

ACTIVITY PERTEMUAN 4

NAMA : Leonhard Dominikus Adiarsa Fernandez

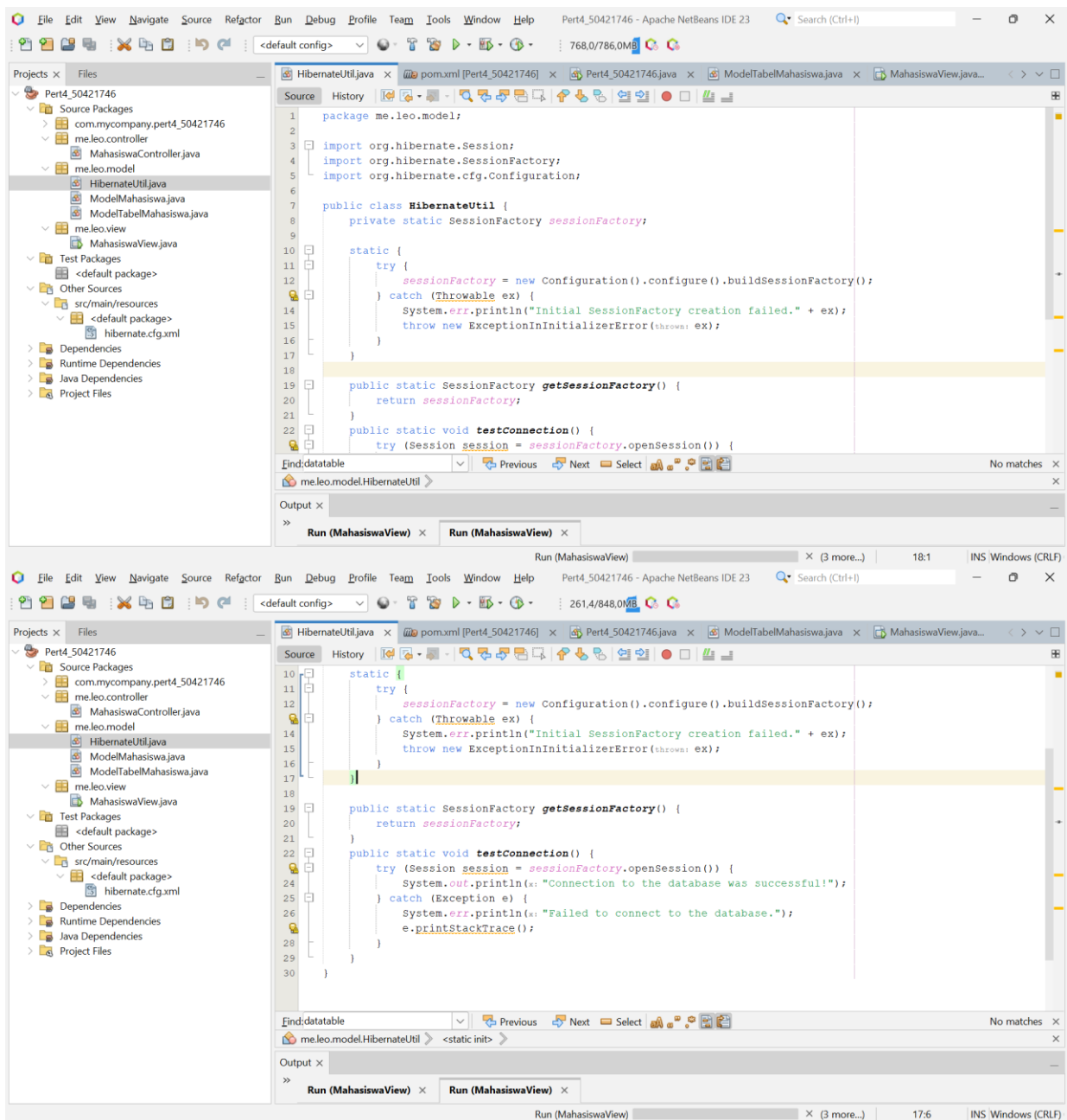
NPM : 50421746

KELAS : 3IA14

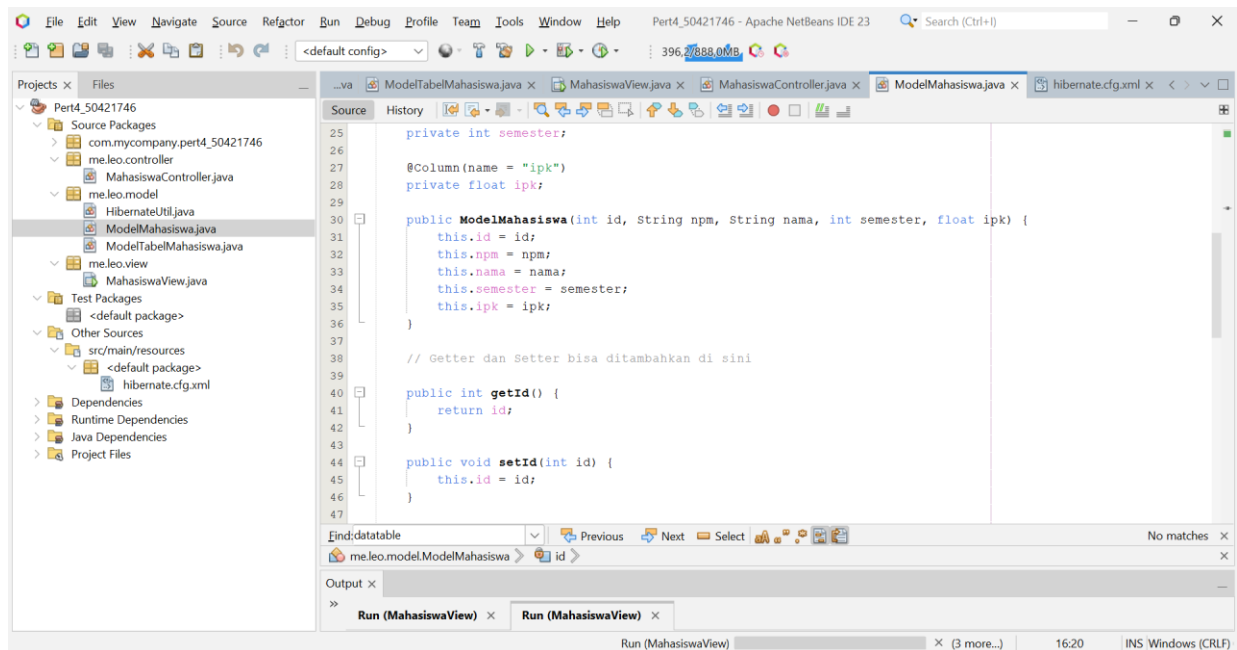
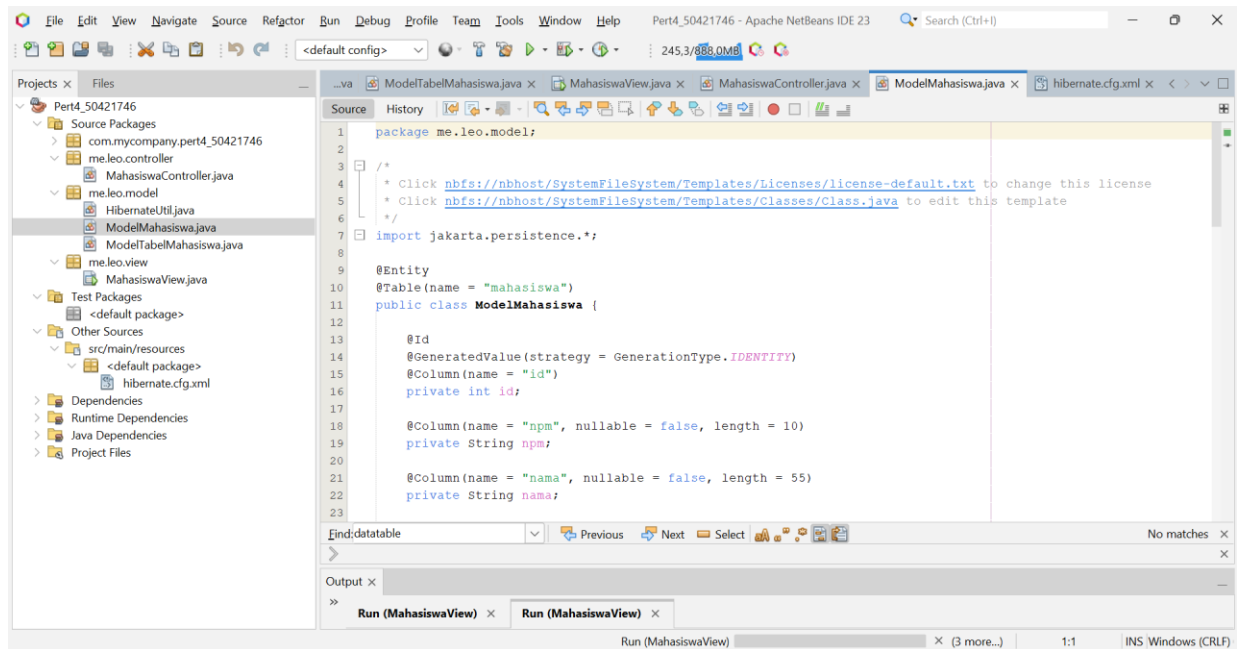
MATERI :

