



Laporan Tugas Pertemuan ke-8

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.

Membuat file dengan nama mybatis-config.xml

```
1  <?xml version="1.0" encoding="UTF-8" ?>
2  <!DOCTYPE configuration
3      PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4      "http://mybatis.org/dtd/mybatis-3-config.dtd">
5  <configuration>
6      <environments default="development">
7          <environment id="development">
8              <transactionManager type="JDBC" />
9              <dataSource type="POOLED">
10                 <property name="driver" value="com.mysql.cj.jdbc.Driver" />
11                 <property name="url" value="jdbc:mysql://localhost:3306/pp2_membership" />
12                 <property name="username" value="root" />
13                 <property name="password" value="" />
14             </dataSource>
15         </environment>
16     </environments>
17     <mappers>
18         <package name="mapper" />
19     </mappers>
20 </configuration>
```

Penjelasan

Konfigurasi ini mendefinisikan bagaimana MyBatis akan berinteraksi dengan database, termasuk pengaturan koneksi dan pemetaan (mapping) antara objek Java dan tabel database. Mengatur koneksi ke database MySQL untuk aplikasi berbasis Java yang menggunakan MyBatis, serta menentukan lokasi mapper SQL yang akan mengelola operasi database.



Membuat folder/package model lalu buat dua kelas yaitu Member dan JenisMember

- JenisMember

```
1  package model;
2  public class JenisMember{
3      private String id;
4      private String nama;
5
6      public void setId(String id){
7          this.id = id;
8      }
9
10     public String getId(){
11         return id;
12     }
13
14     public void setName(String nama){
15         this.nama = nama;
16     }
17
18     public String getName(){
19         return nama;
20     }
21 }
```



- Member

```
1  package model;
2
3  public class Member{
4      private String id;
5      private String nama;
6      private String jenisMemberId;
7      private JenisMember jenisMember;
8
9      public void setId(String id) {
10         this.id = id;
11     }
12
13     public String getId() {
14         return id;
15     }
16
17     public void setName(String nama) {
18         this.nama = nama;
19     }
20
21     public String getName() {
22         return nama;
23     }
24
25     public void setJenisMemberId(String jenisMemberId) {
26         this.jenisMemberId = jenisMemberId;
27     }
```



```
28
29     public String getId() {
30         return id;
31     }
32
33     public JenisMember getId() {
34         return jenisMember;
35     }
36
37     public void setId(JenisMember jenisMember) {
38         this.jenisMember = jenisMember;
39     }
40
41
42 }
```

Penjelasan

Kedua kelas ini membentuk struktur dasar untuk sebuah sistem keanggotaan di mana setiap anggota memiliki jenis keanggotaan tertentu. Dengan pendekatan ORM, kita dapat memetakan objek Java ini ke dalam tabel database, memungkinkan interaksi yang lebih mudah antara aplikasi dan data.



Membuat package mapper, dua kelas dengan nama MemberMapper dan JenisMemberMapper

- JenisMemberMapper

```
1 package mapper;
2
3 import java.util.List;
4 import model.JenisMember;
5 import org.apache.ibatis.annotations.Delete;
6 import org.apache.ibatis.annotations.Insert;
7 import org.apache.ibatis.annotations.Result;
8 import org.apache.ibatis.annotations.Results;
9 import org.apache.ibatis.annotations.Select;
10 import org.apache.ibatis.annotations.Update;
11
12 public interface JenisMemberMapper {
13     @Insert("INSERT INTO jenis_member (id, nama) VALUES(#{id}, #{nama})")
14     public Integer insert(JenisMember jenisMember);
15
16     @Update("UPDATE jenis_member SET nama = #{nama} WHERE id = #{id}")
17     public Integer update(JenisMember jenisMember);
18
19     @Delete("DELETE FROM jenis_member WHERE id = #{id}")
20     public Integer delete(JenisMember jenisMember);
21
22     @Select("SELECT * FROM jenis_member")
23     @Results(value = {
24         @Result(property = "id", column = "id"),
25         @Result(property = "nama", column = "nama")
26     })
27     List<JenisMember> findAll();
28 }
```

