

ACTIVITY PERTEMUAN 4

NAMA : Muhammad Rifqi Dwi Putra
NPM : 51422145
KELAS : 4IA08
MATERI : Konsep dasar object relational mapping (ORM) dan framework hibernate

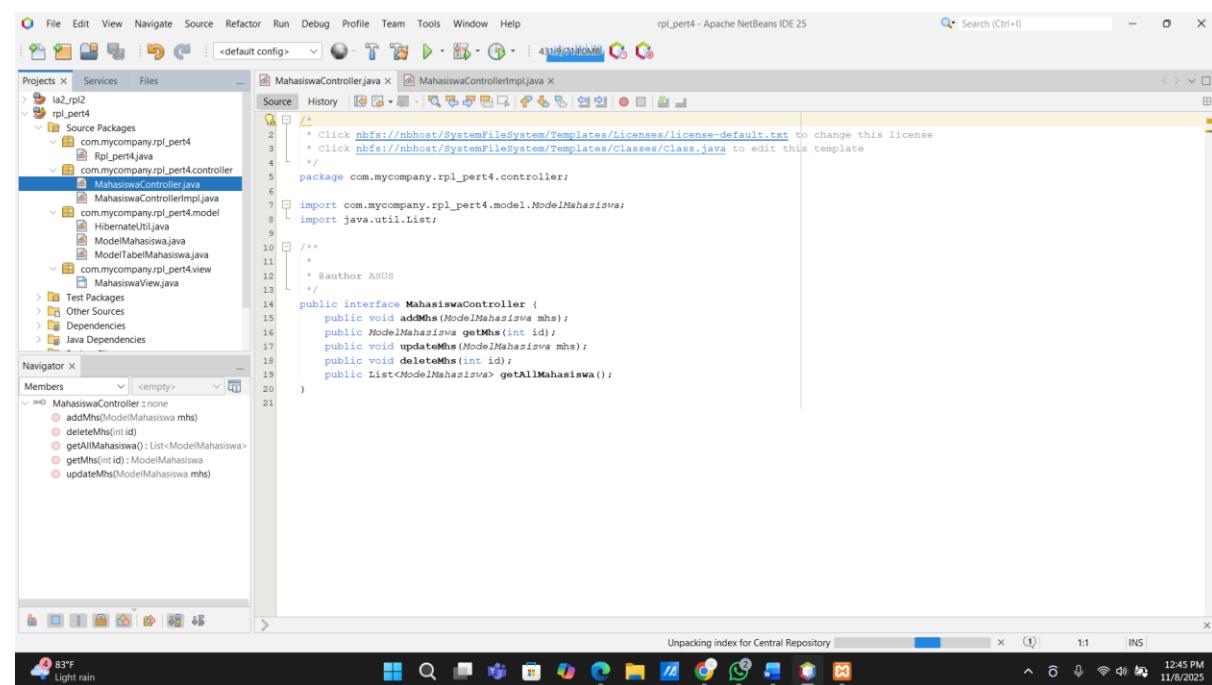
MATA PRAKTIKUM : Rekayasa Perangkat Lunak 2

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

Screenshot kode dan output program

Source Code :

File MahasiswaController.java



The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects View:** Shows the project structure with packages like `la2_rpl2`, `rpl_pert4`, and source packages `com.mycompany.rpl_pert4` and `com.mycompany.rpl_pert4.controller`. The file `MahasiswaController.java` is selected.
- Code Editor:** Displays the Java code for `MahasiswaController`. The code includes imports for `ModelMahasiswa` and `List`, and defines a public interface with methods for adding, updating, deleting, and getting lists of students.

```
1 package com.mycompany.rpl_pert4.controller;
2
3 import com.mycompany.rpl_pert4.model.ModelMahasiswa;
4 import java.util.List;
5
6 /**
7  * @author ASUS
8 */
9 public interface MahasiswaController {
10     public void addMhs(ModelMahasiswa mhs);
11     public ModelMahasiswa getMhs(int id);
12     public void updateMhs(ModelMahasiswa mhs);
13     public void deleteMhs(int id);
14     public List<ModelMahasiswa> getAllMahasiswa();
15 }
```

- Output View:** Shows the message "Unpacking index for Central Repository".
- System Tray:** Shows the date and time as 11/8/2025 at 12:45 PM.

