



Laporan Tugas Pertemuan ke-7

Identitas

- NPM : 223040114
- Nama : Fawwas Nawwaf Sabil
- Kelas : C
- URL Repository Github :

https://github.com/FawwasSabil25/pp2_223040114_C/tree/main/Pertemuan7/Membership

ip

Penjelasan Kode

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

MemberDao.java

```
public class MemberDao {  
    public int insert(Member member){  
        int result = -1;  
        try(Connection connection =  
        MySqlConnection.getInstance().getConnection();){  
            PreparedStatement statement =  
            connection.prepareStatement("Insert into member (id, nama,  
            jenis_member_id) values(?,?,?)");  
            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;  
    }  
}...dst
```

Penjelasan

File MemberDao.java mengatasi data Member, termasuk operasi CRUD ke database.

JenisMemberDao.java

```
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;  
}  
  
...dst
```

Penjelasan

File JenisMemberDao.java menangani data JenisMember, termasuk operasi CRUD di dalam database.

Member.java

```
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 setName(String nama){  
        this.nama = nama;  
    }  
  
    public String getName(){  
        return nama;  
    }  
  
    public void setJenisMember(JenisMember  
jenisMember){  
        this.jenisMember = jenisMember;  
    }  
  
    public JenisMember getJenisMember(){  
        return jenisMember;  
    }  
}
```



```
}
```

Penjelasan

File Member merupakan model utk menyimpan data Member tanpa tampilan

JenisMember.java

```
public class JenisMember {  
  
    private String id;  
    private String nama;  
  
    public void setId(String id){  
        this.id = id;  
    }  
  
    public String getId(){  
        return id;  
    }  
  
    public void setName(String nama){  
        this.nama = nama;  
    }  
  
    public String getName(){  
        return nama;  
    }  
  
}
```

Penjelasan

File JenisMember berfungsi untuk menyimpan data jenis member tanpa tampilan



Jika berkas memiliki output, gunakan tabel ini.

MainFrame.java

```
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);

        this.jenisMemberDao = new JenisMemberDao();
        this.memberDao = new MemberDao();

        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.populateComboJenis();
        }
    }
}
```



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

Penjelasan

.File MainFrame berfungsi sebagai UI yang menampilkan semua tombol yang berfungsi untuk mengakses JenisMemberFrame dan MemberFrame

MemberFrame.java



```
public MemberFrame(MemberDao memberDao, JenisMemberDao jenisMemberDao) {
    setTitle("Membership Management"); // Judul utama jendela
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

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

    // Mengambil data Member dan JenisMember dari database
    this.memberList = this.memberDao.findAll();
    this.jenisMemberList = this.jenisMemberDao.findAll();

    // Label untuk input nama
    JLabel labelInput = new JLabel("Nama:");
    labelInput.setBounds(15, 40, 350, 10);

    // Text field untuk input nama
    JTextField textFieldNama = new JTextField();
    textFieldNama.setBounds(15, 60, 350, 30);

    // Label untuk combo box Jenis Member
    JLabel labelJenis = new JLabel("Jenis Member:");
    labelJenis.setBounds(15, 100, 350, 10);

    // JComboBox untuk memilih Jenis Member
    JComboBox<> comboJenis = new JComboBox<>();
    comboJenis.setBounds(15, 120, 150, 30);

    // Tombol Simpan
    JButton buttonSimpan = new JButton("Simpan");
    buttonSimpan.setBounds(15, 160, 100, 40);

    // Tombol Update
    JButton buttonUpdate = new JButton("Update");
    buttonUpdate.setBounds(130, 160, 100, 40);

    // Tombol Delete
    JButton buttonDelete = new JButton("Delete");
    buttonDelete.setBounds(245, 160, 100, 40);

    // Tabel untuk menampilkan daftar Member
    table = new JTable();
    JScrollPane scrollableTable = new JScrollPane(table);
    scrollableTable.setBounds(15, 220, 350, 200);

    // Model tabel untuk daftar Member
    tableModel = new MemberTableModel(memberList);
    table.setModel(tableModel);

    // Listener untuk tombol Simpan
    buttonSimpan.addActionListener(new
    MemberButtonSimpanActionListener(this, memberDao));

    // Listener untuk tombol Update
    buttonUpdate.addActionListener(new
    MemberButtonUpdateActionListener(this, memberDao));
```



```
// Listener untuk tombol Delete
buttonDelete.addActionListener(new
MemberButtonDeleteActionListener(this, memberDao));

// Menambahkan komponen ke JFrame
this.add(buttonSimpan);
this.add(buttonUpdate);
this.add(buttonDelete);
this.add(textFieldNama);
this.add(labelInput);
this.add(labelJenis);
this.add(comboJenis);
this.add(scrollableTable);

this.setSize(400, 500);
this.setLayout(null);
this.setVisible(true);

// Memuat data Jenis Member ke ComboBox
populateComboJenis();
}

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 updateMember(Member member) {
    int selectedRow = table.getSelectedRow();
    if (selectedRow >= 0) {
        memberList.set(selectedRow, member);
        tableModel.fireTableRowsUpdated(selectedRow, selectedRow);
    }
    textFieldNama.setText("");
}

public void deleteMember(Member member) {
    int selectedRow = table.getSelectedRow();
    if (selectedRow >= 0) {
        memberList.remove(selectedRow);
    }
}
```



```
        tableModel.fireTableRowsDeleted(selectedRow, selectedRow);
    }
    textFieldNama.setText("");
}

public Member getSelectedMember() {
    int selectedRow = table.getSelectedRow();
    if (selectedRow >= 0) {
        return memberList.get(selectedRow);
    }
    return null;
}

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

public void refreshMemberTable() {
    memberList = memberDao.findAll();
    tableModel.setData(memberList);
    tableModel.fireTableDataChanged();
}
}
```