- MemberMapper

```
1 package mapper;
2
3 import java.util.List;
4 import model.JenisMember;
5 import model.Member;
6 import org.apache.ibatis.annotations.Delete;
7 import org.apache.ibatis.annotations.Insert;
8 import org.apache.ibatis.annotations.One;
9 import org.apache.ibatis.annotations.Result;
10 import org.apache.ibatis.annotations.Results;
11 import org.apache.ibatis.annotations.Select;
12 import org.apache.ibatis.annotations.Update;
13
14 public interface MemberMapper {
15
16     @Insert("INSERT INTO member (id, nama, jenis_member_id) VALUES(#{id}, #{nama}, #{jenisMemberId})")
17     public Integer insert(Member member);
18
19     @Update("UPDATE member SET nama = #{nama}, jenis_member_id = #{jenisMemberId} WHERE id = #{id}")
20     public Integer update(Member member);
21
22     @Delete("DELETE FROM member WHERE id = #{id}")
23     public Integer delete(Member member);
24
25     @Select("SELECT * FROM member")
26     @Results(value = {
27         @Result(property = "id", column = "id"),
28         @Result(property = "nama", column = "nama"),
29         @Result(property = "jenisMember", column = "jenis_member_id",
30             javaType = JenisMember.class,
31             one = @One(select = "mapper.MemberMapper.getJenisMember"))
32     })
33     List<Member> findAll();
34
35     @Select("SELECT * FROM jenis_member WHERE id = #{id}")
36     @Results(value = {
37         @Result(property = "id", column = "id"),
38         @Result(property = "nama", column = "nama")
39     })
40     JenisMember getJenisMember(String id);
41 }
```

Penjelasan

Kode di atas mendefinisikan dua interface mapper MyBatis, yaitu JenisMemberMapper dan MemberMapper. Mapper ini berfungsi sebagai antarmuka antara aplikasi Java dan database, mengelola operasi CRUD (Create, Read, Update, Delete) untuk entitas JenisMember dan Member.



Membuat package dao dengan kelas MemberDao dan JenisMemberDao.

- JenisMemberDao

```
1 package dao;
2
3 import java.util.List;
4 import model.JenisMember;
5 import org.apache.ibatis.session.SqlSession;
6 import org.apache.ibatis.session.SqlSessionFactory;
7
8 public class JenisMemberDao {
9
10     private final SqlSessionFactory sqlSessionFactory;
11
12     public JenisMemberDao(SqlSessionFactory sqlSessionFactory) {
13         this.sqlSessionFactory = sqlSessionFactory;
14     }
15
16     public int insert(JenisMember jenisMember){
17         int result;
18         try(SqlSession session = sqlSessionFactory.openSession()){
19             result = session.insert("mapper.JenisMemberMapper.insert", jenisMember);
20             session.commit();
21         }
22
23         return result;
24     }
25 }
```

```
26     public int update(JenisMember jenisMember){
27         int result;
28         try(SqlSession session = sqlSessionFactory.openSession()){
29             result = session.update("mapper.JenisMemberMapper.update", jenisMember);
30             session.commit();
31         }
32         return result;
33     }
34
35     public int delete(String jenisMemberId){
36         int result;
37         try(SqlSession session = sqlSessionFactory.openSession()){
38             result = session.delete("mapper.JenisMemberMapper.delete", jenisMemberId);
39             session.commit();
40         }
41         return result;
42     }
43
44     public List<JenisMember> findAll(){
45         List<JenisMember> jenisMembers;
46         try(SqlSession session = sqlSessionFactory.openSession()){
47             jenisMembers = session.selectList("mapper.JenisMemberMapper.findAll");
48         }
49
50         return jenisMembers;
51     }
52 }
```