File MahasiswaControllerImpl.java

```
* Click nbfs://nbbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
* Click nbfs://nbbhost/SystemFileSystem/Templates/classes/Class.java to edit this template
*/
package com.mycompany.rpl_perl4.controller;

import com.mycompany.rpl_perl4.model.HibernateUtil;
import com.mycompany.rpl_perl4.model.ModelMahasiswa;
import java.util.List;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.Query;

/**
 * Author AUS
 */
public class MahasiswaControllerImpl implements MahasiswaController{

    @Override
    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();
        }
    }

    @Override
    public List<ModelMahasiswa> getAllMahasiswa() {
        List<ModelMahasiswa> list = null;
        Session session = null;
        Transaction trx = null;
        Query query = null;
        ModelMahasiswa mhs = null;
        String hql = "from ModelMahasiswa";
        try {
            session = HibernateUtil.getSessionFactory().openSession();
            trx = session.beginTransaction();
            query = session.createQuery(hql);
            list = query.list();
            session.getTransaction().commit();
        } catch (Exception e) {
            if (trx != null) {
                trx.rollback();
            }
            e.printStackTrace();
        }
        return list;
    }

    @Override
    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();
        }
    }

    @Override
    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();
        }
    }
}
```

```
@Override
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();
    }
}

@Override
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();
    }
}

@Override
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();
    }
}
```

File MahasiswaControllerImpl.java

```

    75     @Override
    76     public List<ModelMahasiswa> getAllMahasiswa() {
    77         Transaction trx = null;
    78         List<ModelMahasiswa> listMhs = null;
    79
    80         try (Session session = HibernateUtil.getSessionFactory().openSession()) {
    81             trx = session.beginTransaction();
    82             // Using HQL (Hibernate Query Language) to fetch all records
    83             Query<ModelMahasiswa> query = session.createQuery("from ModelMahasiswa", ModelMahasiswa.class);
    84             listMhs = query.list(); // Fetch all results
    85
    86             trx.commit(); // Commit transaction
    87         } catch (Exception e) {
    88             if (trx != null) {
    89                 trx.rollback(); // Rollback transaction in case of error
    90             }
    91             e.printStackTrace();
    92         }
    93
    94         // Return the fetched list
    95         return listMhs;
    96     }
    97
    98     @Override
    99     public ModelMahasiswa getMhs(int id) {
100         throw new UnsupportedOperationException("Not supported yet.");
101     }
102 }

