



# Object-Relational Mapping

Praktikum Pemrograman II - 08

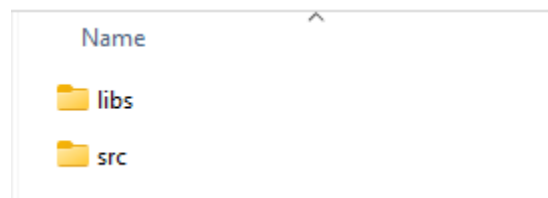


## Object Relational Mapping

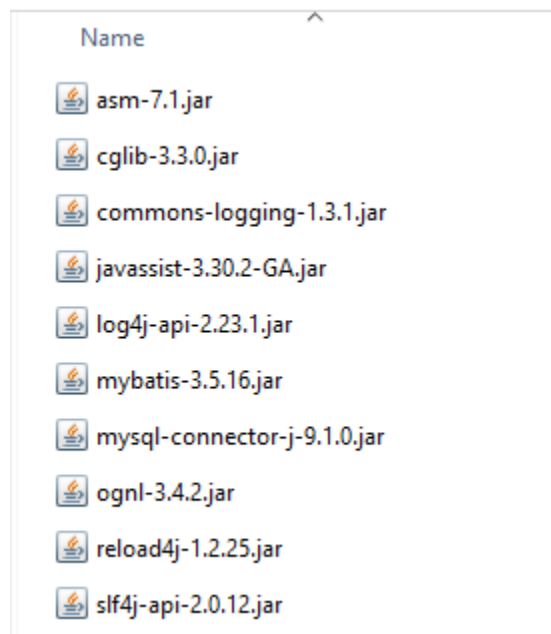
Dalam Java Standard Edition (Java SE), terdapat beberapa library Object-Relational Mapping (ORM) yang dapat digunakan untuk mengelola interaksi antara aplikasi Java dan database relasional. Salah satu library yang populer adalah MyBatis. MyBatis adalah framework ORM yang lebih fleksibel dibandingkan beberapa ORM lain karena memungkinkan kontrol lebih besar terhadap query SQL. MyBatis sangat cocok jika kita membutuhkan ORM tetapi tetap ingin menulis query SQL secara manual.

### Latihan 1

Kita masih akan menggunakan basisdata pp2\_membership yang telah kita buat di modul 7. Pertama-tama buat folder/project baru, misal dengan nama MembershipORM. Kemudian buat 2 folder dengan nama src dan libs. Extract jar yang disediakan ke dalam folder libs, sehingga struktur folder akan terlihat sebagai berikut.



Gambar 1 Struktur Luar Proyek



Gambar 2 Isi Folder Libs

Kemudian, buat file dengan nama mybatis-config.xml dengan isi sebagai berikut.

```
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="1234" />
14       </dataSource>
15     </environment>
16   </environments>
17   <mappers>
18     <package name="mapper"/>
19   </mappers>
20 </configuration>
21
```

Sesuaikan properti dataSource dengan konfigurasi lokal anda.

Selanjutnya, buat folder/package model lalu buat dua kelas yaitu Member dan JenisMember dengan kode sebagai berikut.

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

```
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    {
11        this.id = id;
12    }
13
14    public String getId()
15    {
16        return id;
17    }
18
19    public void setName(String nama)
20    {
21        this.nama = nama;
22    }
23
24    public String getName()
25    {
26        return nama;
27    }
28
29    public String getJenisMemberId() {
30        return jenisMemberId;
31    }
32
33    public void setJenisMemberId(String jenisMemberId) {
34        this.jenisMemberId = jenisMemberId;
35    }
36
```

```

36
37     public void setJenisMember(JenisMember jenisMember)
38     {
39         this.jenisMember = jenisMember;
40     }
41
42     public JenisMember getJenisMember()
43     {
44         return jenisMember;
45     }
46
47 }
48

```

Kemudian, buat package mapper, lalu buat dua kelas dengan nama MemberMapper dan 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
14     @Insert("Insert into jenis_member (id, nama) values(#{id},#{nama})")
15     public Integer insert(JenisMember jenisMember);
16
17     @Update("update jenis_member set nama = #{nama} where id = #{id}")
18     public Integer update(JenisMember jenisMember);
19
20     @Delete("delete from jenis_member where id = #{id}")
21     public Integer delete(JenisMember jenisMember);
22
23     @Select("SELECT * FROM jenis_member")
24     @Results(value = {
25         @Result(property = "id", column = "id"),
26         @Result(property = "nama", column = "nama")})
27     List<JenisMember> findAll();
28
29 }
30

```