- MemberDao

```
1  package dao;
2
3  import java.util.List;
4  import model.Member;
5  import org.apache.ibatis.session.SqlSession;
6  import org.apache.ibatis.session.SqlSessionFactory;
7
8  public class MemberDao {
9      private final SqlSessionFactory sqlSessionFactory;
10
11     public MemberDao(SqlSessionFactory sqlSessionFactory) {
12         this.sqlSessionFactory = sqlSessionFactory;
13     }
14
15     public int insert(Member member){
16         int result;
17         try(SqlSession session = sqlSessionFactory.openSession()){
18             result = session.insert("mapper.MemberMapper.insert", member);
19             session.commit();
20         }
21         return result;
22     }
23
24
25     public int update(Member member){
26         int result;
27         try(SqlSession session = sqlSessionFactory.openSession()){
28             result = session.update("mapper.MemberMapper.update", member);
29             session.commit();
30         }
31     }
32 }
```




```
30     }
31     return result;
32 }
33
34 public int delete(String memberId){
35     int result;
36     try(SqlSession session = sqlSessionFactory.openSession()){
37         result = session.delete("mapper.MemberMapper.delete", memberId);
38         session.commit();
39     }
40     return result;
41 }
42
43 public List<Member> findAll(){
44     List<Member> result;
45     try(SqlSession session = sqlSessionFactory.openSession()){
46         result = session.selectList("mapper.MemberMapper.findAll");
47     }
48     return result;
49 }
50 }
```

Penjelasan

Kode di atas mendefinisikan dua kelas DAO (Data Access Object) untuk mengelola akses data menggunakan MyBatis: JenisMemberDao dan MemberDao. Kelas-kelas ini berfungsi sebagai lapisan antara aplikasi dan database, menangani operasi CRUD (Create, Read, Update, Delete) untuk entitas JenisMember dan Member.

Kelas JenisMemberDao dan MemberDao ini menyediakan abstraksi yang memisahkan logika akses data dari logika bisnis aplikasi. Dengan menggunakan MyBatis, mereka mempermudah eksekusi query SQL dan pengelolaan transaksi, memastikan interaksi yang aman dan efisien dengan database.

Membuat package view.jenismember lalu buat kelas JenisMemberTableModel, JenisMemberButtonSimpanActionListener, dan JenisMemberFrame.

- JenisMemberTableModel



```
1 package View.jenisMember;
2
3 import java.util.List;
4 import javax.swing.table.AbstractTableModel;
5 import model.JenisMember;
6
7 public class JenisMemberTableModel extends AbstractTableModel{
8     private String[] columnNames = {"Nama"};
9     private List<JenisMember> data;
10
11     public JenisMemberTableModel(List<JenisMember> data) {
12         this.data = data;
13     }
14
15     public int getColumnCount() {
16         return columnNames.length;
17     }
18
19     public int getRowCount() {
20         return data.size();
21     }
22
23     public String getColumnName(int column) {
24         return columnNames[column];
25     }
26 }
```

```
27     public Object getValueAt(int row, int col) {
28         JenisMember rowItem = data.get(row);
29         String value="";
30
31         switch(col){
32             case 0:
33                 value = rowItem.getNama();
34                 break;
35         }
36
37         return value;
38     }
39
40     public boolean isCellEditable(int row, int col) {
41         return false;
42     }
43
44     public void add(JenisMember value){
45         data.add(value);
46         fireTableRowsInserted(data.size() - 1, data.size() - 1);
47     }
48
49     public void setData(List<JenisMember> data) {
50         this.data = data;
51         fireTableDataChanged();
52     }
53 }
```

- JenisMemberButtonActionListener



```
1 package view.jenisMember;
2
3 import dao.JenisMemberDao;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.util.UUID;
7 import model.JenisMember;
8
9
10 public class JenisMemberButtonSimpanActionListener implements ActionListener {
11     private JenisMemberFrame jenisMemberFrame;
12     private JenisMemberDao jenisMemberDao;
13
14     public JenisMemberButtonSimpanActionListener(JenisMemberFrame jenisMemberFrame, JenisMemberDao jenisMemberDao) {
15         this.jenisMemberFrame = jenisMemberFrame;
16         this.jenisMemberDao = jenisMemberDao;
17     }
18
19     @Override
20     public void actionPerformed(ActionEvent e) {
21         String nama = this.jenisMemberFrame.getName();
22
23         JenisMember jenisMember = new JenisMember();
24         jenisMember.setId(UUID.randomUUID().toString());
25         jenisMember.setNama(nama);
26
27         this.jenisMemberFrame.addJenisMember(jenisMember);
28         this.jenisMemberDao.insert(jenisMember);
29     }
30 }
31
32 }
```