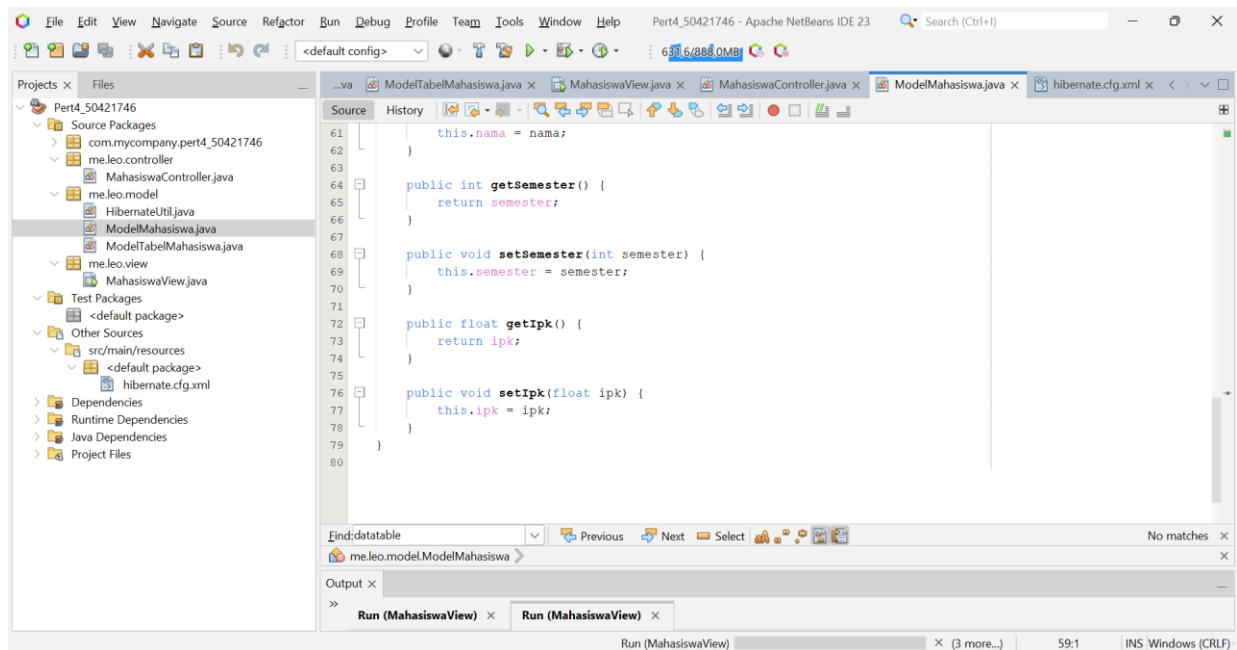
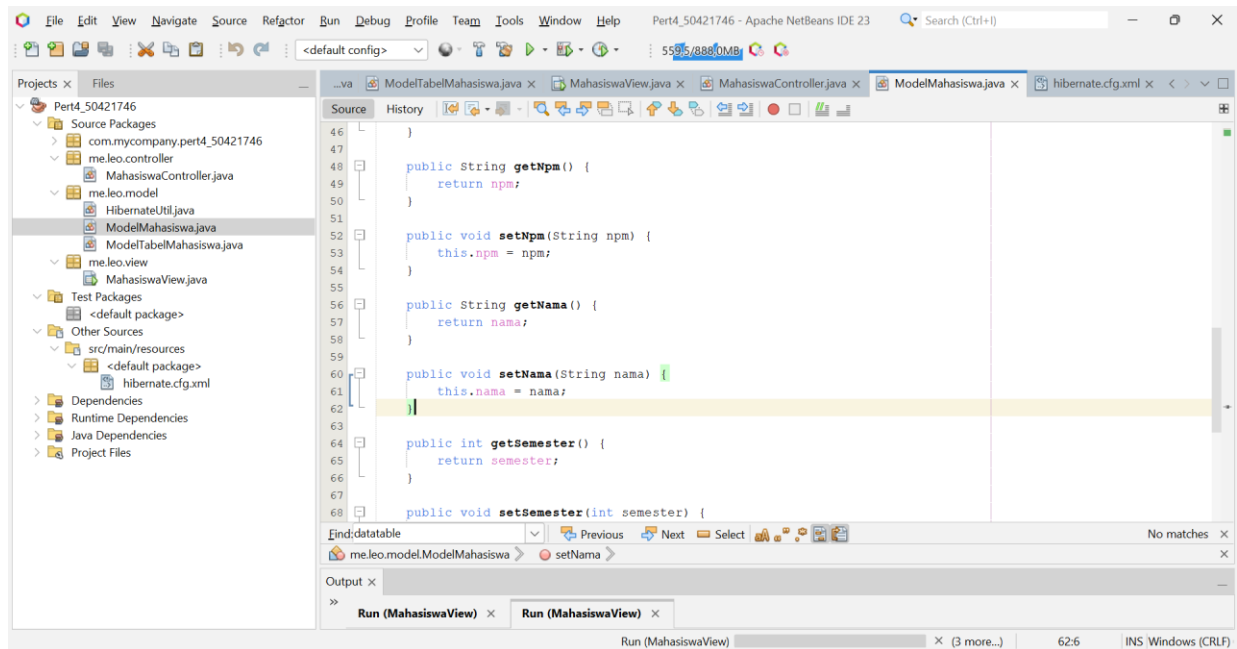
MATA PRAKTIKUM : RPL 2

