

MODUL 13.3

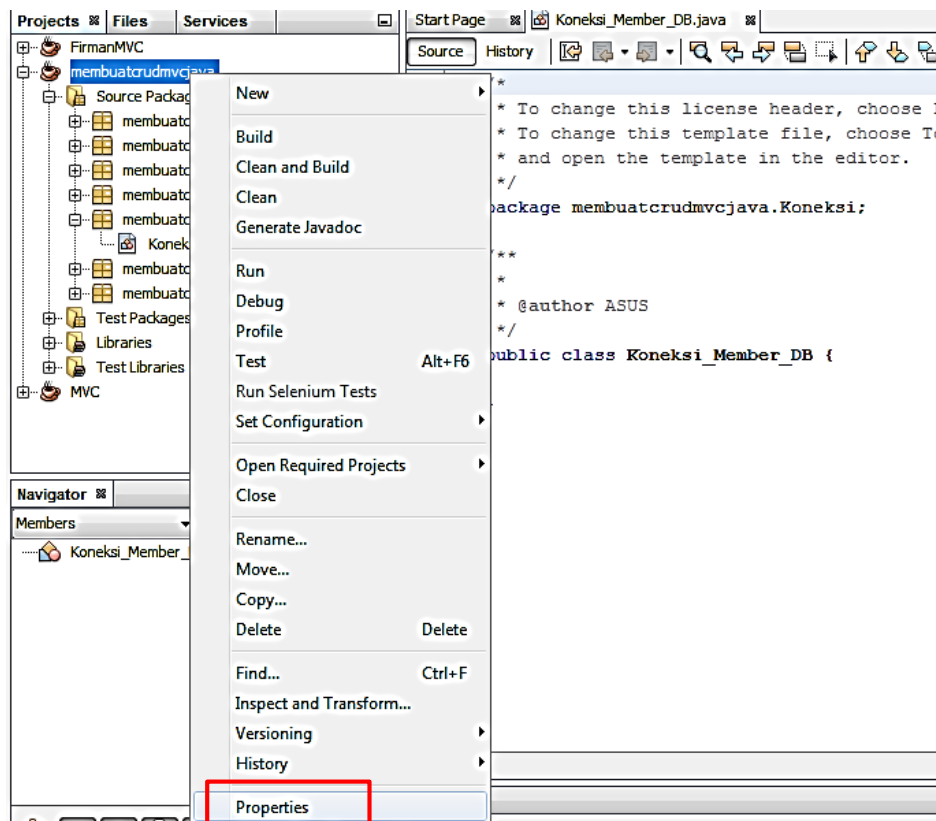
“KONEKSI DATABASE, DAO, DAO IMPLEMENT (MVC)”