- JenisMemberFrame



```
1 package view.jenisMember;
2
3 import dao.JenisMemberDao;
4 import dao.MemberDao;
5
6 import java.awt.event.MouseAdapter;
7 import java.awt.event.MouseEvent;
8 import java.util.List;
9 import javax.swing.JButton;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JScrollPane;
13 import javax.swing.JTable;
14 import javax.swing.JTextField;
15 import model.JenisMember;
16 import model.Member;
17 import memberdetail.MemberDetailFrame;
18
19 public class JenisMemberFrame extends JFrame {
20
21     private List<JenisMember> jenisMemberList;
22     private JTextField textFieldNama;
23     private JenisMemberTableModel tableModel;
24     private JenisMemberDao jenisMemberDao;
25     private MemberDao memberDao;
26     private Member member;
27
28
29     public JenisMemberFrame(JenisMemberDao jenisMemberDao){
30         this.jenisMemberDao = jenisMemberDao;
31         this.jenisMemberList = jenisMemberDao.findAll();
32         this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
33
34         JLabel labelInput = new JLabel(text:"Nama: ");
35         labelInput.setBounds(x:15,y:40,width:350,height:10);
36
37         textFieldNama = new JTextField();
38         textFieldNama.setBounds(x:15,y:60,width:350,height:30);
39
40         JButton button = new JButton(text:"Simpan");
41         button.setBounds(x:15,y:100,width:100,height:40);
42
43         javax.swing.JTable table = new JTable();
44         JScrollPane scrollableTable = new JScrollPane(table);
```



```
45 scrollableTable.setBounds(x:15,y:150,width:350,height:200);
46
47 tableModel = new JenisMemberTableModel(jenisMemberList);
48 table.setModel(tableModel);
49
50 JenisMemberButtonSimpanActionListener actionListener = new JenisMemberButtonSimpanActionListener(this, jenisMemberDao);
51
52 button.addActionListener(actionListener);
53
54
55 table.addMouseListener(new MouseAdapter() {
56     public void mouseClicked(MouseEvent e) {
57         if (e.getClickCount() == 2) { // Double click untuk membuka detail
58             int row = table.getSelectedRow();
59             JenisMember selectedJenisMember = jenisMemberList.get(row);
60             MemberDetailFrame detailFrame = new MemberDetailFrame(member, memberDao, jenisMemberDao);
61
62             detailFrame.setVisible(b:true);
63         }
64     }
65 });
66
67 this.add(button);
68 this.add(textFieldNama);
69 this.add(labelInput);
70 this.add(scrollableTable);
71
72 this.setSize(width:400,height:500);
73 this.setLayout(manager:null);
74
75 }
76
77 public String getName(){
78     return textFieldNama.getText();
79 }
80
81 public void addJenisMember(JenisMember jenisMember){
82     tableModel.add(jenisMember);
83     textFieldNama.setText("");
84 }
85 }
```

Penjelasan

Kode ini membangun antarmuka pengguna menggunakan Java Swing untuk mengelola data jenis keanggotaan dalam aplikasi ORM (Object-Relational Mapping).

Membuat package view.member dan buat kelas MemberTableModel, MemberButtonSimpanActionListener, dan MemberFrame.