HibernateUtil.java

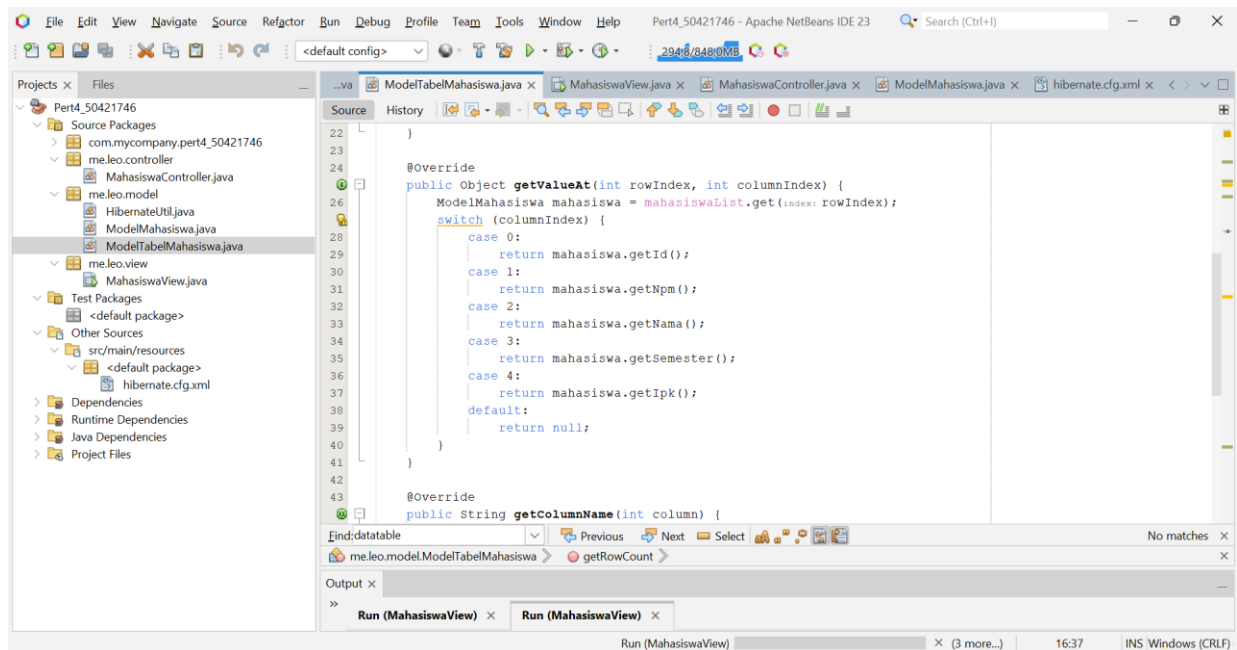
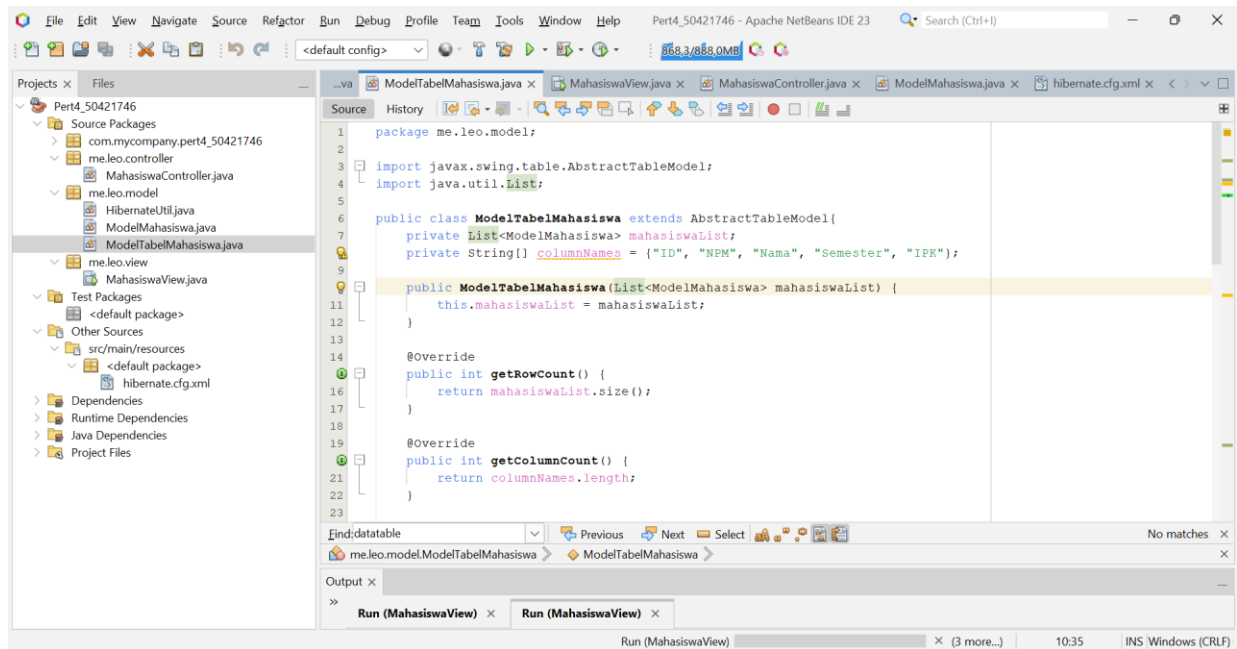