```

File HibernateUtil.java

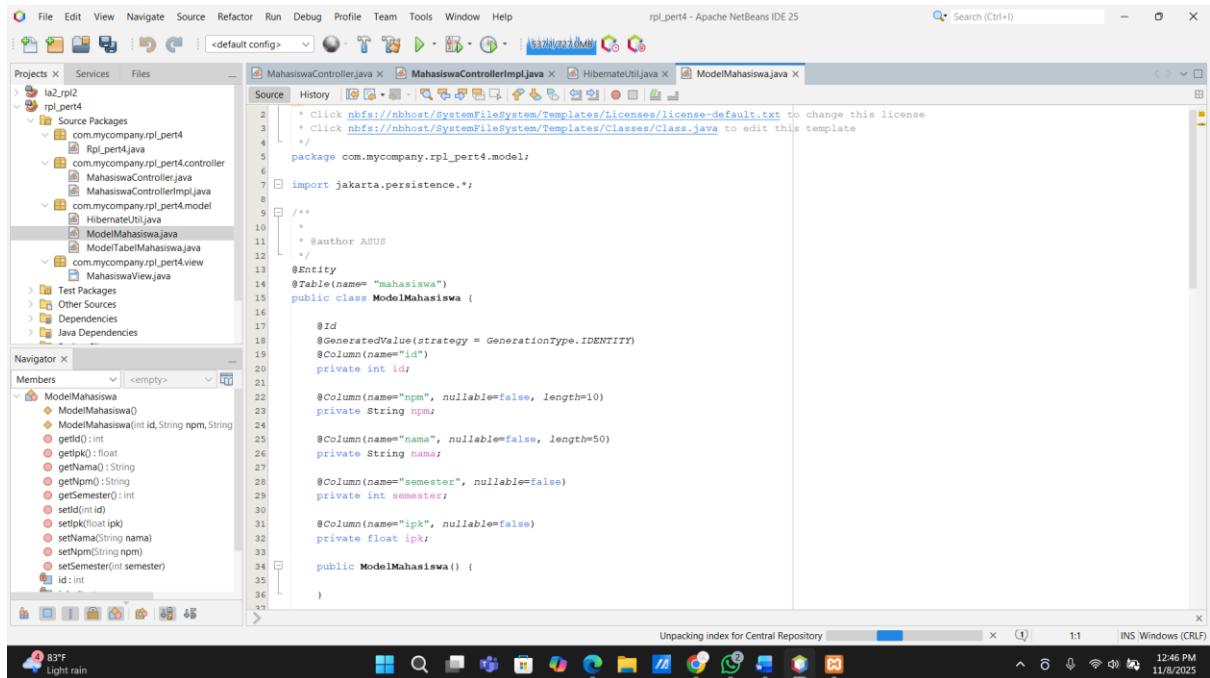
File HibernateUtil.java

```

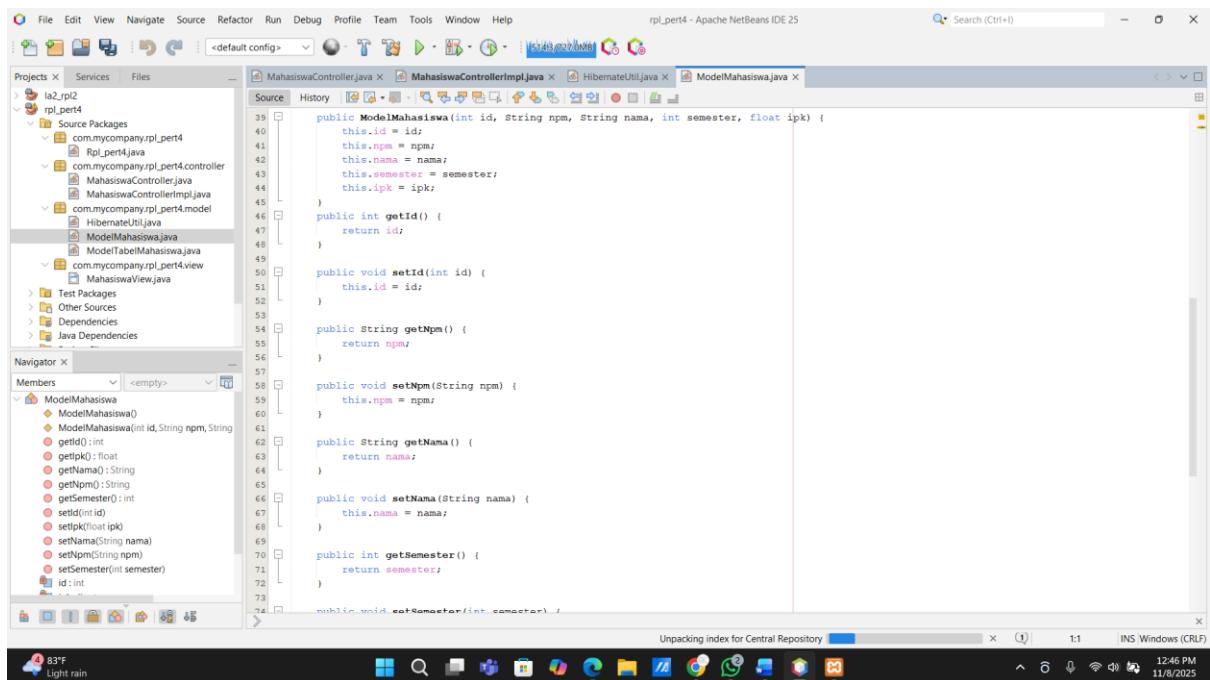
    4 package com.mycompany.rpl_pert4.model;
    5
    6 import org.hibernate.Session;
    7 import org.hibernate.SessionFactory;
    8 import org.hibernate.cfg.Configuration;
    9
   10 /**
   11  * 
   12  * @author ASUS
   13  */
   14 public class HibernateUtil {
   15     private static SessionFactory sessionFactory;
   16
   17     static {
   18         try {
   19             sessionFactory = new Configuration().configure().buildSessionFactory();
   20         } catch (Throwable ex) {
   21             System.err.println("Initial SessionFactory creation failed." + ex);
   22             throw new ExceptionInInitializerError(ex);
   23         }
   24     }
   25
   26     public static SessionFactory getSessionFactory() {
   27         return sessionFactory;
   28     }
   29
   30     public static void testConnection() {
   31         try (Session session = sessionFactory.openSession()) {
   32             System.out.println("Connection to the database was successful");
   33         } catch (Exception e) {
   34             System.err.println("Failed to connect to database.");
   35             e.printStackTrace();
   36         }
   37     }
   38 }

