

ACTIVITY PERTEMUAN 4

NAMA : FX Edrianto Putra Wijaya

NPM : 50421537

KELAS : 4IA13

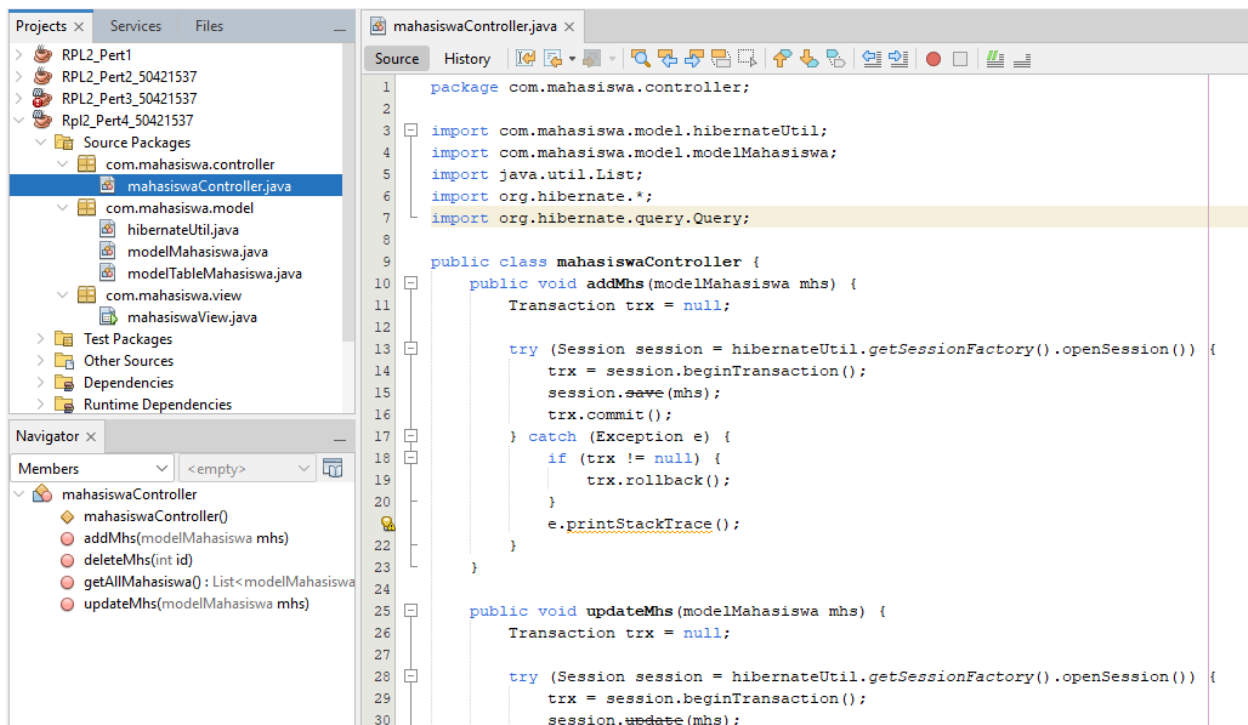
MATERI : hibernate

MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

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

1. A. Label : memberikan info tentang apa yang harus diinputkan ke textbox
B. Textbox : mendapatkan input dari user
C. Button : menjalankan programming yang telah diketik ketika diclick
D. Tabel : Menunjukkan isi database

