Tugas
Praktikum Pemrograman II

UNPAS
Teknik Informatika
Universitas Pasundan

Laporan Pertemuan 7

Identitas

-	NPM	: 223040141
-	Nama	: Muhammad Kholish Kamil
-	Kelas	: A
-	URL Repository Github	: https://github.com/kamil026/pp2_223040141_A

Penjelasan Kode

Jika berkas **tidak memiliki output**, gunakan tabel ini.

Membership

```
1. MySqlConnection.java
   package Pertemuan07.Membership.src.db;

   import java.sql.Connection;
   import java.sql.DriverManager;

   public class MySqlConnection {
       private final static String DB_URL =
   "jdbc:mysql://localhost:3306/pp2_membership";
       private final static String DB_USER = "root";
       private final static String DB_PASS = "";

       private static MySqlConnection instance;

       public static MySqlConnection getInstance() {
           if (instance == null) {
               instance = new MySqlConnection();
           }
           return instance;
       }

       public Connection getConnection() {
           Connection connection = null;
           try {
               Class.forName("com.mysql.cj.jdbc.Driver");
               connection = DriverManager.getConnection(
                   DB_URL, DB_USER, DB_PASS
               );
           } catch (Exception e) {
               e.printStackTrace();
           }
           return connection;
       }
   }



2. JenisMemberDao.java

   package Pertemuan07.Membership.src.dao;
```

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import Pertemuan07.Membership.src.db.MySqlConnection;
import Pertemuan07.Membership.src.model.JenisMember;

/**
 *
 * @author ASUS
 */
public class JenisMemberDao {
    public int insert(JenisMember jenisMember) {
        int result = -1;
        try (Connection connection =
MySqlConnection.getInstance().getConnection()) {
            PreparedStatement statement =
connection.prepareStatement("insert into jenis_member (id, nama)
values (?, ?)");
            statement.setString(1, jenisMember.getId());
            statement.setString(2, jenisMember.getNama());

            result = statement.executeUpdate();

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    public int update(JenisMember jenisMember) {
        int result = -1;
        try (Connection connection =
MySqlConnection.getInstance().getConnection()) {
            PreparedStatement statement =
connection.prepareStatement("update jenis_member set nama = ? where
id = ?");
            statement.setString(1, jenisMember.getNama());
            statement.setString(2, jenisMember.getId());

            result = statement.executeUpdate();

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    public int delete(JenisMember jenisMember) {
        int result = -1;
        try (Connection connection =
MySqlConnection.getInstance().getConnection()) {
            PreparedStatement statement =
connection.prepareStatement("delete from jenis_member where id =
?");
            statement.setString(1, jenisMember.getId());
            result = statement.executeUpdate();

        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    public List<JenisMember> findAll() {
    List<JenisMember> list = new ArrayList<>();
    try (Connection connection =
MySqlConnection.getInstance().getConnection();
         Statement statement = connection.createStatement()) {
        try (ResultSet resultSet = statement.executeQuery("select *
from jenis_member")) {
            // Retrieving the data
            while (resultSet.next()) {
                JenisMember jenisMember = new JenisMember();
                jenisMember.setId(resultSet.getString("id"));
                jenisMember.setNama(resultSet.getString("nama"));
                list.add(jenisMember);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return list;
    }
}
```

3. MemberDao.java

```
package Pertemuan07.Membership.src.dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;

import Pertemuan07.Membership.src.db.MySqlConnection;
import Pertemuan07.Membership.src.model.JenisMember;
import Pertemuan07.Membership.src.model.Member;

public class MemberDao {
    public int insert(Member member) {
        int result = -1;
        try (Connection connection =
MySqlConnection.getInstance().getConnection()) {
            String sql = "INSERT INTO member (id, nama,
jenis_member_id) VALUES (?, ?, ?)";
            PreparedStatement statement =
connection.prepareStatement(sql);

            statement.setString(1, member.getId());
            statement.setString(2, member.getNama());
            statement.setString(3,
member.getJenisMember().getId());

            result = statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    // Update an existing member
    public int update(Member member) {
        int result = -1;
        try (Connection connection =
MySqlConnection.getInstance().getConnection()) {
            String sql = "UPDATE member SET nama = ?,
jenis_member_id = ? WHERE id = ?";
            PreparedStatement statement =
connection.prepareStatement(sql);