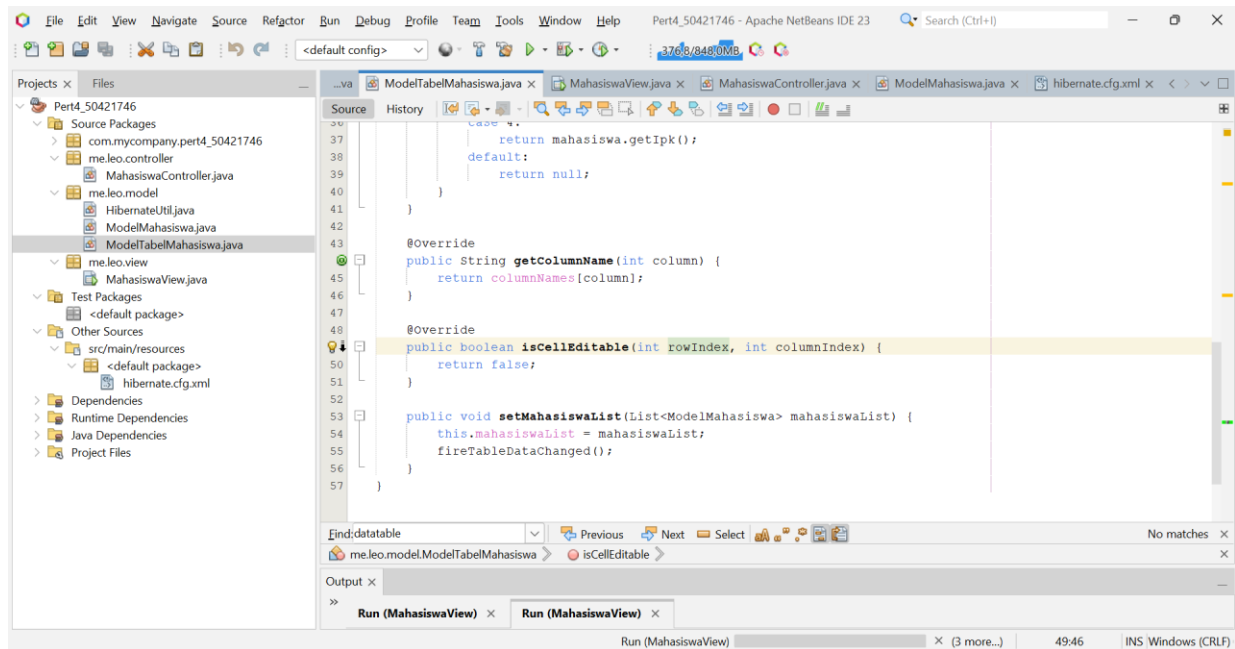
ModelMahasiswa.java





ModelMahasiswa.java





MahasiswaController.java

```
package me.leo.controller;
```

```
import me.leo.model.HibernateUtil;
```

```
import me.leo.model.ModelMahasiswa;
```

```
import java.util.List;
```

```
import org.hibernate.Session;
```

```
import org.hibernate.Transaction;
```

```
import org.hibernate.query.Query;
```

```
public class MahasiswaController {
```

```
    public void addMhs(ModelMahasiswa mhs) {
```

```
        Transaction trx = null;
```

```
        try (Session session = HibernateUtil.getSessionFactory().openSession()) {
```

```
            trx = session.beginTransaction();
```

```
            session.save(mhs);
```

```
            trx.commit();
```

```
        } catch (Exception e) {
```

```
            if (trx != null) trx.rollback();
```

```

        e.printStackTrace();
    }
}

public void updateMhs(ModelMahasiswa mhs) {
    Transaction trx = null;

    try (Session session = HibernateUtil.getSessionFactory().openSession()) {
        trx = session.beginTransaction();

        session.update(mhs);

        trx.commit();
    } catch (Exception e) {
        if (trx != null) trx.rollback();

        e.printStackTrace();
    }
}

public void deleteMhs(int id) {
    Transaction trx = null;

    try (Session session = HibernateUtil.getSessionFactory().openSession()) {
        trx = session.beginTransaction();

        ModelMahasiswa mhs = session.get(ModelMahasiswa.class, id);

        if(mhs != null) {
            session.delete(mhs);

            System.out.println("Berhasil hapus");
        }

        trx.commit();
    } catch (Exception e) {
        if (trx != null) trx.rollback();

        e.printStackTrace();
    }
}
}