2. Berikut ss dari program



```
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
13         try (Session session = hibernateUtil.getSessionFactory().openSession()) {
14             trx = session.beginTransaction();
15             session.save(mhs);
16             trx.commit();
17         } catch (Exception e) {
18             if (trx != null) {
19                 trx.rollback();
20             }
21             e.printStackTrace();
22         }
23     }
24
25     public void updateMhs(modelMahasiswa mhs) {
26         Transaction trx = null;
27
28         try (Session session = hibernateUtil.getSessionFactory().openSession()) {
29             trx = session.beginTransaction();
30             session.update(mhs);
```

```

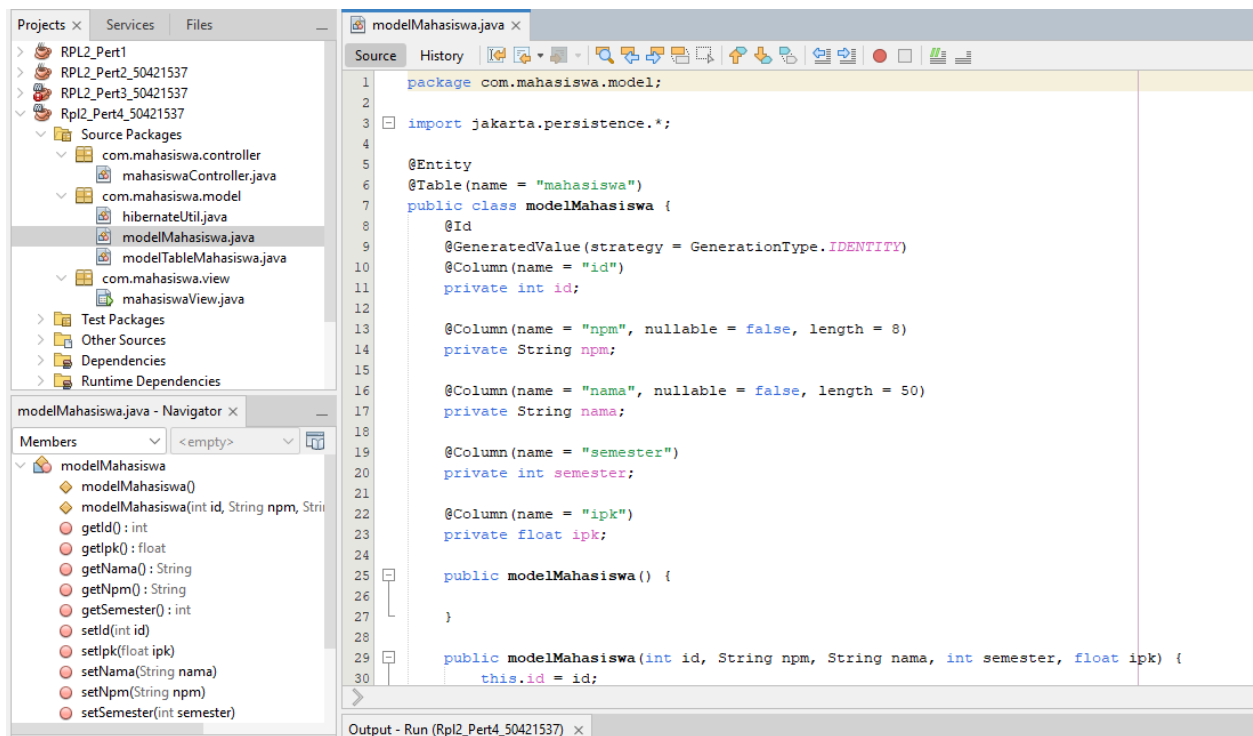
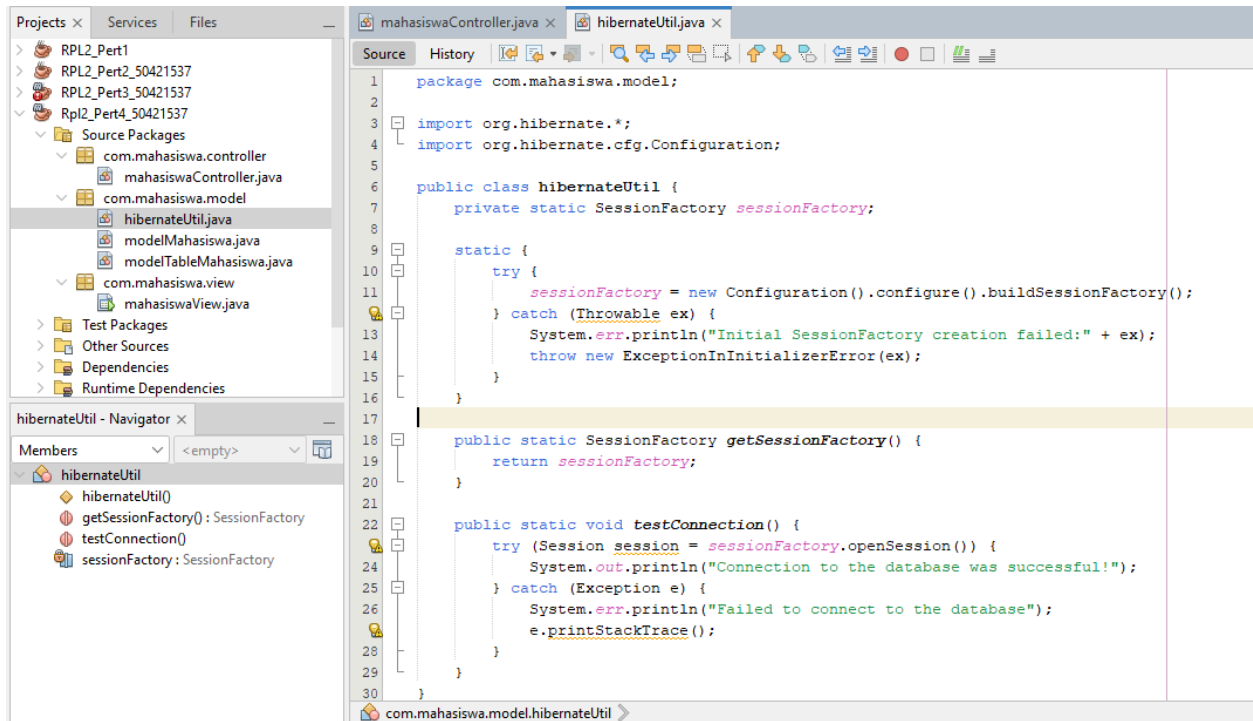
28     try (Session session = hibernateUtil.getSessionFactory().openSession()) {
29         trx = session.beginTransaction();
30         session.update(mhs);
31         trx.commit();
32     } catch (Exception e) {
33         if (trx != null) {
34             trx.rollback();
35         }
36         e.printStackTrace();
37     }
38 }
39
40 public void deleteMhs(int id) {
41     Transaction trx = null;
42
43     try (Session session = hibernateUtil.getSessionFactory().openSession()) {
44         trx = session.beginTransaction();
45         modelMahasiswa mhs = session.get(modelMahasiswa.class, id);
46         if (mhs != null) {
47             session.delete(mhs);
48             System.out.println("Berhasil dihapus");
49         }
50         trx.commit();
51     } catch (Exception e) {
52         if (trx != null) {
53             trx.rollback();
54         }
55         e.printStackTrace();
56     }
57 }