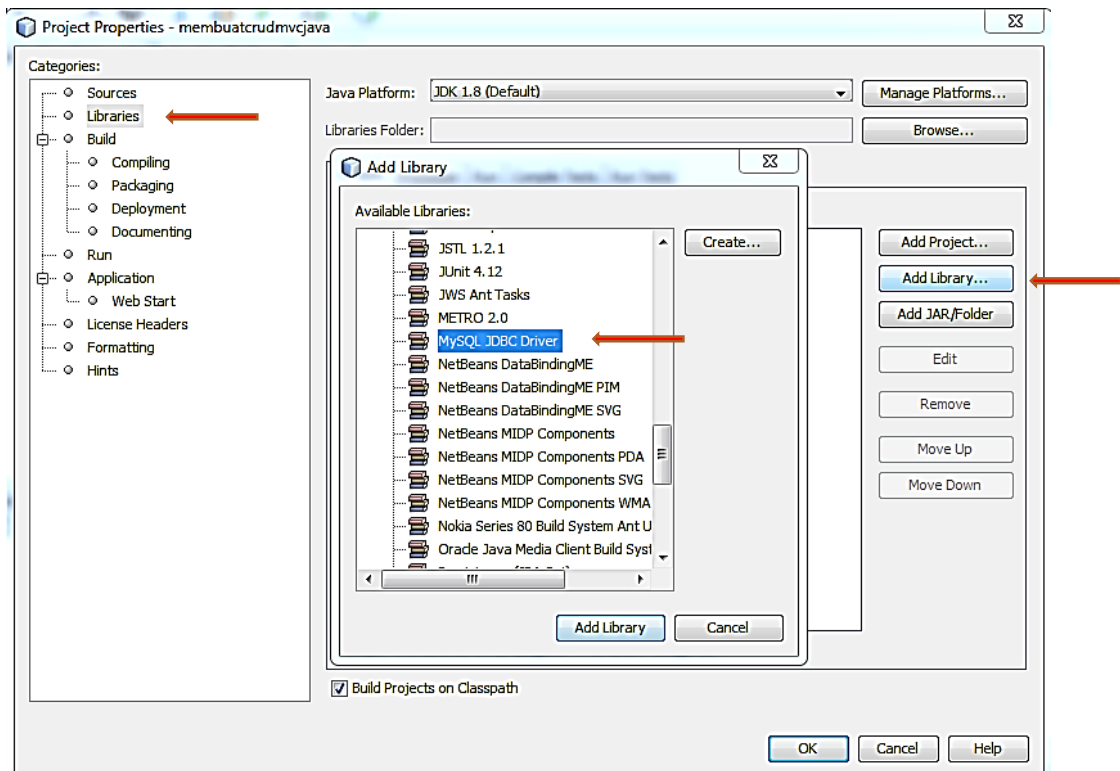
Oleh
SAFIRA MAYA SHOVIE, S.Pd

A. KONEKSI DATABASE

1. Aktifkan Xampp dan Buka localhost/phpmyadmin
2. Masukkan Library MySQL JDBC Driver, **Klik kanan Project - Properties**

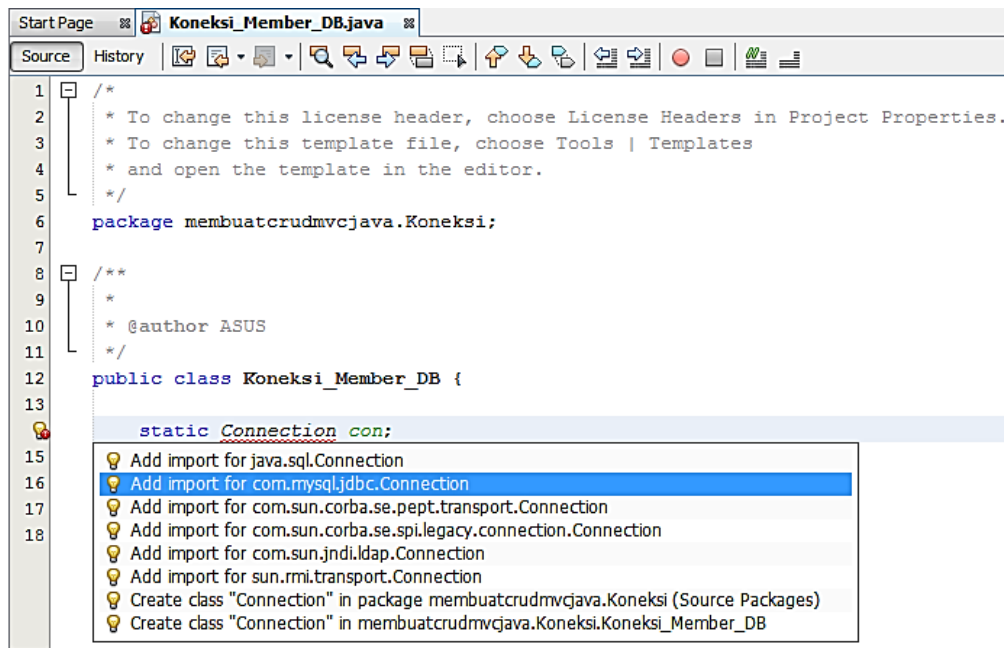


Klik **Libraries** – **Add Library** – **Pilih MySQL JDBC Driver** – **Klik Add Library** – **OK**.



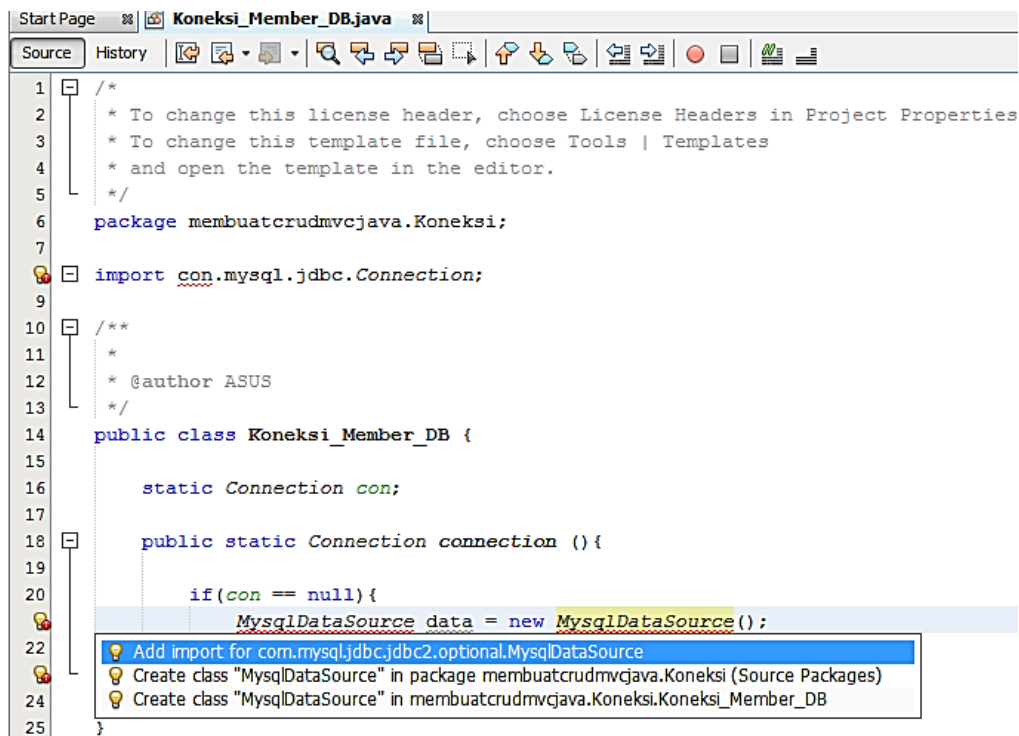
3. Masukkan Syntax pada Source Package membuatcrudmvcjava.Koneksi