```

```

public List<ModelMahasiswa> getAllMahasiswa() {

    Transaction trx = null;

    List<ModelMahasiswa> listMhs = null;

    try (Session session = HibernateUtil.getSessionFactory().openSession()) {

        trx = session.beginTransaction();

        Query<ModelMahasiswa> query = session.createQuery("from ModelMahasiswa",
ModelMahasiswa.class);

        listMhs = query.list();

        trx.commit();

    } catch (Exception e) {

        if (trx != null) trx.rollback();

        e.printStackTrace();

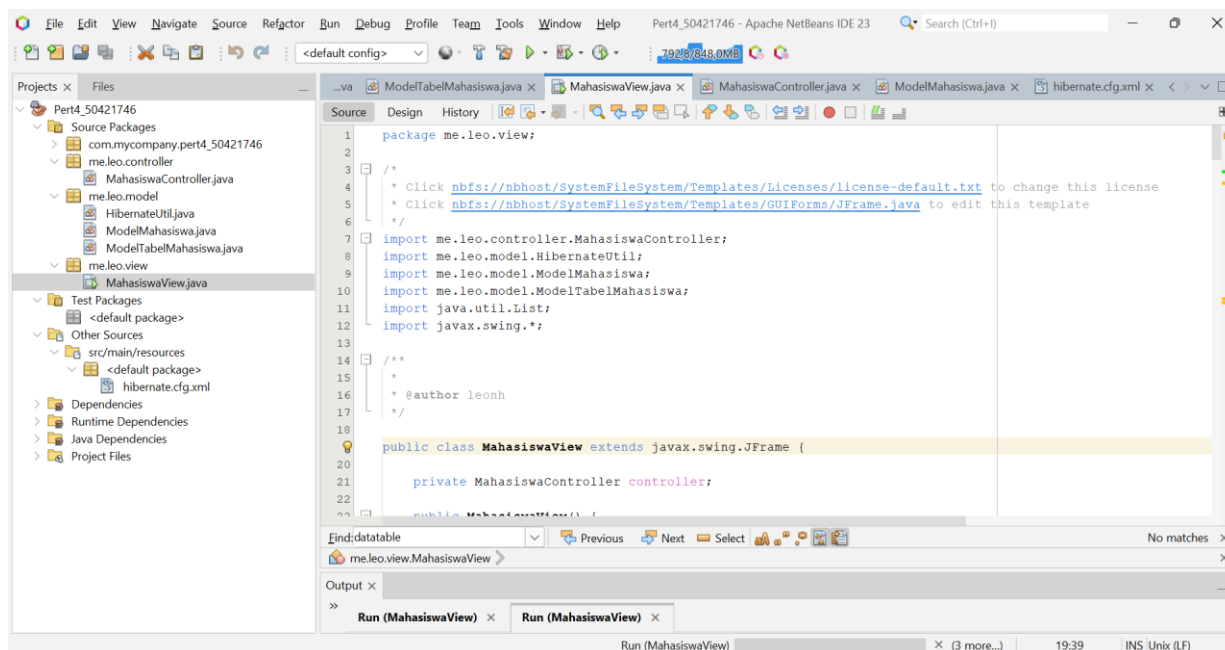
    }

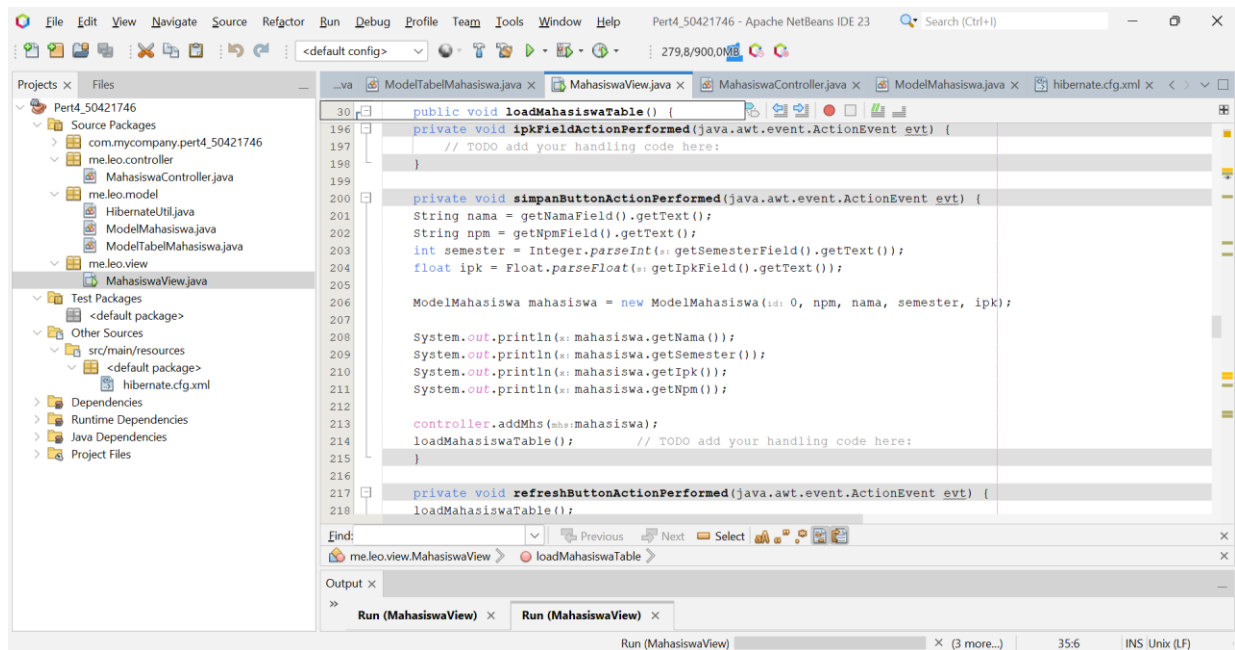
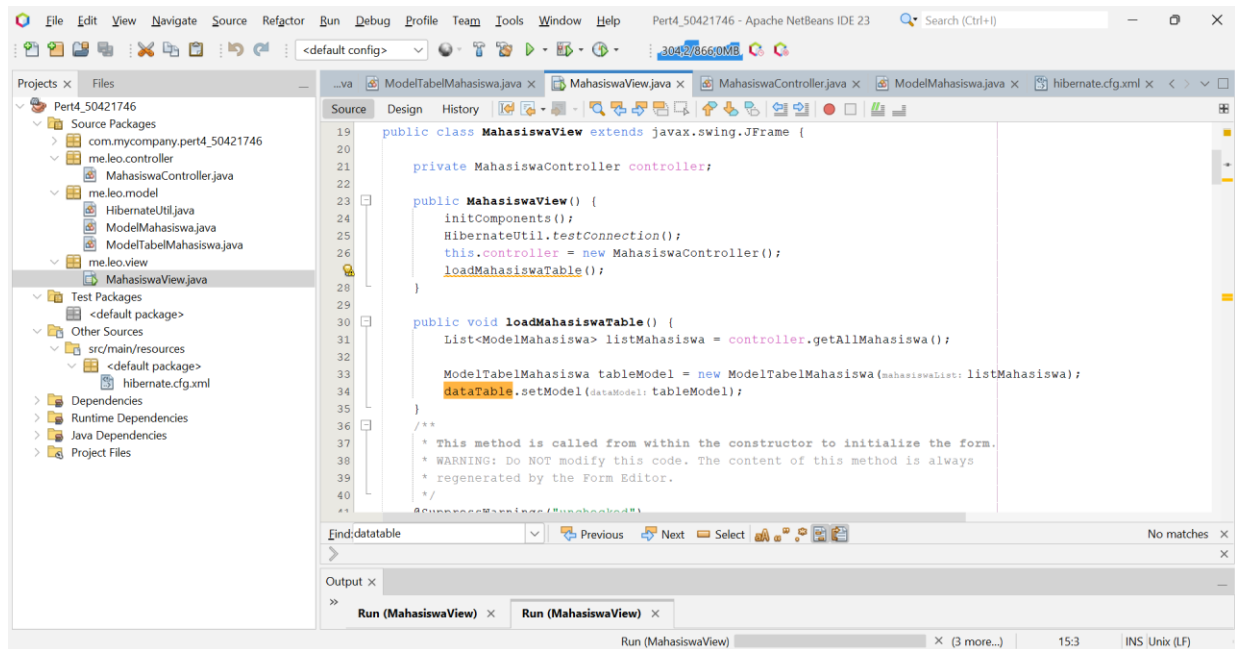
    return listMhs;

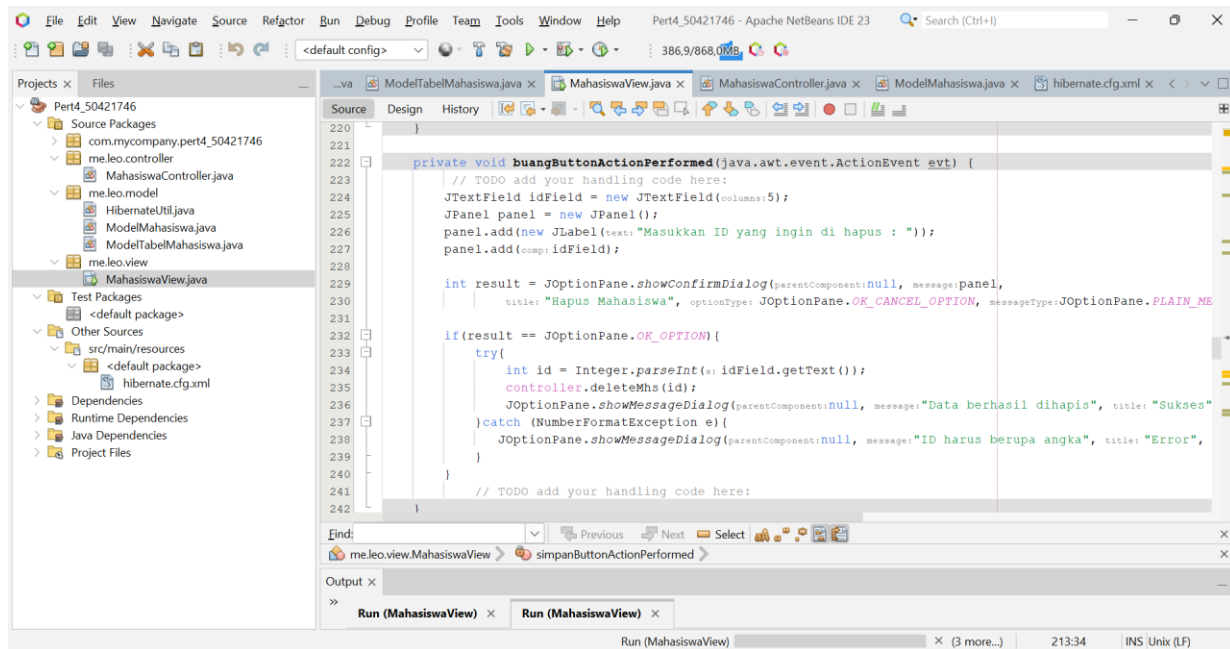
}
}