```

File ModelMahasiswa.java



```
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.mycompany.rpl_pert4.model;
6
7  import jakarta.persistence.*;
8
9  /**
10   * @author ASUS
11   */
12  @Entity
13  @Table(name= "mahasiswa")
14  public class ModelMahasiswa {
15
16      @Id
17      @GeneratedValue(strategy = GenerationType.IDENTITY)
18      @Column(name="id")
19      private int id;
20
21      @Column(name="npm", nullable=false, length=10)
22      private String npm;
23
24      @Column(name="nama", nullable=false, length=50)
25      private String nama;
26
27      @Column(name="semester", nullable=false)
28      private int semester;
29
30      @Column(name="ipk", nullable=false)
31      private float ipk;
32
33      public ModelMahasiswa() {
34
35  }
36
37  }
```



```
39  public ModelMahasiswa(int id, String npm, String nama, int semester, float ipk) {
40      this.id = id;
41      this.npm = npm;
42      this.nama = nama;
43      this.semester = semester;
44      this.ipk = ipk;
45  }
46  public int getId() {
47      return id;
48  }
49  public void setId(int id) {
50      this.id = id;
51  }
52  public String getNpm() {
53      return npm;
54  }
55  public void setNpm(String npm) {
56      this.npm = npm;
57  }
58  public String getNama() {
59      return nama;
60  }
61  public void setNama(String nama) {
62      this.nama = nama;
63  }
64  public int getSemester() {
65      return semester;
66  }
67  public void setSemester(int semester) {
68  }
69  }
```

```


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help



Source Packages com.mycompany.rpl_pert4 Rpl_pert4 MahasiswaController.java MahasiswaControllerImpl.java HibernateUtil.java ModelMahasiswa.java ModelTabelMahasiswa.java ModelTabelMahasiswa.java MahasiswaView.java Test Packages Other Sources Dependencies Java Dependencies



Source History <empty>



```

54 public String getNama() {
55 return nama;
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 setNama(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 public ModelMahasiswa(int id, String npm, String nama, float ipk, int semester) {
87 this.id = id;
88 this.npm = npm;
89 this.nama = nama;
90 this.ipk = ipk;
91 this.semester = semester;
92 }
93
94 }
```



Unpacking index for Central Repository 1:1 INS Windows (CRLF)



83°F Light rain