Pada bagian yang terdapat tanda pentung merah, silahkan di klik dan dibenarkan seperti berikut



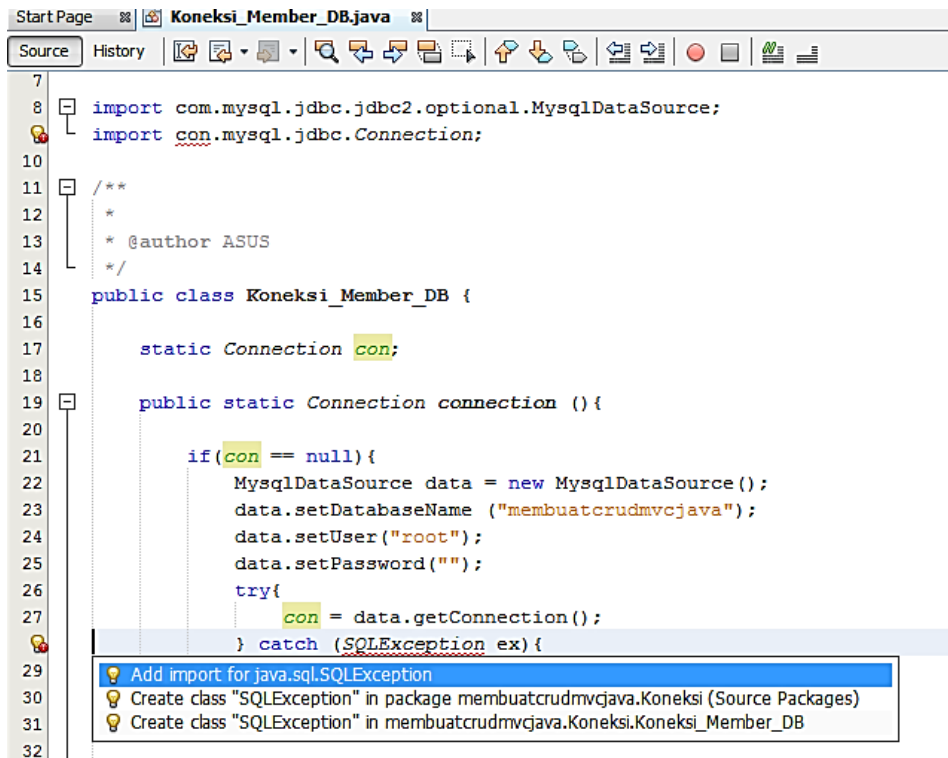
```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package membuatcrudmvcjava.Koneksi;
7
8  /**
9   *
10  * @author ASUS
11  */
12  public class Koneksi_Member_DB {
13
14      static Connection con;
15
16      // Add import for java.sql.Connection
17      // Add import for com.mysql.jdbc.Connection
18      // Add import for com.sun.corba.se.pept.transport.Connection
19      // Add import for com.sun.corba.se.spi.legacy.connection.Connection
20      // Add import for com.sun.jndi.ldap.Connection
21      // Add import for sun.rmi.transport.Connection
22      // Create class "Connection" in package membuatcrudmvcjava.Koneksi (Source Packages)
23      // Create class "Connection" in membuatcrudmvcjava.Koneksi.Koneksi_Member_DB
```

Silahkan melanjutkan menuliskan syntax. Dan apabila ada peringatan tanda pentung merah, maka rubahlah dan sesuaikan dengan berikut ini