            statement.setString(1, member.getNama());
            statement.setString(2,
member.getJenisMember().getId());
            statement.setString(3, member.getId());

            result = statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    // Delete a member
    public int delete(Member member) {
        int result = -1;
        try (Connection connection =
MySqlConnection.getInstance().getConnection()) {
            String sql = "DELETE FROM member WHERE id = ?";
            PreparedStatement statement =
connection.prepareStatement(sql);

            statement.setString(1, member.getId());

            result = statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return result;
    }

    // Find all members with their corresponding JenisMember
    public List<Member> findAll() {
        List<Member> list = new ArrayList<>();

        try (Connection connection =
MySqlConnection.getInstance().getConnection()) {
            String sql = "SELECT member.id, member.nama,
jenis_member.id jenis_member_id, jenis_member.nama
jenis_member_nama " +
                        "FROM member JOIN jenis_member ON
```

```
jenis_member.id = member.jenis_member_id";
            Statement statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery(sql);

            while (resultSet.next()) {
                Member member = new Member();
                member.setId(resultSet.getString("id"));
                member.setNama(resultSet.getString("nama"));

                JenisMember jenisMember = new JenisMember();

jenisMember.setId(resultSet.getString("jenis_member_id"));

jenisMember.setNama(resultSet.getString("jenis_member_nama"));

                member.setJenisMember(jenisMember);
                list.add(member);
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
        return list;
    }
}
```

4. JenisMember.java

```
package Pertemuan07.Membership.src.model;

public class JenisMember {
    private String id;
    private String nama;

    public void setId(String id) {
        this.id = id;
    }

    public String getId() {
        return id;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }

    public void addJenisMember(JenisMember jenisMember) {

        throw new UnsupportedOperationException("Unimplemented
method 'addJenisMember'");
    }
}
```

5. Member.java

```
package Pertemuan07.Membership.src.model;

public class Member {
    private String id;
    private String nama;
    private JenisMember jenisMember;

    public void setId(String id) {
        this.id = id;
    }

    public String getId() {
        return id;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }

    public void setJenisMember(JenisMember jenisMember) {
        this.jenisMember = jenisMember;
    }

    public JenisMember getJenisMember() {
        return jenisMember;
    }
}
```

6. JenisMemberButtonSimpanActionListener.java

```
package Pertemuan07.Membership.src.view.jenismember;

import java.awt.event.*;
import java.util.UUID;
import Pertemuan07.Membership.src.dao.JenisMemberDao;
import Pertemuan07.Membership.src.model.JenisMember;

public class JenisMemberButtonSimpanActionListener implements
ActionListener {
    private JenisMemberFrame jenisMemberFrame;
    private JenisMemberDao jenisMemberDao;

    public JenisMemberButtonSimpanActionListener(JenisMemberFrame
jenisMemberFrame, JenisMemberDao jenisMemberDao) {
        this.jenisMemberFrame = jenisMemberFrame;
        this.jenisMemberDao = jenisMemberDao;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String nama = this.jenisMemberFrame.getNama();
        JenisMember jenisMember = new JenisMember();
        jenisMember.setId(UUID.randomUUID().toString());
        jenisMember.setNama(nama);

        this.jenisMemberFrame.addJenisMember(jenisMember);
        this.jenisMemberDao.insert(jenisMember);
    }
}
```

7. JenisMemberFrame.java

```
package Pertemuan07.Membership.src.view.jenismember;

import javax.swing.*;
import java.util.*;
import Pertemuan07.Membership.src.model.JenisMember;
import Pertemuan07.Membership.src.dao.JenisMemberDao;

public class JenisMemberFrame extends JFrame {
    private List<JenisMember> jenisMemberList;
    private JTextField textFieldNama;
    private JenisMemberTableModel tableModel;
    public JenisMemberFrame(JenisMemberDao jenisMemberDao) {
        this.jenisMemberList = jenisMemberDao.findAll();
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JLabel labelInput = new JLabel("Nama:");
        labelInput.setBounds(15, 40, 350, 10);

        textFieldNama = new JTextField();
        textFieldNama.setBounds(15, 60, 350, 30);