Penjelasan

File MemberFrame berfungsi sebagai UI input untuk nama, jenis member dan daftar member

MemberTableModel.java

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

    public void setData(List<Member> memberList) {
        this.memberList = memberList;
        fireTableDataChanged(); // Memperbarui tampilan tabel
    }

    public MemberTableModel(List<Member> 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];
}

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

}
```

Penjelasan

MemberTableModel berfungsi sebagai controller dalam penampilan tabel

MemberButtonSimpanActionListener.java



```
package view.member;

import java.awt.event.ActionEvent;
import
java.awt.event.ActionListener;
import model.Member; import
dao.MemberDao;

public class MemberButtonSimpanActionListener implements ActionListener {
private MemberFrame memberFrame;

    public MemberButtonSimpanActionListener(MemberFrame memberFrame,
MemberDao memberDao) {
        this.memberFrame = memberFrame;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
// Menambah member baru ke tabel dan database
        String nama = memberFrame.getNama();
        if (!nama.isEmpty()) {
            memberFrame.addMember(new Member(nama, jenisMember));
        }
    }
}
```

Penjelasan

MemberButtonSimpanActionListener berfungsi untuk menetapkan fungsi untuk button

Output



Membership Management

Nama:

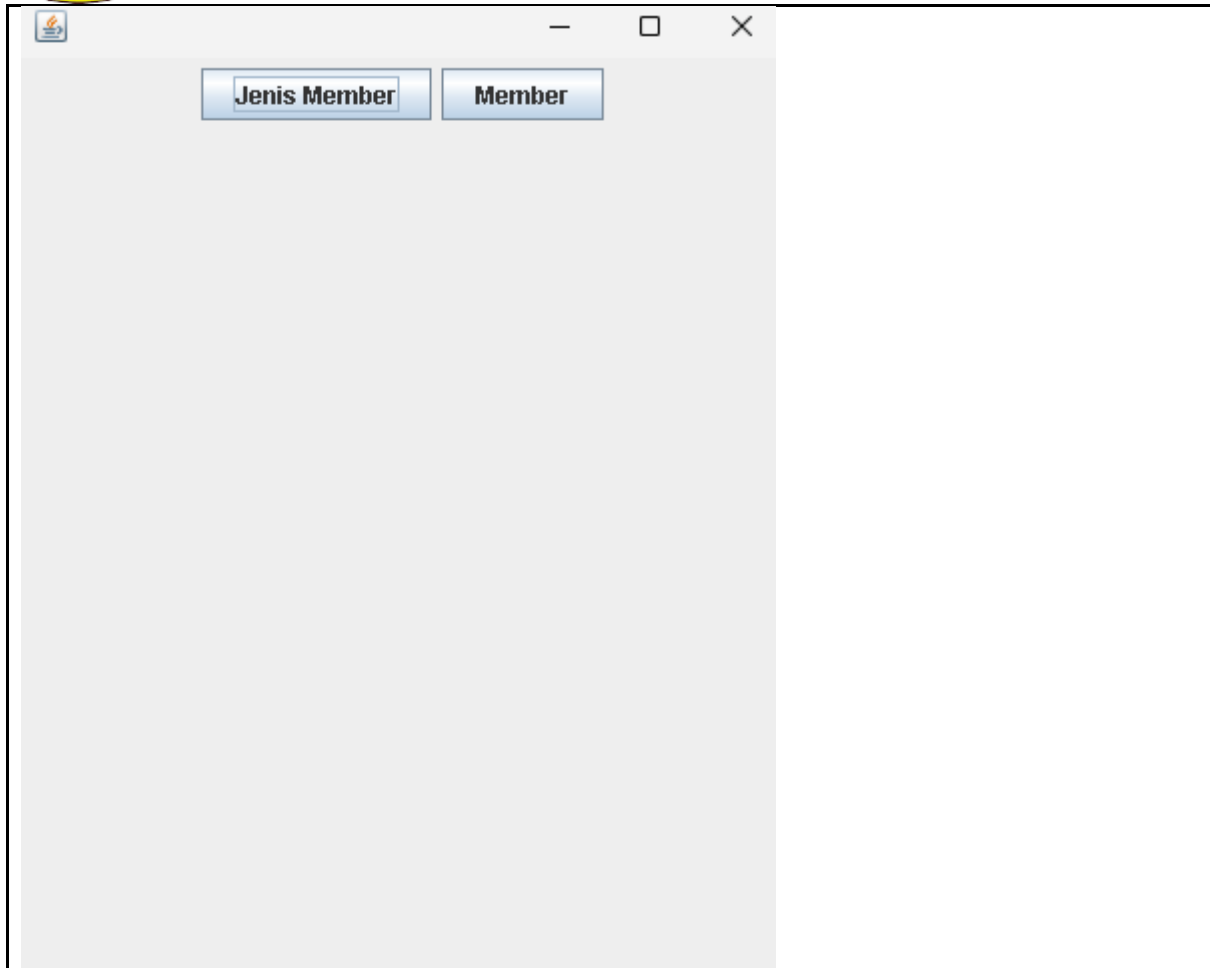
Jenis Member:

Simpan

Update

Delete

Nama	Jenis Member
------	--------------



Catatan:

- Silakan modifikasi dan tambahkan informasi sesuai dengan proyek yang Anda kerjakan baik pada tanda [kalimat].
- Sesuaikan nama file laporan dengan format **Tugas[Pertemuan ke-X]_PP1_NPM**
Contoh: Tugas[P10]_PP1_203040104
- Sesuaikan judul laporan dengan format **Laporan Tugas Pertemuan ke-X** Contoh:
Laporan Tugas Pertemuan ke-10