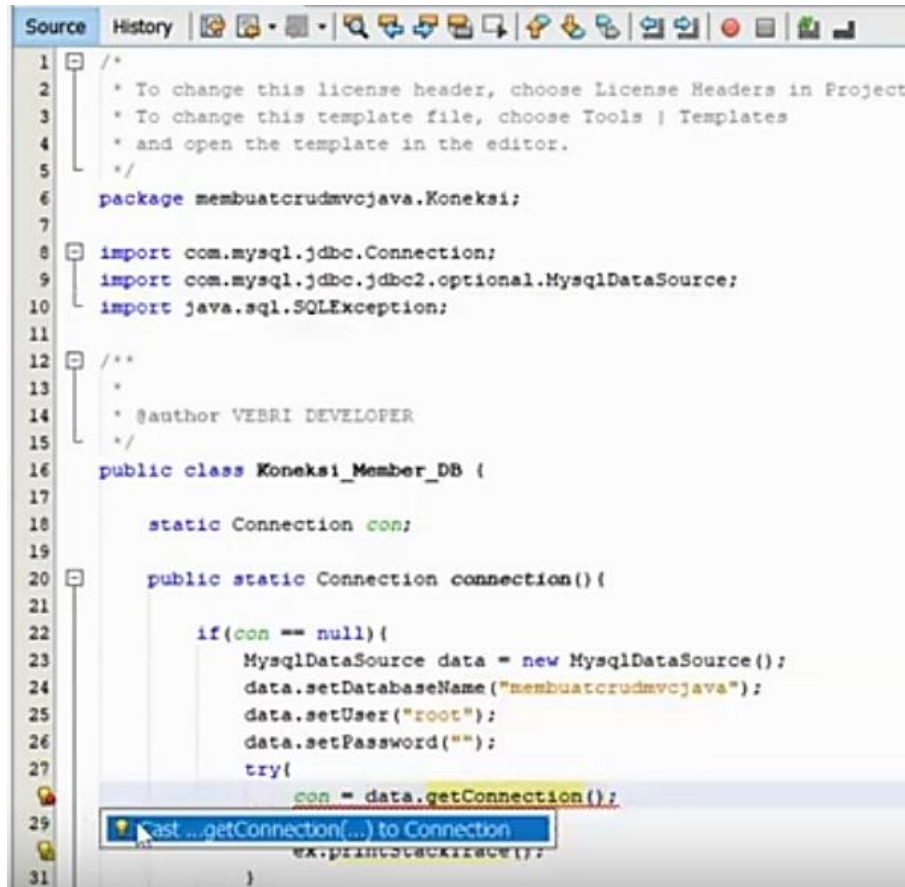
```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package membuatcrudmvcjava.Koneksi;
7
8  import com.mysql.jdbc.Connection;
9
10 /**
11  *
12  * @author ASUS
13  */
14 public class Koneksi_Member_DB {
15
16     static Connection connection;
17
18     public static Connection connection () {
19
20         if(connection == null){
21             MysqlDataSource data = new MysqlDataSource();
22
23             // Add import for com.mysql.jdbc.jdbc2.optional.MysqlDataSource
24             // Create class "MysqlDataSource" in package membuatcrudmvcjava.Koneksi (Source Packages)
25             // Create class "MysqlDataSource" in membuatcrudmvcjava.Koneksi.Koneksi_Member_DB
26         }
27     }
28 }
```

Pada bagian yang terdapat tanda pentung merah/ } *catch (SQLException ex){*, silahkan di klik dan dibenarkan seperti berikut



```
7
8 import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
9 import com.mysql.jdbc.Connection;
10
11 /**
12  *
13  * @author ASUS
14  */
15 public class Koneksi_Member_DB {
16
17     static Connection con;
18
19     public static Connection connection () {
20
21         if (con == null) {
22             MysqlDataSource data = new MysqlDataSource();
23             data.setDatabaseName ("membuatcrudmvcjava");
24             data.setUser ("root");
25             data.setPassword("");
26             try {
27                 con = data.getConnection();
28             } catch (SQLException ex) {
29
30
31
32
```

Pada *con = data.getConnection();*, silahkan dirubah disesuaikan berikut



```
1 /**
2  * To change this license header, choose License Headers in Project
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6 package membuatcrudmvcjava.Koneksi;
7
8 import com.mysql.jdbc.Connection;
9 import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
10 import java.sql.SQLException;
11
12 /**
13  *
14  * @author VIBRI DEVELOPER
15  */
16 public class Koneksi_Member_DB {
17
18     static Connection con;
19
20     public static Connection connection() {
21
22         if (con == null) {
23             MysqlDataSource data = new MysqlDataSource();
24             data.setDatabaseName ("membuatcrudmvcjava");
25             data.setUser ("root");
26             data.setPassword("");
27             try {
28                 con = data.getConnection();
29             }
30             ex.printStackTrace();
31         }
32
```