        JButton button = new JButton("Simpan");
        button.setBounds(15, 100, 100, 40);

        JTable table = new JTable();
        JScrollPane scrollableTable = new JScrollPane(table);
        scrollableTable.setBounds(15, 150, 350, 200);

        tableModel = new JenisMemberTableModel(jenisMemberList);
        table.setModel(tableModel);

        JenisMemberButtonSimpanActionListener actionListener =
            new JenisMemberButtonSimpanActionListener(this,
jenisMemberDao);
        button.addActionListener(actionListener);

        this.add(button);
        this.add(textFieldNama);
        this.add(labelInput);
        this.add(scrollableTable);

        this.setSize(400, 500);
        this.setLayout(null);
    }

    public String getNama() {
        return textFieldNama.getText();
    }

    public void addJenisMember(JenisMember jenisMember) {
        tableModel.add(jenisMember);
        textFieldNama.setText("");
    }
}
```

8. JenisMemberTableModel.java

```
package Pertemuan07.Membership.src.view.jenismember;

import javax.swing.table.*;
import java.util.List;
import Pertemuan07.Membership.src.model.JenisMember;

class JenisMemberTableModel extends AbstractTableModel {
    private String[] columnNames = {"Nama"};
    private List<JenisMember> data;

    public JenisMemberTableModel(List<JenisMember> data) {
        this.data = data;
    }

    @Override
    public int getColumnCount() {
        return columnNames.length;
    }

    @Override
    public int getRowCount() {
        return data.size();
    }

    @Override
    public String getColumnName(int col) {
        return columnNames[col];
    }

    @Override
    public Object getValueAt(int row, int col) {
        JenisMember rowItem = data.get(row);
        String value = "";
        switch (col) {
            case 0:
                value = rowItem.getNama();
                break;
        }
        return value;
    }

    @Override
    public boolean isCellEditable(int row, int col) {
        return false;
    }

    public void add(JenisMember value) {
        data.add(value);
        fireTableRowsInserted(data.size() - 1, data.size() - 1);
    }
}
```

9. MainButtonActionListener.java

```
package Pertemuan07.Membership.src.view.main;


import java.awt.event.*;

public class MainButtonActionListener implements ActionListener{
    private MainFrame mainFrame;

    public MainButtonActionListener(MainFrame mainFrame){
        this.mainFrame = mainFrame;
```

```
       }

       public void actionPerformed(ActionEvent event){
           if(event.getSource() == mainFrame.getButtonJenisMember()){
               mainFrame.showJenisMemberFrame();
           } else {
               mainFrame.showMemberFrame();
           }
       }
   }

10.      MemberButtonSimpanActionListener.java
   package Pertemuan07.Membership.src.view.member;

   import java.awt.event.*;
   import java.util.UUID;
   import Pertemuan07.Membership.src.model.*;
   import Pertemuan07.Membership.src.dao.MemberDao;

   public class MemberButtonSimpanActionListener implements
   ActionListener {
       private MemberFrame memberFrame;
       private MemberDao memberDao;

       public MemberButtonSimpanActionListener(MemberFrame
   memberFrame, MemberDao memberDao) {
           this.memberFrame = memberFrame;
           this.memberDao = memberDao;
       }
       @Override
       public void actionPerformed(ActionEvent e) {
           // Get the name from the frame
           String nama = this.memberFrame.getNama();

           // Validate if name is not empty
           if (nama.isEmpty()) {
               this.memberFrame.showAlert("Nama tidak boleh kosong");
           } else {
               // Get selected JenisMember from the frame
               JenisMember jenisMember =
   this.memberFrame.getJenisMember();

               // Create new Member instance
               Member member = new Member();
               // Generate a unique ID using UUID
               member.setId(UUID.randomUUID().toString());
               member.setNama(nama);
               member.setJenisMember(jenisMember);

               // Update both the frame and database
               this.memberFrame.addMember(member);
               this.memberDao.insert(member);
           }
       }
   }