```

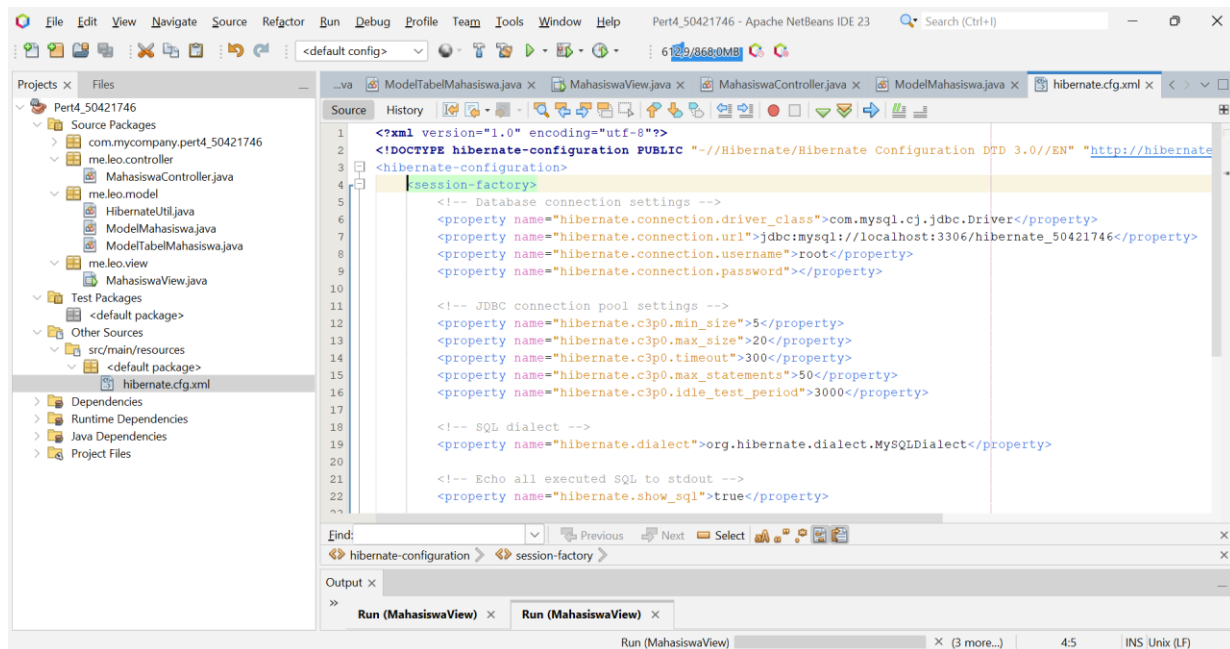
MahasiswaView.java

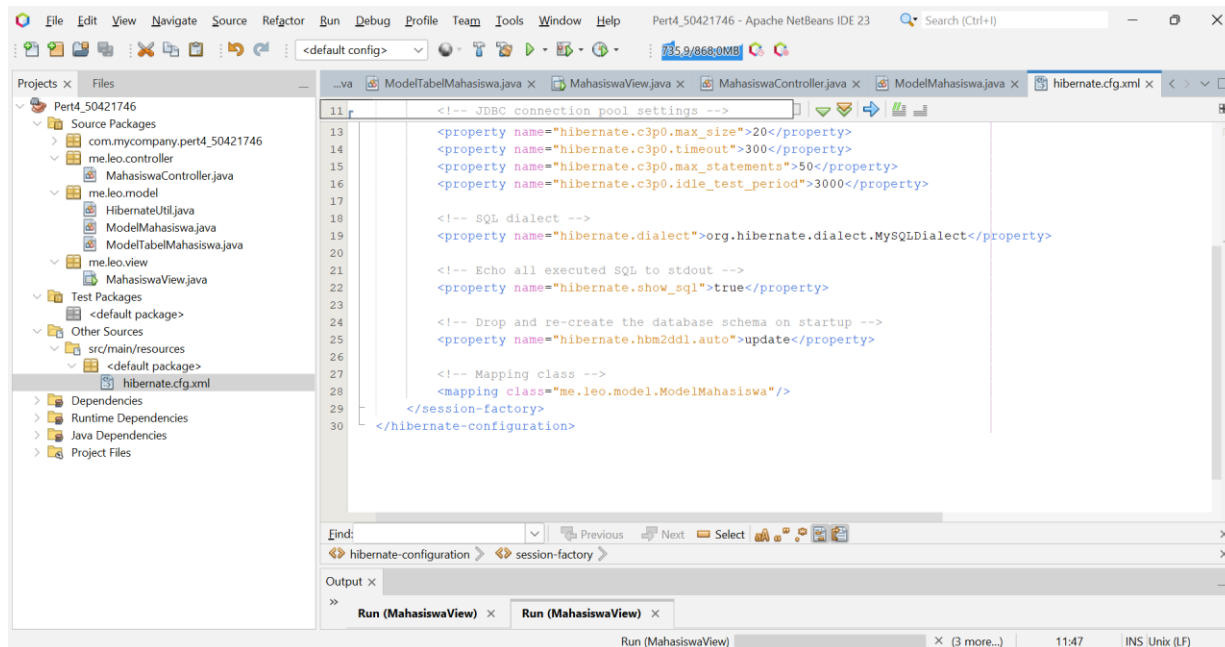






Hibernate.cfg.xml





Error:

Connection to the database was successful!