```

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(JenisMember jenisMember);
21
22     @Delete("delete from member where id = #{id}")
23     public Integer delete(JenisMember jenisMember);
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, one = @One(select = "getJenisMember"))})
31     List<Member> findAll();
32
33     @Select("SELECT * FROM jenis_member WHERE id = #{id}")
34     @Results(value = {
35         @Result(property = "id", column = "id"),
36         @Result(property = "nama", column = "nama")})
37     JenisMember getJenisMember(String jenisMemberId);
38 }
39

```

Lalu, buatlah package dao dengan kelas MemberDao dan 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         }
21
22         return result;
23     }
24
25     public int update(JenisMember jenisMember) {
26         int result;
27         try (SqlSession session = sqlSessionFactory.openSession()) {
28             result = session.update("mapper.JenisMemberMapper.update", jenisMember);
29         }
30         return result;
31     }
32

```

```

32
33     public int delete(JenisMember jenisMember) {
34         int result;
35         try (SqlSession session = sqlSessionFactory.openSession()) {
36             result = session.delete("mapper.JenisMemberMapper.delete", jenisMember);
37         }
38         return result;
39     }
40
41     public List<JenisMember> findAll() {
42         List<JenisMember> result;
43         try (SqlSession session = sqlSessionFactory.openSession()) {
44             result = session.selectList("mapper.JenisMemberMapper.findAll");
45         }
46
47         return result;
48     }
49
50 }
51

```



```

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         }
20         return result;
21     }
22
23     public int update(Member member) {
24         int result;
25         try (SqlSession session = sqlSessionFactory.openSession()) {
26             result = session.update("mapper.MemberMapper.update", member);
27         }
28         return result;
29     }
30

```

```

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

```

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

```
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     {
13         this.data = data;
14     }
15
16     public int getColumnCount() {
17         return columnNames.length;
18     }
19
20     public int getRowCount() {
21         return data.size();
22     }
23
24     public String getColumnName(int col) {
25         return columnNames[col];
26     }
27 }
```

```
27
28 public Object getValueAt(int row, int col) {
29     JenisMember rowItem = data.get(row);
30     String value = "";
31     switch (col)
32     {
33         case 0:
34             value = rowItem.getNama();
35             break;
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 {
46     data.add(value);
47     fireTableRowsInserted(data.size() - 1, data.size() - 1);
48 }
49 }
50
```

```

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 public class JenisMemberButtonSimpanActionListener implements ActionListener {
10
11     private JenisMemberFrame jenisMemberFrame;
12     private JenisMemberDao jenisMemberDao;
13
14     public JenisMemberButtonSimpanActionListener(JenisMemberFrame jenisMemberFrame,
15         JenisMemberDao jenisMemberDao) {
16         this.jenisMemberFrame = jenisMemberFrame;
17         this.jenisMemberDao = jenisMemberDao;
18     }
19
20     @Override
21     public void actionPerformed(ActionEvent e) {
22         String nama = this.jenisMemberFrame.getNama();
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

```

```

1  package view.jenismember;
2
3  import dao.JenisMemberDao;
4  import java.util.List;
5  import javax.swing.JButton;
6  import javax.swing.JFrame;
7  import javax.swing.JLabel;
8  import javax.swing.JScrollPane;
9  import javax.swing.JTable;
10 import javax.swing.JTextField;
11 import model.JenisMember;
12
13 public class JenisMemberFrame extends JFrame {
14
15     private List<JenisMember> jenisMemberList;
16     private JTextField textFieldNama;
17     private JenisMemberTableModel tableModel;
18     private JenisMemberDao jenisMemberDao;
19
20     public JenisMemberFrame(JenisMemberDao jenisMemberDao) {
21         this.jenisMemberDao = jenisMemberDao;
22         this.jenisMemberList = jenisMemberDao.findAll();
23         this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
24
25         JLabel labelInput = new JLabel("Nama:");
26         labelInput.setBounds(15, 40, 350, 10);
27
28         textFieldNama = new JTextField();
29         textFieldNama.setBounds(15, 60, 350, 30);
30
31         JButton button = new JButton("Simpan");
32         button.setBounds(15, 100, 100, 40);
33
34         javax.swing.JTable table = new JTable();
35         JScrollPane scrollableTable = new JScrollPane(table);
36         scrollableTable.setBounds(15, 150, 350, 200);
37