11.      MemberFrame.java
   package Pertemuan07.Membership.src.view.member;
```

```
import javax.swing.*;
import java.util.*;
import Pertemuan07.Membership.src.model.*;
import Pertemuan07.Membership.src.dao.MemberDao;
import Pertemuan07.Membership.src.dao.JenisMemberDao;

public class MemberFrame extends JFrame {
    private List<JenisMember> jenisMemberList;
    private List<Member> memberList;
    private JTextField textFieldNama;
    private MemberTableModel tableModel;
    @SuppressWarnings("rawtypes")
    private JComboBox comboJenis;
    private MemberDao memberDao;
    private JenisMemberDao jenisMemberDao;

    @SuppressWarnings("rawtypes")
    public MemberFrame(MemberDao memberDao, JenisMemberDao 
jenisMemberDao) {
        this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        this.memberDao = memberDao;
        this.jenisMemberDao = jenisMemberDao;

        this.memberList = this.memberDao.findAll();
        this.jenisMemberList = this.jenisMemberDao.findAll();

        // Create and setup name label and field
        JLabel labelInput = new JLabel("Nama:");
        labelInput.setBounds(15, 40, 350, 10);

        textFieldNama = new JTextField();
        textFieldNama.setBounds(15, 60, 350, 30);

        // Create and setup member type label and combo box
        JLabel labelJenis = new JLabel("Jenis Member:");
        labelJenis.setBounds(15, 100, 350, 10);

        comboJenis = new JComboBox();
        comboJenis.setBounds(15, 120, 350, 30);

        // Create and setup save button
        JButton button = new JButton("Simpan");
        button.setBounds(15, 160, 100, 40);

        // Create and setup table with scrollpane
        JTable table = new JTable();
        JScrollPane scrollableTable = new JScrollPane(table);
        scrollableTable.setBounds(15, 220, 350, 200);

        // Setup table model
        tableModel = new MemberTableModel(memberList);
        table.setModel(tableModel);

        // Setup action listener
        MemberButtonSimpanActionListener actionListener = 
            new MemberButtonSimpanActionListener(this, memberDao);
        button.addActionListener(actionListener);

        // Add all components to frame
        this.add(button);
        this.add(textFieldNama);
        this.add(labelInput);
        this.add(labelJenis);
        this.add(comboJenis);
        this.add(scrollableTable);

        // Set frame size and layout
        this.setSize(400, 500);
        this.setLayout(null);

        // Populate combo box
        populateComboJenis();
    }

    @SuppressWarnings("unchecked")
    public void populateComboJenis() {
        this.jenisMemberList = this.jenisMemberDao.findAll();
        comboJenis.removeAllItems();
        for (JenisMember jenisMember : this.jenisMemberList) {
            comboJenis.addItem(jenisMember.getNama());
        }
    }

    public String getNama() {
        return textFieldNama.getText();
    }

    public JenisMember getJenisMember() {
        return jenisMemberList.get(comboJenis.getSelectedIndex());
    }

    public void addMember(Member member) {
        tableModel.add(member);
        textFieldNama.setText("");
    }

    public void showAlert(String message) {
        JOptionPane.showMessageDialog(this, message);
    }
}
```

12. MemberTabelModel.java

```
package Pertemuan07.Membership.src.view.member;

import javax.swing.table.AbstractTableModel;
import java.util.List;

import Pertemuan07.Membership.src.model.Member;

class MemberTableModel extends AbstractTableModel {
    private final String[] columnNames = {"Nama", "Jenis Member"};
    private final List<Member> data;

    public MemberTableModel(List<Member> data) {
        this.data = data;
    }


    public int getColumnCount() {
        return columnNames.length;
    }


    public int getRowCount() {
        return data.size();
    }


    public String getColumnName(int col) {
        return columnNames[col];
    }


    public Object getValueAt(int row, int col) {
        Member rowItem = data.get(row);
        String value = "";
        switch (col) {
            case 0:
                value = rowItem.getNama();
                break;
            case 1:
                value = rowItem.getJenisMember().getNama();
                break;
        }
        return value;
    }

    public boolean isCellEditable(int row, int col) {
        return false;
    }

    public void add(Member value) {
        data.add(value);
        fireTableRowsInserted(data.size() - 1, data.size() - 1);
    }
}
```

13. MemberFrame.java