Hibernate: select mm1_0.id,mm1_0.ipk,mm1_0.nama,mm1_0.npm,mm1_0.semester from mahasiswa mm1_0

Exception in thread "AWT-EventQueue-0" java.lang.IllegalStateException:
org.hibernate.resource.jdbc.internal.LogicalConnectionManagedImpl@7c625966 is closed

at
org.hibernate.resource.jdbc.internal.AbstractLogicalConnectionImplementor.errorIfClosed(AbstractLogicalConnectionImplementor.java:37)

at
org.hibernate.resource.jdbc.internal.AbstractLogicalConnectionImplementor.rollback(AbstractLogicalConnectionImplementor.java:130)

at
org.hibernate.resource.transaction.backend.jdbc.internal.JdbcResourceLocalTransactionCoordinatorImpl\$TransactionDriverControlImpl.rollback(JdbcResourceLocalTransactionCoordinatorImpl.java:289)

at org.hibernate.engine.transaction.internal.TransactionImpl.rollback(TransactionImpl.java:142)

at me.leo.controller.MahasiswaController.getAllMahasiswa(MahasiswaController.java:61)

at me.leo.view.MahasiswaView.loadMahasiswaTable(MahasiswaView.java:31)

at me.leo.view.MahasiswaView.<init>(MahasiswaView.java:27)

at me.leo.view.MahasiswaView\$7.run(MahasiswaView.java:274)

at java.desktop/java.awt.event.InvocationEvent.dispatch(InvocationEvent.java:318)

at java.desktop/java.awt.EventQueue.dispatchEventImpl(EventQueue.java:773)

at java.desktop/java.awt.EventQueue\$4.run(EventQueue.java:720)

at java.desktop/java.awt.EventQueue\$4.run(EventQueue.java:714)

```

        at java.base/java.security.AccessController.doPrivileged(AccessController.java:400)

        at
java.base/java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:87)

        at java.desktop/java.awt.EventQueue.dispatchEvent(EventQueue.java:742)
        at java.desktop/java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:203)
        at java.desktop/java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:124)
        at java.desktop/java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:113)
        at java.desktop/java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:109)
        at java.desktop/java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
        at java.desktop/java.awt.EventDispatchThread.run(EventDispatchThread.java:90)

```

Full code MahasiswaView.java

```

package me.leo.view;

/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/GUIForms/JFrame.java to edit this template
 */

import me.leo.controller.MahasiswaController;
import me.leo.model.HibernateUtil;
import me.leo.model.ModelMahasiswa;
import me.leo.model.ModelTabelMahasiswa;
import java.util.List;
import javax.swing.*;

/**
 *
 * @author leonh
 */

public class MahasiswaView extends javax.swing.JFrame {

```

```

private MahasiswaController controller;

public MahasiswaView() {
    initComponents();
    HibernateUtil.testConnection();
    this.controller = new MahasiswaController();
    loadMahasiswaTable();
}

public void loadMahasiswaTable() {
    List<ModelMahasiswa> listMahasiswa = controller.getAllMahasiswa();

    ModelTabelMahasiswa tableModel = new ModelTabelMahasiswa(listMahasiswa);
    dataTable.setModel(tableModel);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jLabel1 = new javax.swing.JLabel();
    jLabel2 = new javax.swing.JLabel();
    jLabel3 = new javax.swing.JLabel();
    jLabel4 = new javax.swing.JLabel();
    npmField = new javax.swing.JTextField();
    namaField = new javax.swing.JTextField();
    semesterField = new javax.swing.JTextField();

```

```
ipkField = new javax.swing.JTextField();

simpanButton = new javax.swing.JButton();

uangButton = new javax.swing.JButton();

refreshButton = new javax.swing.JButton();

jScrollPane1 = new javax.swing.JScrollPane();

dataTable = new javax.swing.JTable();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);


jLabel1.setText("NPM");


jLabel2.setText("Nama");


jLabel3.setText("Semester");


jLabel4.setText("IPK");


npmField.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        npmFieldActionPerformed(evt);

    }

});


namaField.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        namaFieldActionPerformed(evt);

    }

});


ipkField.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        ipkFieldActionPerformed(evt);

    }

});
```

```

    }
});

simpanButton.setText("Simpan");

simpanButton.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        simpanButtonActionPerformed(evt);

    }

});

uangButton.setText("Uang");

uangButton.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        uangButtonActionPerformed(evt);

    }

});

refreshButton.setText("Refresh");

refreshButton.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(java.awt.event.ActionEvent evt) {

        refreshButtonActionPerformed(evt);

    }

});

dataTable.setModel(new javax.swing.table.DefaultTableModel(

    new Object [][] {

        { null, null, null, null },

        { null, null, null, null },

        { null, null, null, null },

        { null, null, null, null }

    },

    new String [] {

```

```
"Title 1", "Title 2", "Title 3", "Title 4"
    }
));

jScrollPane1.setViewportViewView(dataTable);


javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);

layout.setHorizontalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()

                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()

                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .add(layout.createSequentialGroup()

                                    .addGap(35, 35, 35)

                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addComponent(jLabel1)
                                        .addComponent(jLabel2)
                                        .addComponent(jLabel3)
                                        .addComponent(jLabel4))
                                    .addGap(58, 58, 58))
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

                                    .addContainerGap()

                                    .addComponent(simpanButton)

                                    .addGap(18, 18, 18)))
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .add(layout.createSequentialGroup()

                                    .addComponent(buangButton)

                                    .addGap(18, 18, 18)

                                    .addComponent(refreshButton))
                                .addComponent(ipkField, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)
```

```

        .addComponent(semesterField, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(namaField, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)

        .addComponent(npmField, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE)))

    .addGroup(layout.createSequentialGroup())

    .addContainerGap()

    .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 375,
javax.swing.GroupLayout.PREFERRED_SIZE)))

    .addContainerGap(19, Short.MAX_VALUE))

);

layout.setVerticalGroup(

    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addGroup(layout.createSequentialGroup())

    .addGap(48, 48, 48)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel1)

    .addComponent(npmField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGap(18, 18, 18)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel2)

    .addComponent(namaField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGap(18, 18, 18)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel3)

    .addComponent(semesterField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGap(18, 18, 18)

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

    .addComponent(jLabel4)

    .addComponent(ipkField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))

    .addGap(29, 29, 29)