4. Lanjutkan selesaikan penulisan syntax seperti berikut. Kemudian run untuk mengetahui program berhasil atau belum.

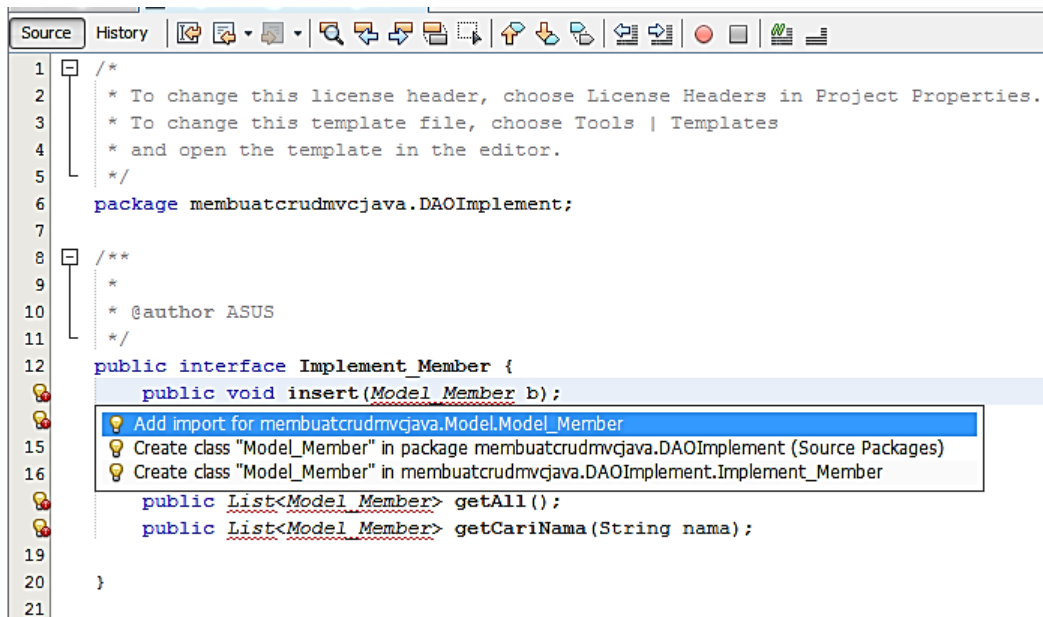
```
6 package membuatcrudmvcjava.Koneksi;
7 import com.mysql.jdbc.Connection;
8 import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
9 import java.sql.SQLException;
10 /**
11  *
12  * @author ASUS
13  */
14 public class Koneksi_Member_DB {
15
16     static Connection con;
17
18     public static Connection connection () {
19
20         if(con == null){
21             MysqlDataSource data = new MysqlDataSource();
22             data.setDatabaseName ("membuatcrudmvcjava");
23             data.setUser("root");
24             data.setPassword("");
25             try{
26                 con = (Connection) data.getConnection();
27             } catch (SQLException ex) {
28                 ex.printStackTrace();
29             }
30         }
31         return con;
32     }
33 }
```

B. MENGISI SYNTAX DAO IMPLEMENT

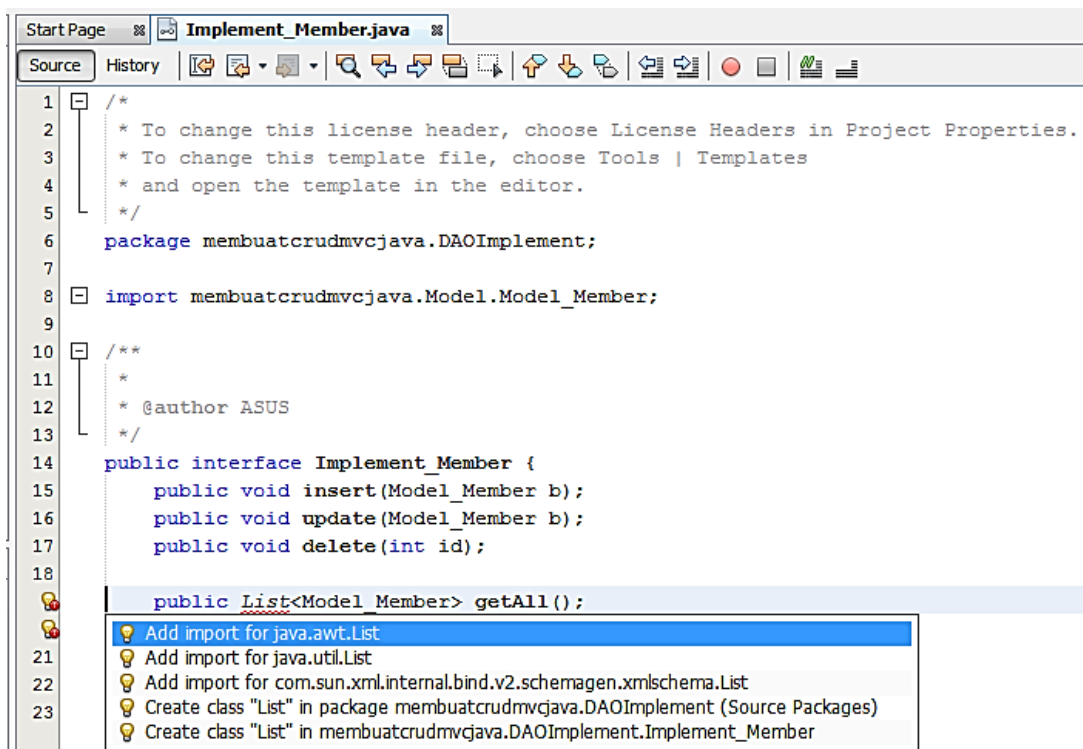
1. Tuliskan syntax berikut

```
6 package membuatcrudmvcjava.DAOImplement;
7
8 /**
9  *
10  * @author ASUS
11  */
12 public interface Implement_Member {
13     public void insert(Model Member b);
14     public void update(Model Member b);
15     public void delete(int id);
16
17     public List<Model Member> getAll();
18     public List<Model Member> getCariNama(String nama);
19
20 }
```