```

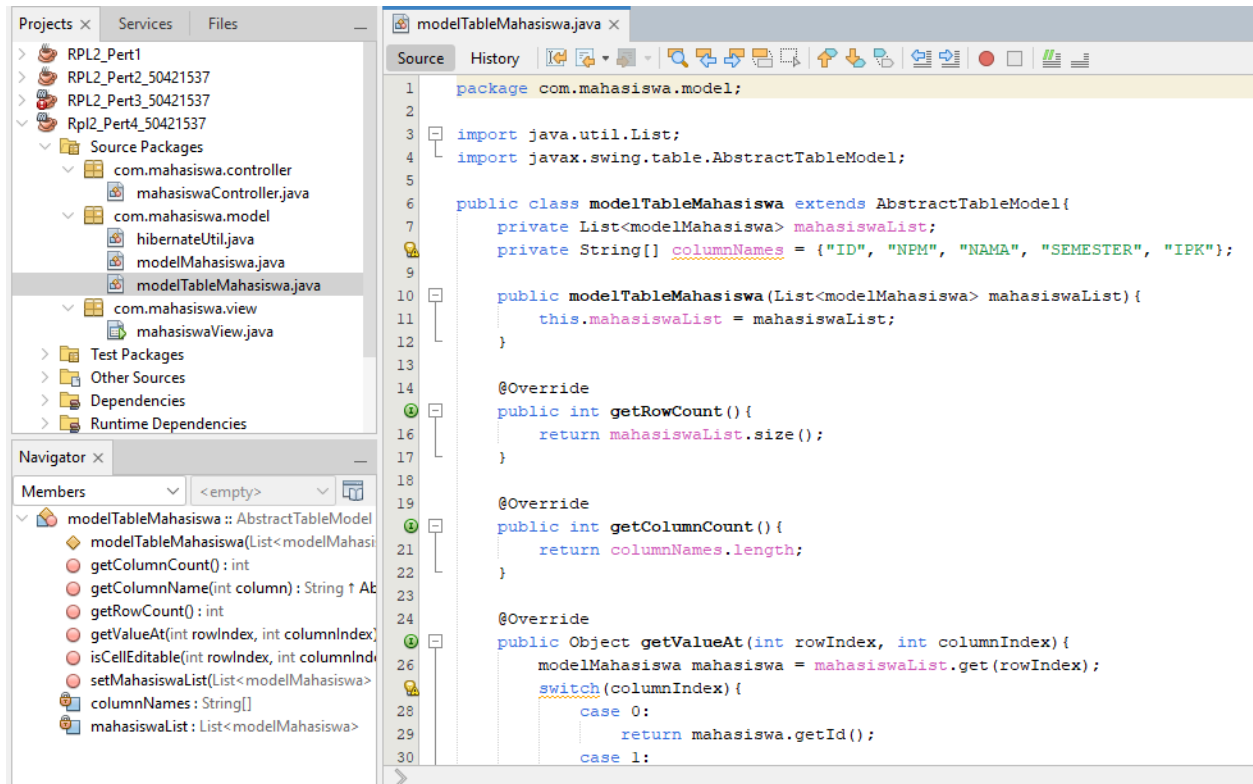
```

58
59 public List<modelMahasiswa> getAllMahasiswa() {
60     Transaction trx = null;
61     List<modelMahasiswa> listMhs = null;
62
63     try (Session session = hibernateUtil.getSessionFactory().openSession()) {
64         trx = session.beginTransaction();
65         Query<modelMahasiswa> query = session.createQuery("from modelMahasiswa", modelMahasiswa.class);
66         listMhs = query.list();
67         trx.commit();
68     } catch (Exception e) {
69         if (trx != null) {
70             trx.rollback();
71         }
72         e.printStackTrace();
73     }
74
75     return listMhs;
76 }
77
78 }

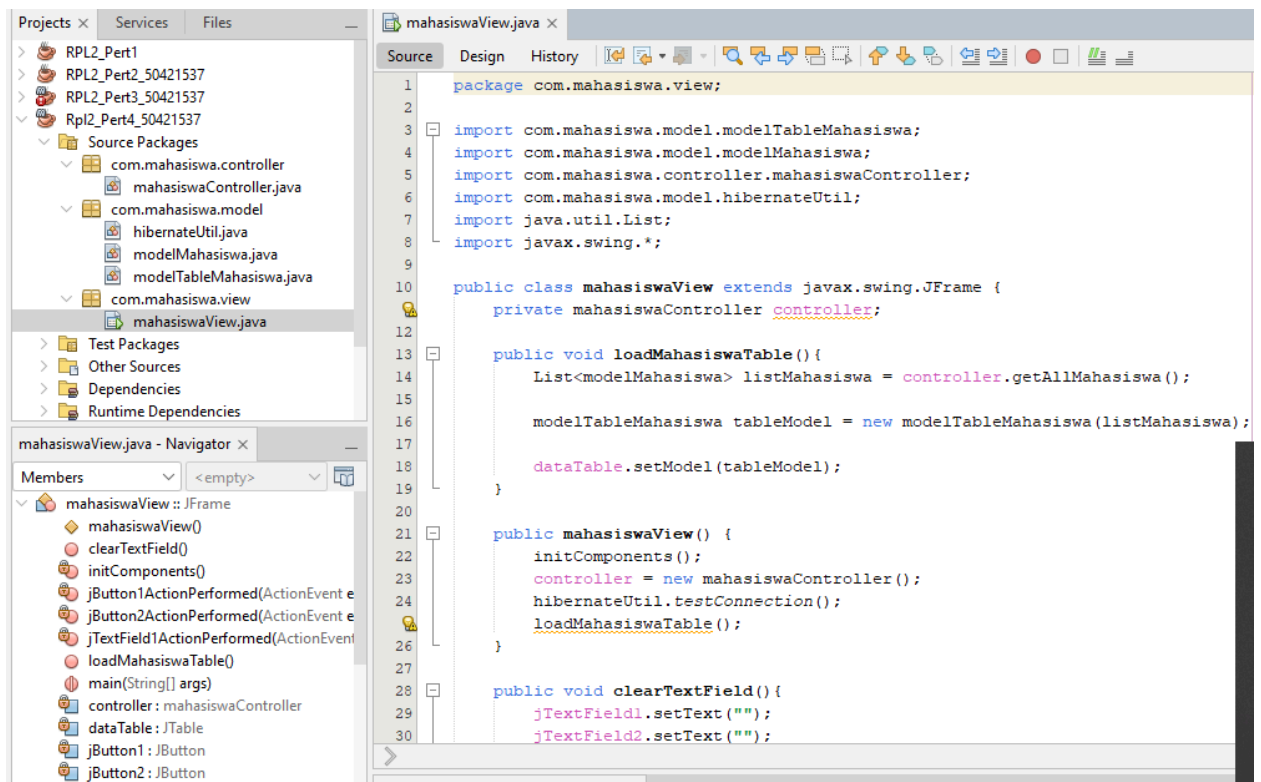
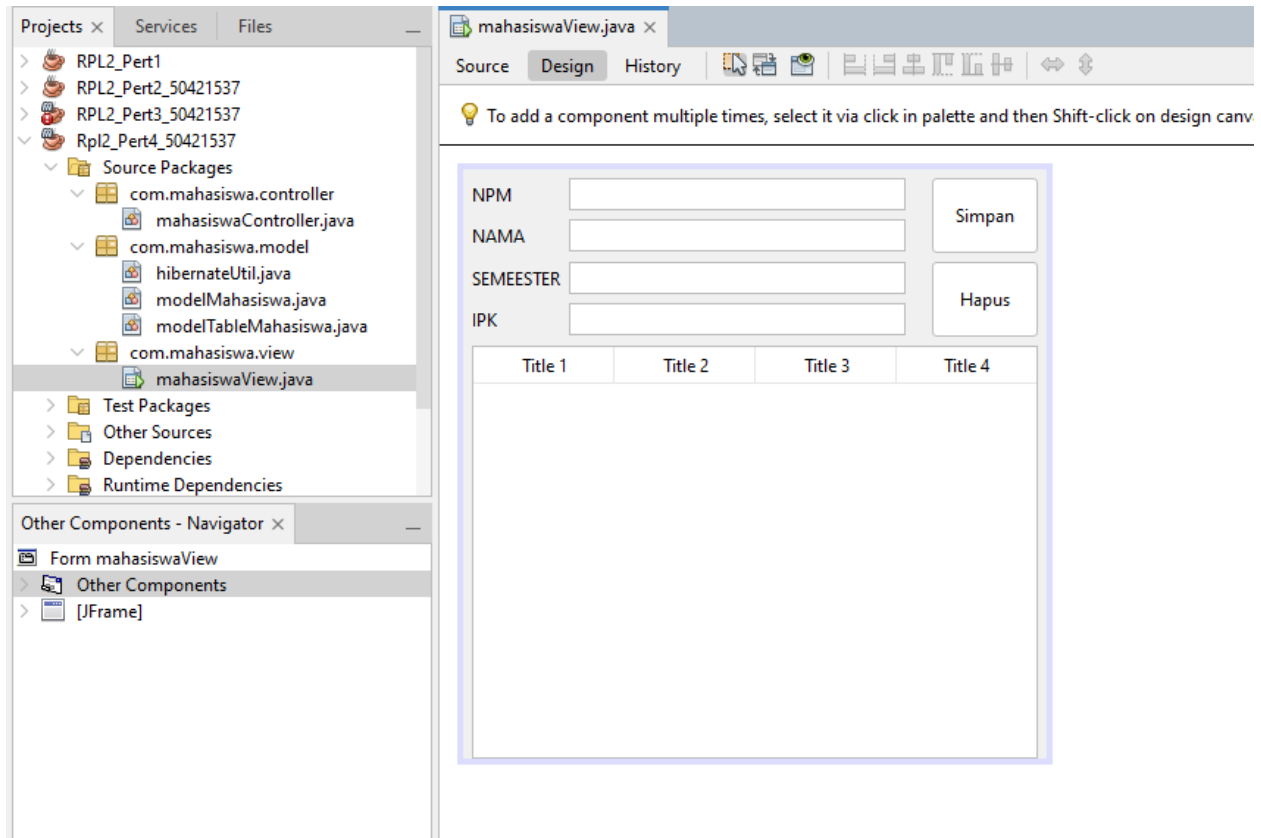
```



```
49  [ ] public void setNpm(String npm) {  
50      [ ]     this.npm = npm;  
51      [ ] }  
52  
53  [ ] public String getNama() {  
54      [ ]     return nama;  