```



```

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(simpanButton, javax.swing.GroupLayout.PREFERRED_SIZE, 23,
                javax.swing.GroupLayout.PREFERRED_SIZE)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                .addComponent(buangButton)

                .addComponent(refreshButton)))

        .addGap(18, 18, 18)

        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 275,
            javax.swing.GroupLayout.PREFERRED_SIZE)

        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))

    );

    pack();
} // </editor-fold>

```

```

private void npmFieldActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void namaFieldActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void ipkFieldActionPerformed(java.awt.event.ActionEvent evt) {

    // TODO add your handling code here:

}

```

```

private void simpanButtonActionPerformed(java.awt.event.ActionEvent evt) {

    String nama = getNamaField().getText();

    String npm = getNpmField().getText();

    int semester = Integer.parseInt(getSemesterField().getText());

    float ipk = Float.parseFloat(getIpkField().getText());
}

```

```

ModelMahasiswa mahasiswa = new ModelMahasiswa(0, npm, nama, semester, ipk);

System.out.println(mahasiswa.getNama());
System.out.println(mahasiswa.getSemester());
System.out.println(mahasiswa.getIpk());
System.out.println(mahasiswa.getNpm());

controller.addMhs(mahasiswa);

loadMahasiswaTable();    // TODO add your handling code here:
}

private void refreshButtonActionPerformed(java.awt.event.ActionEvent evt) {
loadMahasiswaTable();

    // TODO add your handling code here:
}

private void buangButtonActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:

    JTextField idField = new JTextField(5);

    JPanel panel = new JPanel();

    panel.add(new JLabel("Masukkan ID yang ingin di hapus : "));

    panel.add(idField);

    int result = JOptionPane.showConfirmDialog(null, panel,

        "Hapus Mahasiswa", JOptionPane.OK_CANCEL_OPTION, JOptionPane.PLAIN_MESSAGE);

    if(result == JOptionPane.OK_OPTION){

        try{

            int id = Integer.parseInt(idField.getText());

            controller.deleteMhs(id);

            JOptionPane.showMessageDialog(null, "Data berhasil dihapus", "Sukses",
JOptionPane.INFORMATION_MESSAGE);

```

```

    } catch (NumberFormatException e){

        JOptionPane.showMessageDialog(null, "ID harus berupa angka", "Error",
JOptionPane.ERROR_MESSAGE);

    }

}

// TODO add your handling code here:

}

/**

 * @param args the command line arguments

 */

public static void main(String args[]) {

    /* Set the Nimbus look and feel */

    //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

    /* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

    * For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

    */

    try {

        for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {

            if ("Nimbus".equals(info.getName())) {

                javax.swing.UIManager.setLookAndFeel(info.getClassName());

                break;

            }

        }

    } catch (ClassNotFoundException ex) {

        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);

    } catch (InstantiationException ex) {

        java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);

    } catch (IllegalAccessException ex) {

```

```
java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
```

```
java.util.logging.Logger.getLogger(MahasiswaView.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
    }
```

```
//</editor-fold>
```

```
/* Create and display the form */
```

```
java.awt.EventQueue.invokeLater(new Runnable() {
```

```
    public void run() {
```

```
        new MahasiswaView().setVisible(true);
```

```
    }
```

```
});
```

```
}
```

```
// Variables declaration - do not modify
```

```
private javax.swing.JButton buangButton;
```

```
private javax.swing.JTable dataTable;
```

```
private javax.swing.JTextField ipkField;
```

```
private javax.swing.JLabel jLabel1;
```

```
private javax.swing.JLabel jLabel2;
```

```
private javax.swing.JLabel jLabel3;
```

```
private javax.swing.JLabel jLabel4;
```

```
private javax.swing.JScrollPane jScrollPane1;
```

```
private javax.swing.JTextField namaField;
```

```
private javax.swing.JTextField npmField;
```

```
private javax.swing.JButton refreshButton;
```

```
private javax.swing.JTextField semesterField;
```

```
private javax.swing.JButton simpanButton;
```

```
// End of variables declaration
```

```
public MahasiswaController getController() {  
    return controller;  
}
```

```
public void setController(MahasiswaController controller) {  
    this.controller = controller;  
}
```

```
public JButton getBuangButton() {  
    return buangButton;  
}
```

```
public void setBuangButton(JButton buangButton) {  
    this.buangButton = buangButton;  
}
```

```
public JTable getDataTable() {  
    return dataTable;  
}
```

```
public void setDataTable(JTable dataTable) {  
    this.dataTable = dataTable;  
}
```

```
public JTextField getIpkField() {  
    return ipkField;  
}
```

```
public void setIpkField(JTextField ipkField) {  
    this.ipkField = ipkField;  
}
```

```
public JLabel getjLabel1() {  
    return jLabel1;  
}
```

```
public void setjLabel1(JLabel jLabel1) {  
    this.jLabel1 = jLabel1;  
}
```

```
public JLabel getjLabel2() {  
    return jLabel2;  
}
```

```
public void setjLabel2(JLabel jLabel2) {  
    this.jLabel2 = jLabel2;  
}
```

```
public JLabel getjLabel3() {  
    return jLabel3;  
}
```

```
public void setjLabel3(JLabel jLabel3) {  
    this.jLabel3 = jLabel3;  
}
```

```
public JLabel getjLabel4() {  
    return jLabel4;  
}
```

```
public void setjLabel4(JLabel jLabel4) {  
    this.jLabel4 = jLabel4;  
}
```

```
public JScrollPane getjScrollPane1() {  
    return jScrollPane1;  
}
```

```
public void setjScrollPane1(JScrollPane jScrollPane1) {  
    this.jScrollPane1 = jScrollPane1;  
}
```

```
public JTextField getNamaField() {  
    return namaField;  
}
```

```
public void setNamaField(JTextField namaField) {  
    this.namaField = namaField;  
}
```

```
public JTextField getNpmField() {  
    return npmField;  
}
```

```
public void setNpmField(JTextField npmField) {  
    this.npmField = npmField;  
}
```

```
public JButton getRefreshButton() {  
    return refreshButton;  
}
```

```
public void setRefreshButton(JButton refreshButton) {  
    this.refreshButton = refreshButton;  
}
```

```
public JTextField getSemesterField() {  
    return semesterField;  
}
```

```
public void setSemesterField(JTextField semesterField) {  
    this.semesterField = semesterField;  
}
```

```
public JButton getSimpanButton() {  
    return simpanButton;  
}
```

```
public void setSimpanButton(JButton simpanButton) {  
    this.simpanButton = simpanButton;  
}
```

```
}
```