Karena ada error, silahkan lakukan pembenaran seperti berikut. Klik pada baris **public void insert(Model_Member b);**, jika terdapat pilihan, pilihlah baris 1 **Add import**.



Selanjutnya pada bagian `public List<Model_Member> getAll();`, silahkan diganti **Add import for java.awt.List**

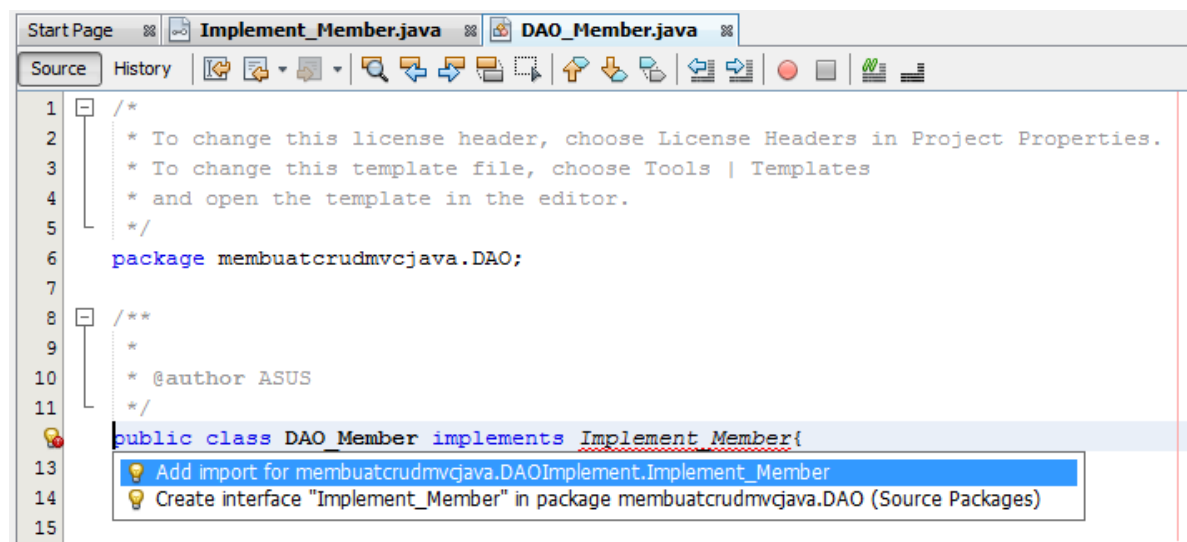


2. Samakan syntaxnya seperti berikut

```
6 package membuatcrudmvcjava.DAOImplement;  
7  
8 import java.awt.List;  
9 import membuatcrudmvcjava.Model.Model_Member;  
10  
11 /**  
12  *  
13  * @author ASUS  
14  */  
15 public interface Implement_Member {  
16  
17     public void insert(Model_Member b);  
18  
19     public void update(Model_Member b);  
20  
21     public void delete(int id);  
22  
23     public java.util.List<Model_Member> getALL();  
24  
25     public java.util.List<Model_Member> getCariNama(String nama);  
26  
27 }
```