- MemberTableModel



```
1 package view.member;
2
3 import java.util.List;
4 import javax.swing.table.AbstractTableModel;
5 import model.Member;
6
7 public class MemberTableModel extends AbstractTableModel {
8     private String[] columnNames = { "Nama", "Jenis Member" };
9     private List<Member> data;
10
11     public MemberTableModel(List<Member> data) {
12         this.data = data;
13     }
14
15     public int getColumnCount() {
16         return columnNames.length;
17     }
18
19     public int getRowCount() {
20         return data.size();
21     }
22
23     public String getColumnName(int col){
24         return columnNames[col];
25     }
26
27     public Object getValueAt(int row, int col) {
28         Member rowItem = data.get(row);
```

```
30     String value = "";
31
32     switch(col) {
33     case 0:
34         value = rowItem.getNama();
35         break;
36     case 1:
37         // Add null check for getJenisMember()
38         if (rowItem.getJenisMember() != null) {
39             value = rowItem.getJenisMember().getNama();
40         } else {
41             value = "No Jenis Member"; // or any default value
42         }
43         break;
44     }
45
46     return value;
47 }
48
49
50 public boolean isCellEditable(int row, int col){
51     return false;
52 }
53
54 public void add(Member value){
55     data.add(value);
56     fireTableRowsInserted(data.size() - 1, data.size() - 1);
57 }
58
59 public void removeMember(Member member) {
60     data.remove(member);
61     fireTableRowsDeleted(data.size() - 1, data.size() - 1);
62 }
63
64 public void setData(List<Member> data) {
65     this.data = data;
66 }
67 }
```

- MemberButtonSimpanActionListener



```
1 package view.member;
2
3 import dao.MemberDao;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 import java.util.UUID;
7 import model.JenisMember;
8 import model.Member;
9
10 public class MemberButtonSimpanActionListener implements ActionListener {
11     private MemberFrame memberFrame;
12     private MemberDao memberDao;
13
14     public MemberButtonSimpanActionListener(MemberFrame memberFrame, MemberDao memberDao) {
15         this.memberFrame = memberFrame;
16         this.memberDao = memberDao;
17     }
18
19     public void actionPerformed(ActionEvent e){
20         String nama = this.memberFrame.getName();
21         if(nama.isEmpty()){
22             this.memberFrame.showAlert(message:"Nama tidak boleh kosong");
23         } else {
24             JenisMember jenisMember = this.memberFrame.getJenisMember();
25             Member member = new Member();
26             member.setId(UUID.randomUUID().toString());
27             member.setNama(nama);
28             member.setJenisMember(jenisMember);
29             member.setJenisMemberId(jenisMember.getId());
30             this.memberFrame.addMember(member);
31             this.memberDao.insert(member);
32         }
33     }
34 }
35 }
```