```

```

37
38     tableModel = new JenisMemberTableModel(jenisMemberList);
39     table.setModel(tableModel);
40
41     JenisMemberButtonSimpanActionListener actionListener =
42     new JenisMemberButtonSimpanActionListener(this, jenisMemberDao);
43
44     button.addActionListener(actionListener);
45
46     this.add(button);
47     this.add(textFieldNama);
48     this.add(labelInput);
49     this.add(scrollableTable);
50
51     this.setSize(400, 500);
52     this.setLayout(null);
53 }
54
55 public String getNama() {
56     return textFieldNama.getText();
57 }
58
59 public void addJenisMember(JenisMember jenisMember) {
60     tableModel.add(jenisMember);
61     textFieldNama.setText("");
62 }
63
64 }

```

Kemudian, buatlah package view.member dan buat kelas MemberTableModel, MemberButtonSimpanActionListener, dan MemberFrame.

```
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     {
13         this.data = data;
14     }
15
16     public int getColumnCount() {
17         return columnNames.length;
18     }
19
20     public int getRowCount() {
21         return data.size();
22     }
23
24     public String getColumnName(int col) {
25         return columnNames[col];
26     }
27 }
```

```

27
28 public Object getValueAt(int row, int col) {
29     Member rowItem = data.get(row);
30     String value = "";
31     switch (col)
32     {
33         case 0:
34             value = rowItem.getNama();
35             break;
36         case 1:
37             value = rowItem.getJenisMember().getNama();
38             break;
39     }
40     return value;
41 }
42
43 public boolean isCellEditable(int row, int col) {
44     return false;
45 }
46
47 public void add(Member value)
48 {
49     data.add(value);
50     fireTableRowsInserted(data.size() - 1, data.size() - 1);
51 }
52
53 }
54

```



```

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
12     private MemberFrame memberFrame;
13     private MemberDao memberDao;
14
15     public MemberButtonSimpanActionListener(MemberFrame memberFrame, MemberDao memberDao) {
16         this.memberFrame = memberFrame;
17         this.memberDao = memberDao;
18     }
19
20     @Override
21     public void actionPerformed(ActionEvent e) {
22         String nama = this.memberFrame.getNama();
23         if (nama.isEmpty()) {
24             this.memberFrame.showAlert("Nama tidak boleh kosong");
25         } else {
26             JenisMember jenisMember = this.memberFrame.getJenisMember();
27             Member member = new Member();
28             member.setId(UUID.randomUUID().toString());
29             member.setNama(nama);
30             member.setJenisMember(jenisMember);
31             member.setJenisMemberId(jenisMember.getId());
32             this.memberFrame.addMember(member);
33             this.memberDao.insert(member);
34         }
35     }
36 }
37
38

```

```

1 package view.member;
2
3 import dao.JenisMemberDao;
4 import dao.MemberDao;
5 import java.util.List;
6 import javax.swing.JButton;
7 import javax.swing.JComboBox;
8 import javax.swing.JFrame;
9 import javax.swing.JLabel;
10 import javax.swing.JOptionPane;
11 import javax.swing.JScrollPane;
12 import javax.swing.JTable;
13 import javax.swing.JTextField;
14 import model.JenisMember;
15 import model.Member;
16
17 public class MemberFrame extends JFrame {
18     private List<JenisMember> jenisMemberList;
19     private List<Member> memberList;
20     private JTextField textFieldNama;
21     private MemberTableModel tableModel;
22     private JComboBox comboJenis;
23     private MemberDao memberDao;
24     private JenisMemberDao jenisMemberDao;
25
26     public MemberFrame(MemberDao memberDao, JenisMemberDao jenisMemberDao){
27
28         this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
29
30         this.memberDao = memberDao;
31         this.jenisMemberDao = jenisMemberDao;
32
33         this.memberList = this.memberDao.findAll();
34         this.jenisMemberList = this.jenisMemberDao.findAll();
35
36         JLabel labelInput = new JLabel("Nama:");
37         labelInput.setBounds(15,40,350,10);
38

```

```

38
39     textFieldNama = new JTextField();
40     textFieldNama.setBounds(15,60,350,30);
41
42     JLabel labelJenis = new JLabel("Jenis Member:");
43     labelJenis.setBounds(15,100,350,10);
44
45
46     comboJenis = new JComboBox();
47     comboJenis.setBounds(15,120,150,30);
48
49     JButton button = new JButton("Simpan");
50     button.setBounds(15,160,100,40);
51
52     javax.swing.JTable table = new JTable();
53     JScrollPane scrollableTable = new JScrollPane(table);
54     scrollableTable.setBounds(15,220,350,200);
55
56     tableModel = new MemberTableModel(memberList);
57     table.setModel(tableModel);
58
59     MemberButtonSimpanActionListener actionListener
60         = new MemberButtonSimpanActionListener(this, memberDao);
61
62     button.addActionListener(actionListener);
63
64     this.add(button);
65     this.add(textFieldNama);
66     this.add(labelInput);
67     this.add(labelJenis);
68     this.add(comboJenis);
69     this.add(scrollableTable);
70
71     this.setSize(400,500);
72     this.setLayout(null);
73 }
74