C. MENGISI SYNTAX DAO

1. Tambahkan *Implements Implement_Member{*. Kemudian *Add import* seperti berikut



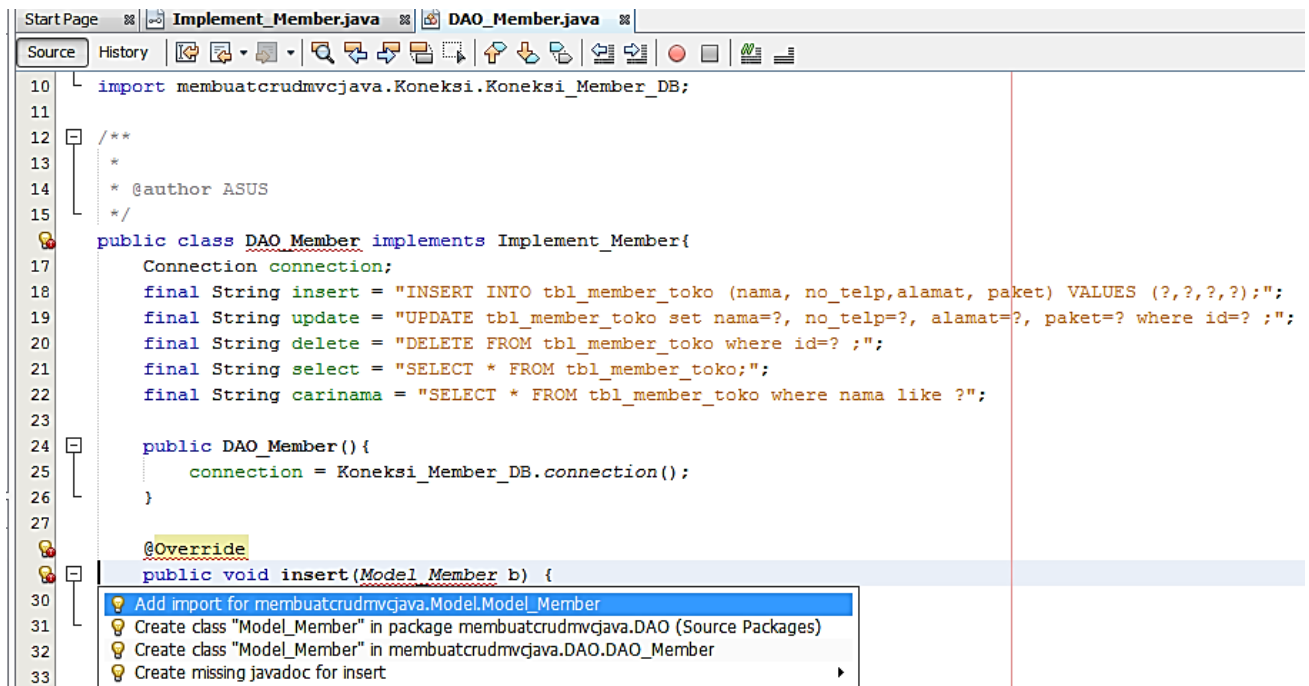
2. Masukkan syntax *Connection connection;* lalu *import Connection* seperti berikut

```
1  /*
2  * To change this license header, choose License Headers in Project Properties.
3  * To change this template file, choose Tools | Templates
4  * and open the template in the editor.
5  */
6  package membuatcrudmvcjava.DAO;
7
8  /**
9   *
10  * @author ASUS
11  */
12  public class DAO_Member implements Implement_Member{
13      Connection connection;
14  }
15
```

3. Lanjutkan syntax berikut, kemudian lakukan *import koneksi DB*

```
6  package membuatcrudmvcjava.DAO;
7
8  import java.sql.Connection;
9  import membuatcrudmvcjava.DAOImplement.Implement_Member;
10
11  /**
12   *
13   * @author ASUS
14   */
15  public class DAO_Member implements Implement_Member{
16      Connection connection;
17      final String insert = "INSERT INTO tbl_member_toko (nama, no_telp,alamat, paket) VALUES (?,?,?,?);";
18      final String update = "UPDATE tbl_member_toko set nama=?, no_telp=?, alamat=?, paket=? where id=? ;";
19      final String delete = "DELETE FROM tbl_member_toko where id=? ;";
20      final String select = "SELECT * FROM tbl_member_toko;";
21      final String carinama = "SELECT * FROM tbl_member_toko where nama like ?";
22
23      public DAO_Member() {
24          connection = Koneksi_Member_DB.connection();
25      }
26  }
27
```


4. Selanjutnya meng **@Override insert**, kemudian import di **insert(Model_Member b) {**, pilih **import ModelMember**

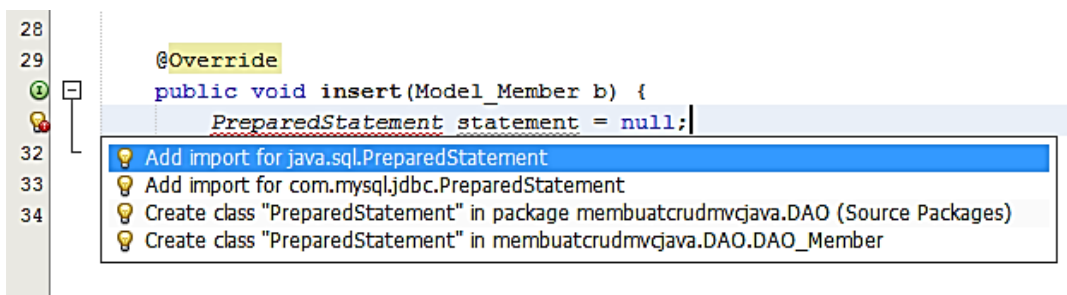


```
10 import membuatcrudmvcjava.Koneksi.Koneksi_Member_DB;
11
12 /**
13  *
14  * @author ASUS
15  */
16
17 public class DAO_Member implements Implement_Member{
18     Connection connection;
19     final String insert = "INSERT INTO tbl_member_toko (nama, no_telp,alamat, paket) VALUES (?,?=?,?);";
20     final String update = "UPDATE tbl_member_toko set nama=?, no_telp=?, alamat=?, paket=? where id=? ";
21     final String delete = "DELETE FROM tbl_member_toko where id=? ";
22     final String select = "SELECT * FROM tbl_member_toko;";
23     final String carinama = "SELECT * FROM tbl_member_toko where nama like ?";
24
25     public DAO_Member() {
26         connection = Koneksi_Member_DB.connection();
27     }
28
29     @Override
30     public void insert(Model_Member b) {
31
32     }
33 }
```