- MemberFrame



```
1 package view.member;
2
3 import dao.JenisMemberDao;
4 import dao.MemberDao;
5 import java.awt.event.MouseAdapter;
6 import java.awt.event.MouseEvent;
7 import java.util.List;
8 import javax.swing.JButton;
9 import javax.swing.JComboBox;
10 import javax.swing.JFrame;
11 import javax.swing.JLabel;
12 import javax.swing.JOptionPane;
13 import javax.swing.JScrollPane;
14 import javax.swing.JTable;
15 import javax.swing.JTextField;
16 import model.JenisMember; // Import class baru
17 import model.Member;
18 import memberdetail.MemberDetailFrame;
19
20 public class MemberFrame extends JFrame {
21
22     private List<JenisMember> jenisMemberList;
23     private List<Member> memberList;
24     private JTextField textFieldNama;
25     private MemberTableModel tableModel;
26     private JComboBox comboJenis;
27     private MemberDao memberDao;
28     private JenisMemberDao jenisMemberDao;
29     public MemberFrame(MemberDao memberDao, JenisMemberDao jenisMemberDao){
30         this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
31         this.memberDao = memberDao;
32         this.jenisMemberDao = jenisMemberDao;
33
34         this.memberList = this.memberDao.findAll();
35         this.jenisMemberList = this.jenisMemberDao.findAll();
36
37         JLabel labelInput = new JLabel(text:"Nama: ");
38         labelInput.setBounds(x:15,y:40,width:350,height:10);
39
40         textFieldNama = new JTextField();
41         textFieldNama.setBounds(x:15,y:60,width:350,height:30);
42
43         JLabel labelJenis = new JLabel(text:"Jenis Member: ");
44         labelJenis.setBounds(x:15,y:100,width:350,height:10);
```



```
46     comboJenis = new JComboBox();
47     comboJenis.setBounds(x:15,y:120,width:150,height:30);
48
49     JButton button = new JButton(text:"Simpan");
50     button.setBounds(x:15,y:160,width:100,height:40);
51
52     javax.swing.JTable table = new JTable();
53     JScrollPane scrollableTable = new JScrollPane(table);
54     scrollableTable.setBounds(x:15,y:210,width:350,height:200);
55
56     tableModel = new MemberTableModel(memberList);
57     table.setModel(tableModel);
58
59     MemberButtonSimpanActionListener actionListener = new MemberButtonSimpanActionListener(this, memberDao);
60
61     button.addActionListener(actionListener);
62
63     table.addMouseListener(new MouseAdapter() {
64     public void mouseClicked(MouseEvent e) {
65         if (e.getClickCount() == 2) { // Double click untuk membuka detail
66             int row = table.getSelectedRow();
67             Member selectedMember = memberList.get(row);
68             MemberDetailFrame detailFrame = new MemberDetailFrame(selectedMember, memberDao, jenisMemberDao);
69             detailFrame.populateComboJenis();
70             detailFrame.setVisible(b:true);
71         }
72     }
73     });
74
75     this.add(button);
76     this.add(textFieldNama);
77     this.add(labelInput);
78     this.add(scrollableTable);
79     this.add(labelJenis);
80     this.add(comboJenis);
81
82     this.setSize(width:400,height:500);
83     this.setLayout(manager:null);
84 }
```



```
85
86  public void populateComboJenis(){
87      this.jenisMemberList = this.jenisMemberDao.findAll();
88      comboJenis.removeAllItems();
89      for(JenisMember jenisMember: this.jenisMemberList){
90          comboJenis.addItem(jenisMember.getNama());
91      }
92  }
93
94  public String getName(){
95      return textFieldNama.getText();
96  }
97
98  public JenisMember getJenisMember(){
99      return jenisMemberList.get(comboJenis.getSelectedIndex());
100  }
101
102  public void addMember(Member member){
103      tableModel.add(member);
104      textFieldNama.setText("");
105  }
106
107  public void showAlert(String message){
108      JOptionPane.showMessageDialog(this, message);
109  }
110 }
```

Penjelasan

Kode ini memperlihatkan struktur Java Swing untuk aplikasi manajemen anggota, dengan berbagai komponen seperti tabel, form input, dan logika event handling. Kode ini mengikuti pola MVC (Model-View-Controller) yang baik, memisahkan logika bisnis (DAO dan ActionListener) dari tampilan (Swing GUI). Struktur ini memudahkan pemeliharaan dan pengembangan fitur baru.

Membuat package view.main lalu kelas MainButtonActionListener dan MainFrame

- MainButtonActionListener



```
1 package view.main;
2
3 import java.awt.event.ActionEvent;
4 import java.awt.event.ActionListener;
5
6 public class MainButtonActionListener implements ActionListener {
7     private MainFrame mainFrame;
8
9     public MainButtonActionListener(MainFrame mainFrame) {
10         this.mainFrame = mainFrame;
11     }
12
13     @Override
14     public void actionPerformed(ActionEvent e) {
15         if(e.getSource() == mainFrame.getButtonJenisMember()){
16             mainFrame.showJenisMemberFrame();
17         } else {
18             mainFrame.showMemberFrame();
19         }
20     }
21 }
22 }
```