```

```

74
75     public void populateComboJenis()
76     {
77         this.jenisMemberList = this.jenisMemberDao.findAll();
78         comboJenis.removeAllItems();
79         for (JenisMember jenisMember: this.jenisMemberList)
80         {
81             comboJenis.addItem(jenisMember.getNama());
82         }
83     }
84
85     public String getNama()
86     {
87         return textFieldNama.getText();
88     }
89
90     public JenisMember getJenisMember()
91     {
92         return jenisMemberList.get(comboJenis.getSelectedIndex());
93     }
94
95     public void addMember(Member member)
96     {
97         tableModel.add(member);
98         textFieldNama.setText("");
99     }
100
101     public void showAlert(String message)
102     {
103         JOptionPane.showMessageDialog(this, message);
104     }
105 }
106

```

Kemudian, buat package view.main lalu kelas MainButtonActionListener dan MainFrame.

```
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
8     private MainFrame mainFrame;
9
10    public MainButtonActionListener(MainFrame mainFrame) {
11        this.mainFrame = mainFrame;
12    }
13
14    @Override
15    public void actionPerformed(ActionEvent e) {
16        if (e.getSource() == mainFrame.getButtonJenisMember()) {
17            mainFrame.showJenisMemberFrame();
18        } else {
19            mainFrame.showMemberFrame();
20        }
21    }
22 }
23
```

```

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(400,500);
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("Jenis Member");
34         this.buttonMember = new JButton("Member");
35
36         this.buttonJenisMember.addActionListener(actionListener);
37         this.buttonMember.addActionListener(actionListener);
38

```

```

38
39     this.add(buttonJenisMember);
40     this.add(buttonMember);
41 }
42
43 public JButton getButtonJenisMember()
44 {
45     return buttonJenisMember;
46 }
47
48 public JButton getButtonMember()
49 {
50     return buttonMember;
51 }
52
53 public void showJenisMemberFrame()
54 {
55     if (jenisMemberFrame == null)
56     {
57         jenisMemberFrame = new JenisMemberFrame(jenisMemberDao);
58     }
59     jenisMemberFrame.setVisible(true);
60 }
61
62 public void showMemberFrame()
63 {
64     if (memberFrame == null)
65     {
66         memberFrame = new MemberFrame(memberDao, jenisMemberDao);
67     }
68     memberFrame.populateComboJenis();
69     memberFrame.setVisible(true);
70 }
71 }
72

```

Kemudian, buat package main dan kelas Main sebagai berikut.

```

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
16         String resource = "mybatis-config.xml";
17         InputStream inputStream = Resources.getResourceAsStream(resource);
18         SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(inputStream);
19
20         JenisMemberDao jenisMemberDao = new JenisMemberDao(sqlSessionFactory);
21         MemberDao memberDao = new MemberDao(sqlSessionFactory);
22         MainFrame f = new MainFrame(jenisMemberDao, memberDao);
23
24         javax.swing.SwingUtilities.invokeLater(() -> {
25             f.setVisible(true);
26         });
27     }
28 }
29

```

Selanjutnya, compile kelas Main dengan perintah berikut

```
javac -cp "C:\Users\wdgus\Documents\Unpas\Praktikum Prog
II\MembershipORM\src;C:\Users\wdgus\Documents\Unpas\Praktikum Prog II\MembershipORM\libs\*"
src/main/Main.java
```

Kita perlu meng-compile kelas di dalam package mapper karena kelas-kelas tersebut tidak memiliki dependency langsung dengan kelas Main.

```
javac -cp "C:\Users\wdgus\Documents\Unpas\Praktikum Prog
II\MembershipORM\src;C:\Users\wdgus\Documents\Unpas\Praktikum Prog II\MembershipORM\libs\*"
src/mapper/*.java
```

Kemudian, jalankan kelas Main dengan perintah berikut.

```
java -cp "C:\Users\wdgus\Documents\Unpas\Praktikum Prog
II\MembershipORM\src;C:\Users\wdgus\Documents\Unpas\Praktikum Prog II\MembershipORM\libs\*"
main.Main
```

## Latihan 2

Lengkapi implementasi update dan delete data pada aplikasi di Latihan 1.