```
package Pertemuan07.Membership.src.view.main;

import java.awt.FlowLayout;
import javax.swing.*;
import
Pertemuan07.Membership.src.view.jenismember.JenisMemberFrame;
import Pertemuan07.Membership.src.view.member.MemberFrame;
import Pertemuan07.Membership.src.dao.*;

public class MainFrame extends JFrame {
    private JenisMemberFrame jenisMemberFrame;
    private MemberFrame memberFrame;
    private JButton buttonJenisMember;
    private JButton buttonMember;
    private JenisMemberDao jenisMemberDao;
    private MemberDao memberDao;

    public MainFrame(){
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 500);
        // Inisialisasi DAO
        this.jenisMemberDao = new JenisMemberDao();
        this.memberDao = new MemberDao();
            // Inisialisasi frame
            this.jenisMemberFrame = new
JenisMemberFrame(jenisMemberDao);
            this.memberFrame = new MemberFrame(memberDao,
jenisMemberDao);
        this.setLayout(new FlowLayout());
        MainButtonActionListener actionListener = new
MainButtonActionListener(this);

        this.buttonJenisMember = new JButton("Jenis Member");
        this.buttonMember = new JButton("Member");

        this.buttonJenisMember.addActionListener(actionListener);
        this.buttonMember.addActionListener(actionListener);

        this.add(buttonJenisMember);
        this.add(buttonMember);
    }

    public JButton getButtonJenisMember(){
        return buttonJenisMember;
    }

    public JButton getButtonMember(){
        return buttonMember;
    }

    public void showJenisMemberFrame(){
        if(jenisMemberFrame != null){
            jenisMemberFrame = new
JenisMemberFrame(jenisMemberDao);
        }

        jenisMemberFrame.setVisible(true);
    }

    public void showMemberFrame(){
        if(memberFrame == null){
            memberFrame = new MemberFrame(memberDao,
jenisMemberDao);
        }
        memberFrame.populateComboJenis();
        memberFrame.setVisible(true);
    }

    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                MainFrame mainFrame = new MainFrame();
                mainFrame.setVisible(true);
            }
        });
    }
}
```

Penjelasan

1. **MySqlConnection.java**

Mengatur koneksi database MySQL dengan metode singleton.

2. **JenisMemberDao.java**
   Operasi CRUD untuk entitas JenisMember (insert, update, delete, findAll).

3. **MemberDao.java**
   Operasi CRUD untuk entitas Member, termasuk data JenisMember.

4. **JenisMember.java**
   Model data untuk JenisMember (id, nama).

5. **Member.java**
   Model data untuk Member (id, nama, jenisMember).

6. **JenisMemberButtonSimpanActionListener.java**
   Event listener tombol simpan JenisMember.

7. **JenisMemberFrame.java**
   Antarmuka untuk mengelola JenisMember (input, tabel, simpan).

8. **JenisMemberTableModel.java**
   Table model untuk tabel JenisMember.

9. **MainButtonActionListener.java**:
   Event listener tombol utama membuka antarmuka JenisMember/Member.

10. **MemberButtonSimpanActionListener.java**
    Event listener tombol simpan Member.

11. **MemberFrame.java**
    Antarmuka untuk mengelola Member (input, tabel, simpan).

12. **MemberTableModel.java**
    Table model untuk tabel Member, termasuk jenisMember.

Penjelasan Kode

Jika **berkas memiliki output**, gunakan tabel ini.

MainFrame.java

```
package Pertemuan07.Membership.src.view.main;

import java.awt.FlowLayout;
import javax.swing.*;
import Pertemuan07.Membership.src.view.jenismember.JenisMemberFrame;
import Pertemuan07.Membership.src.view.member.MemberFrame;
import Pertemuan07.Membership.src.dao.*;

public class MainFrame extends JFrame {
    private JenisMemberFrame jenisMemberFrame;
    private MemberFrame memberFrame;
    private JButton buttonJenisMember;
    private JButton buttonMember;
    private JenisMemberDao jenisMemberDao;
    private MemberDao memberDao;

