**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/Pertemuan9/MembershipORM

**Penjelasan Kode**

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

|  |
| --- |
| **MemberDao.java** |
| public class MemberDao {      private final  SqlSessionFactory sqlSessionFactory;      public MemberDao(SqlSessionFactory sqlSessionFactory){          this.sqlSessionFactory = sqlSessionFactory;      }      public int insert(Member member){          int result;          try (SqlSession sqlSession = sqlSessionFactory.openSession()){              result = sqlSession.insert("mapper.member.insert", member);          }          return result;      }      public int update(Member member){          int result;          try (SqlSession sqlSession = sqlSessionFactory.openSession()){              result = sqlSession.update("mapper.member.update", member);          }          return result;      }      public int delete(Member member){          int result;          try (SqlSession sqlSession = sqlSessionFactory.openSession()){              result = sqlSession.delete("mapper.member.delete", member);          }          return result;      }      public List<Member> findAll(){          list<Member> result;          try (SqlSession sqlSession = sqlSessionFactory.openSession()){              result = sqlSession.selectList("mapper.member.findAll");          }          return result;      }  } |
| **Penjelasan** |
| File MemberDao.java mengatasi data Member, termasuk operasi CRUD ke database. |
| **JenisMemberDao.java** |
| public class JenisMemberDao {       private final SqlSessionFactory sqlSessionFactory;      public JenisMemebrDao(SqlSessionFactory sqlSessionFactory){          this.sqlSessionFactory = sqlSessionFactory;      }      public int insert(JenisMember jenisMember){          int result;          try(SqlSession session = sqlSessionFactory.openSession()){              result = session.insert("mapper.jenisMemberMapper.insert", jenisMember);          }          return result;      }      public int update(JenisMember jenisMember){          int result;          try(SqlSession session = sqlSessionFactory.openSession()){              result = session.update("mapper.jenisMemberMapper.update", jenisMember);          }          return result;      }      public int delete(JenisMember jenisMember){          int result;          try(SqlSession session = sqlSessionFactory.openSession()){              result = session.delete("mapper.jenisMemberMapper.delete", jenisMember);          }          return result;      }      public List<JenisMember> findAll(){          List<JenisMember> result;          try(SqlSqssion session = sqlSessionFactory.openSession()){              result = session.selectList("mapper.jenisMemberMapper.findAll");          }          return result;      }  } |

|  |
| --- |
| **Penjelasan** |
| File JenisMemberDao.java menangani data JenisMember, termasuk operasi CRUD di dalam database. |
| **Member.java** |
| private 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 String getJenisMemberId() {          return JenisMemberId;      }      public void setJenisMemberId(String jenisMemberId) {          JenisMemberId = jenisMemberId;      }      public void setJenisMemberId(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;      private 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;      }  } |
| **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(JenisMemberDao jenisMemberDao, MemberDao memberDao){          this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);          this.setSize(400,500);          this.jenisMemberDao = jenisMemberDao;          this.memberDao = 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 = new MemberFrame(memberDao, jenisMemberDao);          }          memberFrame.populateComboJenis();          memberFrame.setVisible(true);      }} |

|  |
| --- |
| **Penjelasan** |
| .File MainFrame berfungsi sebagai UI yang menampilkan semua tombol yang berfungsi untuk mengakses JenisMemberFrame dan MemberFrame |
| **MemberFrame.java** |

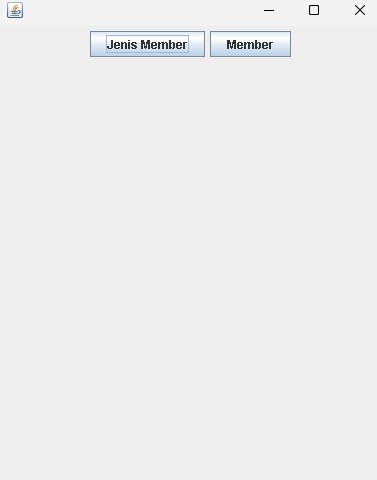
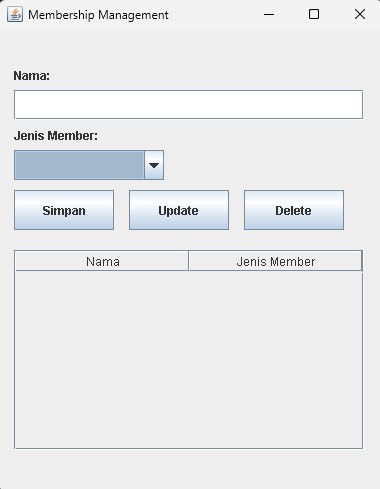
|  |
| --- |
| public class MemberFrame extends JFrame{      private List<JenisMember> jenisMemberList;      private List<Member> memberList;      private JTextField textFieldNama;      private MemberTableModel tableModel;      private JComboBox<E> comboJenis;      private MemberDao memberDao;      private JenisMemberDao jenisMemberDao;      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();          JLabel labelInput = new JLabel("Nama");          labelInput.setBounds(15,40,350,10);          textFieldNama = new JTextField();          textFieldNama.setBounds(15,60,350,30);          JLabel labelJenis = new JLabel("Jenis Member:");          labelJenis.setBounds(15,100,350,10);          comboJenis = new JComboBox();          comboJenis.setBounds(15,120,150,30);          javax.swing.JTable table = new JTable();          JScrollPane scrollableTable = new JScrollPane(table);          scrollableTable.setBounds(15, 220, 350, 200);          tableModel = new MemberTableModel(memberList);          table.setModel(TableModel);          MemberButtonSimpanActionListener actionListener              = new MemberButtonSimpanActionListener(this, memberDao);          button.addActionListener(actionListener);          this.add(button);          this.add(textFieldNama);          this.add(labelInput);          this.add(labelJenis);          this.add(comboJenis);          this.add(scrollableTable);          this.setSize(400,500);          this.setLayout(null);      }      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);      }  } |

|  |
| --- |
| 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** |
| public class MemberTableModel extends AbstractTableModel{      private String[] columnNames = {"Nama", "Jenis Member"};      private 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.getName();                  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);      }  } |

|  |
| --- |
| **Penjelasan** |
| MemberTableModel berfungsi sebagai controller dalam penampilan tabel |
| **MemberButtonSimpanActionListener.java** |

|  |
| --- |
| public class MemberButtonSimpanActionListener implements ActionListener{      private MemberFrame memberFrame;      private MemberDao memberDao;      public MemberButtonSimpanActionListener(MemberFrame memberFrame, MemberDao memberDao) {          this.memberFaame = memberFrame;          this.memberDao = memberDao;      }      @Override      public void actionPerformed(ActionEvent e) {          String nama = this.memberFrame.getNama();          if(nama.isEmpty()){              this.memberFrame.showAlert("Nama tidak boleh kosong");          }else{              JenisMember jenisMember = this.memberFrame.getJenisMember();              Member member = new Member();              member.setId(UUID.randomUUID().toString());              member.setNama(nama);              member.setJenisMember(jenisMember);              member.setJenisMemberId(jenisMember.getId());              this.memberFrame.addMember(member);              this.memberDao.insert(member);          }      }  } |
| **Penjelasan** |
| MemberButtonSimpanActionListener berfungsi untuk menetapkan fungsi untuk button |
| **Output** |



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