- MainFrame



```
1 package view.main;
2
3 import dao.JenisMemberDao;
4 import dao.MemberDao;
5 import java.awt.FlowLayout;
6 import javax.swing.JButton;
7 import javax.swing.JFrame;
8 import view.jenisMember.JenisMemberFrame;
9 import view.member.MemberFrame;
10
11 public class MainFrame extends JFrame {
12     private JenisMemberFrame jenisMemberFrame ;
13     private MemberFrame memberFrame;
14     private JButton buttonJenisMember;
15     private JButton buttonMember;
16     private JenisMemberDao jenisMemberDao;
17     private MemberDao memberDao;
18
19     public MainFrame(JenisMemberDao jenisMemberDao, MemberDao memberDao){
20         this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21
22         this.setSize(width:400,height:500);
23
24         this.jenisMemberDao = jenisMemberDao;
25         this.memberDao = memberDao;
26
27         this.jenisMemberFrame = new JenisMemberFrame(jenisMemberDao);
28         this.memberFrame = new MemberFrame(memberDao, jenisMemberDao);
29
30         this.setLayout(new FlowLayout());
31         MainButtonActionListener actionListener = new MainButtonActionListener(this);
32
33         this.buttonJenisMember = new JButton(text:"Jenis Member: ");
34         this.buttonMember = new JButton(text:"Member");
```

```
36     this.buttonJenisMember.addActionListener(actionListener);
37     this.buttonMember.addActionListener(actionListener);
38
39     this.add(buttonJenisMember);
40     this.add(buttonMember);
41 }
42
43 public JButton getButtonJenisMember(){
44     return buttonJenisMember;
45 }
46
47 public JButton getButtonMember(){
48     return buttonMember;
49 }
50
51 public void showJenisMemberFrame(){
52     if(jenisMemberFrame == null){
53         jenisMemberFrame = new JenisMemberFrame(jenisMemberDao);
54     }
55     jenisMemberFrame.setVisible(b:true);
56 }
57
58
59 public void showMemberFrame(){
60
61     if(memberFrame == null){
62         memberFrame = new MemberFrame(memberDao, jenisMemberDao);
63     }
64     memberFrame.populateComboJenis();
65     memberFrame.setVisible(b:true);
66 }
67 }
```

Penjelasan

Kode ini merupakan bagian dari aplikasi manajemen anggota menggunakan Java Swing, yang berfungsi sebagai tampilan utama untuk mengakses dua modul: **Jenis Member** dan **Member**.

Kelas ini membangun sistem navigasi sederhana tetapi efektif di Java Swing, memisahkan logika event handling dari tampilan.

Penjelasan Kode

Jika berkas memiliki output, gunakan tabel ini.