55      [ ] }  
56  
57  [ ] public void setNama(String nama) {  
58      [ ]     this.nama = nama;  
59      [ ] }  
60  
61  [ ] public int getSemester() {  
62      [ ]     return semester;  
63      [ ] }  
64  
65  [ ] public void setSemester(int semester) {  
66      [ ]     this.semester = semester;  
67      [ ] }  
68  
69  [ ] public float getIpk() {  
70      [ ]     return ipk;  
71      [ ] }  
72  
73  [ ] public void setIpk(float ipk) {  
74      [ ]     this.ipk = ipk;  
75      [ ] }  
76  }  
77
```



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



```
mahasiswaView.java x
Source Design History
170
171 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
172     String npm = jTextField1.getText();
173     String nama = jTextField2.getText();
174     int semester = Integer.parseInt(jTextField3.getText());
175     float ipk = Float.parseFloat(jTextField4.getText());
176     modelMahasiswa mahasiswa = new modelMahasiswa(0, npm, nama, semester, ipk);
177     System.out.println(mahasiswa.getIpk());
178     System.out.println(mahasiswa.getNama());
179     System.out.println(mahasiswa.getSemester());
180     System.out.println(mahasiswa.getNpm());
181
182     controller.addMhs(mahasiswa);
183     loadMahasiswaTable();
184     clearTextField();
185 }
186
187 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
188     JTextField idField = new JTextField(5);
189
190     JPanel panel = new JPanel();
191     panel.add(new JLabel("Masukan ID yang ingin dihapus :"));
192     panel.add(idField);
193
194     int result = JOptionPane.showConfirmDialog(null, panel, "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION, JOptionPane.WARNING_MESSAGE);
195
196     if(result == JOptionPane.OK_OPTION){
197         try{
198             int id = Integer.parseInt(idField.getText());
```