus berupa angka", "Error", JOptionPane.ERROR_MESSAGE);  
189         }  
190     }  
191 }
```

Output  
Saat input



Saat data dihapus





NPM

NAMA

SEMESTER

IPK

Simpan

Hapus

ID	NPM	NAMA	SEMESTER	IPK
----	-----	------	----------	-----

hibernate-configuration

```
<hibernate-configuration PUBLIC "-//Hibernate/HibernateConfiguration_3.0.dtd">
<session-factory>
<property name="hibernate.connection.driver_class">org.hibernate.jdbc.Driver
<property name="hibernate.connection.url">jdbc:hibernate.connection.username=hibernate.connection.password=hibernate
</property>
<property name="hibernate.c3p0.min_size">5</property>
<property name="hibernate.c3p0.max_size">20</property>
<property name="hibernate.c3p0.timeout">300</property>
<property name="hibernate.c3p0.max_statements">hibernate.c3p0.idle_test_period=hibernate
</property>
<property name="hibernate.dialect">org.hibernate.dialect.HibernateDialect
</property>
<property name="hibernate.show_sql">true</property>
<property name="hibernate.hbm2ddl.auto">update
</property>
<mapping class="com.mahasiswa.model.ModelMahasiswa">
</mapping>
</session-factory>
</hibernate-configuration>
```