Membuat package main dan kelas Main

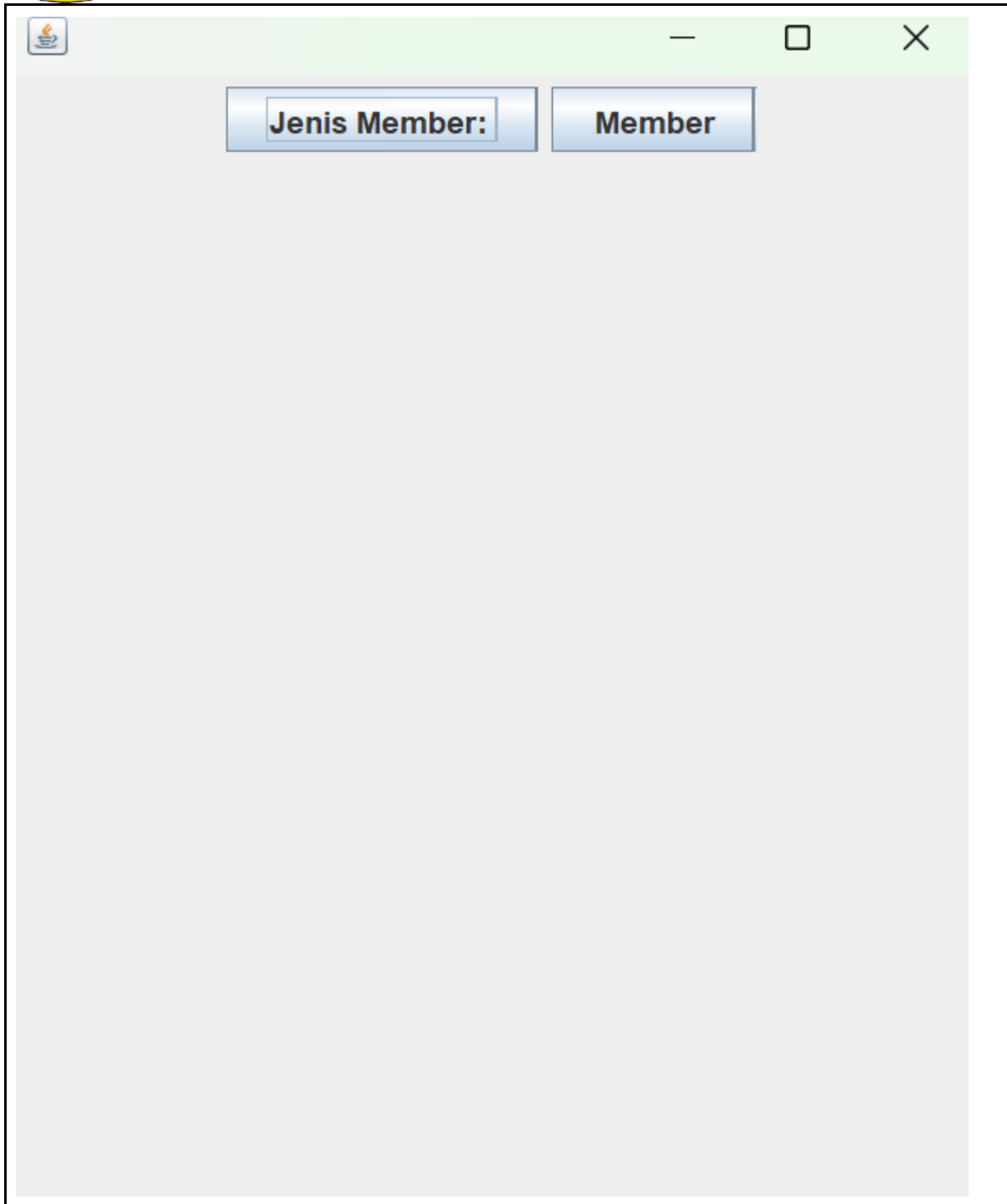


```
1 package main;
2
3 import dao.JenisMemberDao;
4 import dao.MemberDao;
5 import java.io.File;
6 import java.io.IOException;
7 import java.io.InputStream;
8 import org.apache.ibatis.io.Resources;
9 import org.apache.ibatis.session.SqlSessionFactory;
10 import org.apache.ibatis.session.SqlSessionFactoryBuilder;
11 import view.main.MainFrame;
12
13 public class Main {
14     public static void main(String[] args) throws IOException {
15         String resource = "mybatis-config.xml";
16         InputStream inputStream = Resources.getResourceAsStream(resource);
17         SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
18
19         JenisMemberDao jenisMemberDao = new JenisMemberDao(sqlSessionFactory);
20         MemberDao memberDao = new MemberDao(sqlSessionFactory);
21         MainFrame f = new MainFrame(jenisMemberDao, memberDao);
22
23         javax.swing.SwingUtilities.invokeLater(() -> {
24             f.setVisible(true);
25         });
26     }
27 }
```

Penjelasan


Kode di atas adalah file **Main.java**, yang berfungsi sebagai titik masuk (*entry point*) dari aplikasi Membership ORM berbasis Java dengan integrasi MyBatis.

Output



The image shows a screenshot of a web application window. The window has a light green title bar with a small icon on the left and standard window controls (minimize, maximize, close) on the right. The main content area is light gray and contains two blue buttons at the top. The first button is labeled "Jenis Member:" and the second button is labeled "Member". The rest of the page is empty.




—□×

Nama:

Simpan

Nama
Kholish



—□×

Nama:

Jenis Member:

VVIP ▼

Simpan

Nama	Jenis Member
Olis	VVIP