    public MainFrame(){
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setSize(400, 500);
        // Inisialisasi DAO
        this.jenisMemberDao = new JenisMemberDao();
        this.memberDao = new MemberDao();
            // Inisialisasi frame
            this.jenisMemberFrame = new JenisMemberFrame(jenisMemberDao);
            this.memberFrame = new MemberFrame(memberDao,
jenisMemberDao);

        this.setLayout(new FlowLayout());
        MainButtonActionListener actionListener = new
MainButtonActionListener(this);

        this.buttonJenisMember = new JButton("Jenis Member");
        this.buttonMember = new JButton("Member");

        this.buttonJenisMember.addActionListener(actionListener);
        this.buttonMember.addActionListener(actionListener);


        this.add(buttonJenisMember);
        this.add(buttonMember);
    }

    public JButton getButtonJenisMember(){
        return buttonJenisMember;
    }

    public JButton getButtonMember(){
        return buttonMember;
    }

    public void showJenisMemberFrame(){
        if(jenisMemberFrame != null){
            jenisMemberFrame = new JenisMemberFrame(jenisMemberDao);
        }

        jenisMemberFrame.setVisible(true);
    }

    public void showMemberFrame(){
        if(memberFrame == null){
            memberFrame = new MemberFrame(memberDao, jenisMemberDao);
        }
        memberFrame.populateComboJenis();
        memberFrame.setVisible(true);
    }

    public static void main(String[] args) {
        javax.swing.SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                MainFrame mainFrame = new MainFrame();
                mainFrame.setVisible(true);
            }
        });
    }
}
```

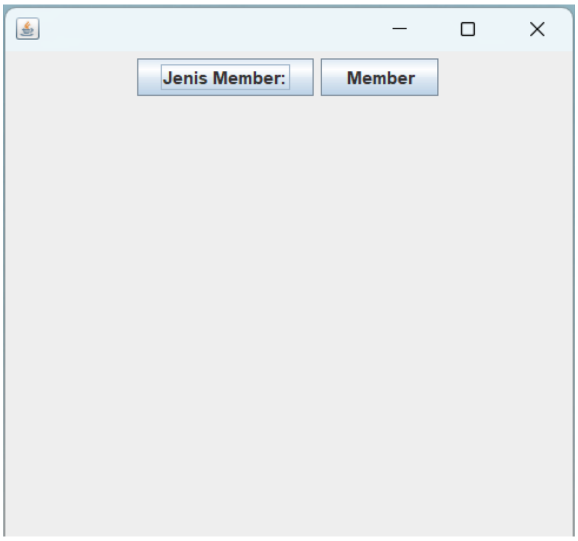
Penjelasan

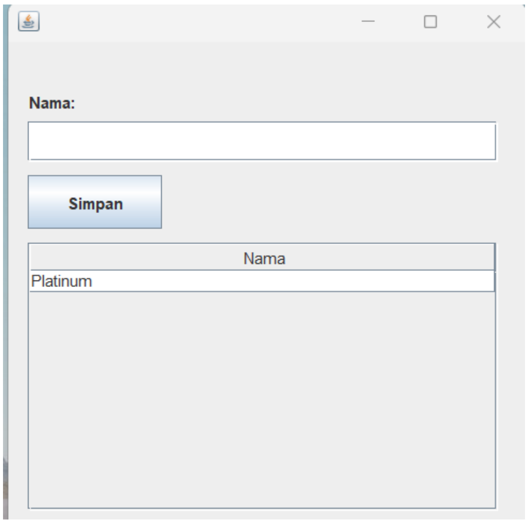
MainFrame adalah antarmuka utama aplikasi yang menyediakan tombol untuk membuka dua fitur utama: Jenis Member dan Member.

Fungsinya:

1. Inisialisasi: Membuat objek DAO (JenisMemberDao, MemberDao) dan frame (JenisMemberFrame, MemberFrame).
2. Tombol Navigasi: Tombol "Jenis Member" membuka pengelolaan jenis member, tombol "Member" membuka pengelolaan data member.
3. Navigasi: Metode showJenisMemberFrame() dan showMemberFrame() mengatur tampilan frame masing-masing.
4. Main Method: Memulai aplikasi dengan menampilkan MainFrame menggunakan SwingUtilities.

Output





Nama
Platinum