```

File ModelTabelMahasiswa.java

```


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help



Source Packages com.mycompany.rpl_pert4 Rpl_pert4 MahasiswaController.java MahasiswaControllerImpl.java HibernateUtil.java ModelMahasiswa.java ModelTabelMahasiswa.java ModelTabelMahasiswa.java MahasiswaView.java Test Packages Other Sources Dependencies Java Dependencies



Source History <empty>



```

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.mycompany.rpl_pert4.model;
6
7 import java.util.List;
8 import javax.swing.table.AbstractTableModel;
9
10 /**
11 * @author ASUS
12 */
13
14 public class ModelTabelMahasiswa extends AbstractTableModel {
15
16 private List<ModelMahasiswa> mahasiswaList;
17 private String[] columnNames = {"ID", "NPM", "Nama", "Semester", "IPR"};
18
19 public ModelTabelMahasiswa(List<ModelMahasiswa> mahasiswaList) {
20 this.mahasiswaList = mahasiswaList;
21 }
22
23 @Override
24 public int getRowCount() {
25 return mahasiswaList.size(); // Jumlah baris sesuai dengan jumlah data mahasiswa
26 }
27
28 @Override
29 public int getColumnCount() {
30 return columnNames.length; // Jumlah kolom sesuai dengan jumlah elemen dalam columnNames
31 }
32
33 @Override
34 public Object getValueAt(int rowIndex, int columnIndex) {
35 ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
36 switch (columnIndex) {
37 case 0:
38 return mahasiswa.getId();
39 case 1:
40 return mahasiswa.getNpm();
41 case 2:
42 return mahasiswa.getNama();
43 case 3:
44 return mahasiswa.getSemester();
45 case 4:
46 return mahasiswa.getIpk();
47 }
48 }
49 }
```



Missing mandatory Classpath entries. Resolve Project Problems.



Unpacking index for Central Repository 1:1 INS Windows (CRLF)



83°F Light rain

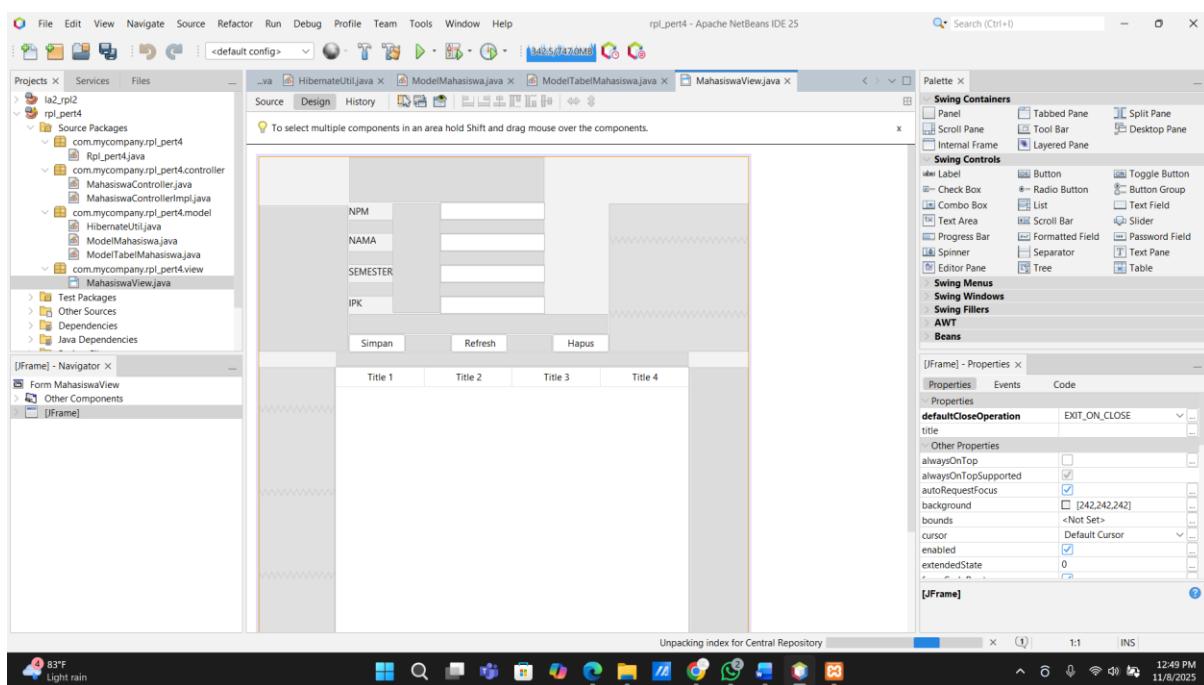

```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Title Bar:** rpl_pert4 - Apache NetBeans IDE 25
- Toolbar:** File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help.
- Search Bar:** Search (Ctrl+I)
- Project Explorer (Projects X):** Shows the project structure with packages like la2_rpl2, rpl_pert4, and source code files such as Rpl_pert4.java, MahasiswaController.java, MahasiswaControllerImpl.java, HibernateUtil.java, ModelMahasiswa.java, and ModelTabelMahasiswa.java.
- Code Editor:** Displays the Java code for `MahasiswaControllerImpl.java`. The code implements the `AbstractTableModel` and defines methods for getting column names, checking editability, and setting data lists.
- Navigator:** Shows the members of the `ModelTabelMahasiswa` class, including methods like `getColumnName`, `getColumnCount`, and `isCellEditable`.
- Bottom Status Bar:** Unpacking index for Central Repository

```
32     @Override
33     public Object getValueAt(int rowIndex, int columnIndex) {
34         ModelMahasiswa mahasiswa = mahasiswaList.get(rowIndex);
35         switch (columnIndex) {
36             case 0:
37                 return mahasiswa.getId();
38             case 1:
39                 return mahasiswa.getNm();
40             case 2:
41                 return mahasiswa.getNm();
42             case 3:
43                 return mahasiswa.getNm();
44             case 4:
45                 return mahasiswa.getNm();
46             default:
47                 return null;
48         }
49     }
50
51     @Override
52     public String getColumnName(int column) {
53         return columnNames[column]; // Mengatur nama kolom
54     }
55
56     @Override
57     public boolean isCellEditable(int rowIndex, int columnIndex) {
58         return false; // Semua sel tidak dapat diedit
59     }
60
61     // Method untuk menambahkan atau memodifikasi data, jika dibutuhkan
62     public void setMahasiswaList(List<ModelMahasiswa> mahasiswaList) {
63         this.mahasiswaList = mahasiswaList;
64         fireTableDataChanged(); // Memberitahu JTable bahwa data telah berubah
65     }
66 }
```

File MahasiswaView.java



Database

Databases

Database	Collation	Action
ap2b	utf8mb4_general_ci	Check privileges
bank	utf8mb4_general_ci	Check privileges
bank_51422145	utf8mb4_general_ci	Check privileges
information_schema	utf8_general_ci	Check privileges
ktib	utf8mb4_general_ci	Check privileges
la_pert3_db	utf8mb4_general_ci	Check privileges
mhs01	utf8mb4_general_ci	Check privileges
mysql	utf8mb4_general_ci	Check privileges
pasien_51422145	utf8mb4_general_ci	Check privileges
Console	utf8_general_ci	Check privileges
performance_schema	utf8_general_ci	Check privileges

Output :

Tambah/Simpan data

NPM: 51422145
NAMA: ammad Rofiq Dwi Putra
SEMESTER: 7
IPK: 4

Simpan Refresh Hapus

ID	NPM	Nama	Semester	IPK
1	51422145	Muhammad ...	7	4.0

```
stax.LocalXmlResourceResolver resolveEntity
http://hibernate.sourceforge.net/hibernate-configuration. Use namespace http://www.hibernate.org/dtd/hibernat
onFactoryInitiator initiateService

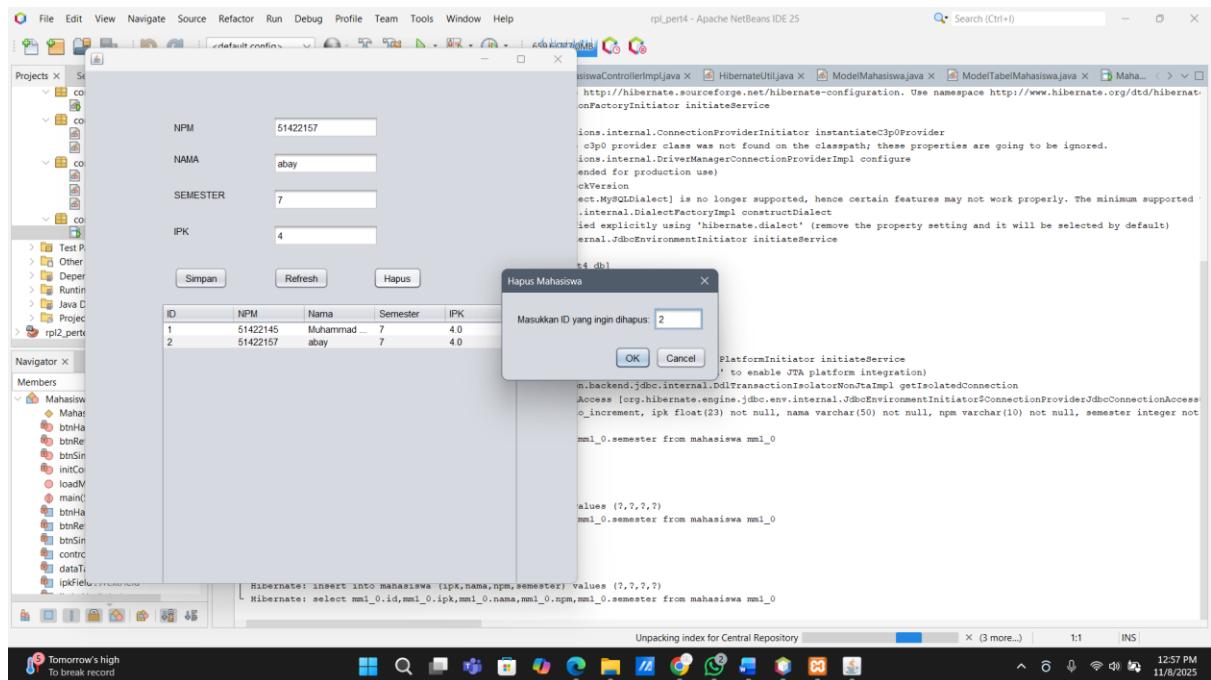
ions.internal.ConnectionProviderInitiator instantiateCjP0Provider
cjp0 provider class was not found on the classpath; these properties are going to be ignored.
ions.internal.DriverManagerConnectionProviderImpl configure
ended for production use)
sakai
ect.MySQLDialect] is no longer supported, hence certain features may not work properly. The minimum supported
.internal.DialectFactoryImpl constructDialect
ied explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)
ernal.JdbcEnvironmentInitiator initiateService

t4_db

jta.platform.internal.JtaPlatformInitiator initiateService
.transaction.jta.platform" to enable JTA platform integration)
n.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
Access [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess
o_increment, ipk float(2) not null, nama varchar(50) not null, npm varchar(10) not null, semester integer not
null]
nmi_0.semester from mahasiswa nmi_0

Hibernate: INSERT INTO mahasiswa (ipk,nama,npm,semester) values (7,?,?,?)
Hibernate: select nmi_0.id,nmi_0.ipk,nmi_0.nama,nmi_0.npm,nmi_0.semester from mahasiswa nmi_0
```

Hapus data



Setelah di refresh