IDE Suggestion Box:

- Add import for membuatcrudmvcjava.Model.Model_Member
- Create class "Model_Member" in package membuatcrudmvcjava.DAO (Source Packages)
- Create class "Model_Member" in membuatcrudmvcjava.DAO.DAO_Member
- Create missing javadoc for insert

5. Pada syntax **PreparedStatement statement = null;** silahkan **Add import PreparedStatement** pada baris 1 seperti berikut.

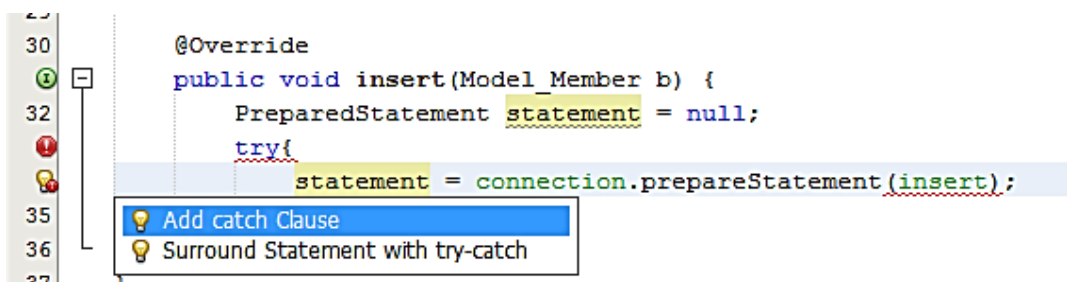


```
28
29 @Override
30 public void insert(Model_Member b) {
31     PreparedStatement statement = null;
32
33 }
34
```

IDE Suggestion Box:

- Add import for java.sql.PreparedStatement
- Add import for com.mysql.jdbc.PreparedStatement
- Create class "PreparedStatement" in package membuatcrudmvcjava.DAO (Source Packages)
- Create class "PreparedStatement" in membuatcrudmvcjava.DAO.DAO_Member

6. Selanjutnya masukkan **try{** kemudian **Add catch Clause**



```
30 @Override
31 public void insert(Model_Member b) {
32     PreparedStatement statement = null;
33     try{
34         statement = connection.prepareStatement(insert);
35     }
36 }
37
```

IDE Suggestion Box:

- Add catch Clause
- Surround Statement with try-catch

```

33      @Override
34      public void insert(Model_Member b) {
35          PreparedStatement statement = null;
36          try{
37              statement = connection.prepareStatement(insert);
38              statement.setString(1, b.getNama());
39              statement.setString(2, b.getNo_telp());
40              statement.setString(3, b.getAlamat());
41              statement.setString(4, b.getPaket());
42              statement.executeUpdate();
43              ResultSet rs = statement.getGeneratedKeys();
44              while(rs.next()){
45                  b.setId(rs.getInt(1));
46              }




```

7. Pada syntax ***ResultSet rs = statement.getGeneratedKeys();*** selanjutnya ***Add import for java.sql.ResultSet***

```

32
33      @Override
34      public void insert(Model_Member b) {
35          PreparedStatement statement = null;
36          try{
37              statement = connection.prepareStatement(insert);
38              statement.setString(1, b.getNama());
39              statement.setString(2, b.getNo_telp());
40              statement.setString(3, b.getAlamat());
41              statement.setString(4, b.getPaket());
42              statement.executeUpdate();
43              ResultSet rs = statement.getGeneratedKeys();
44
45              } catch (SQLException ex) {
46                  Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
47              }
48          }
49      }
50
51
52
53

```

 Add import for java.sql.ResultSet
 Create class "ResultSet" in package membuatcrudmvjava.DAO (Source Packages)
 Create class "ResultSet" in membuatcrudmvjava.DAO.DAO_Member

8. Rubahlah syntax ***catch (SQLException ex) { Logger.getLogger*** menjadi seperti berikut

```

49      } catch (SQLException ex) {
50          ex.printStackTrace();
51      } finally {
52          try{
53              statement.close();
54          } catch (SQLException ex) {
55              ex.printStackTrace();
56          }

```

9. Selanjutnya meng *@Override Update* seperti berikut

```
60      @Override
61      public void update(Model_Member b) {
62          PreparedStatement statement = null;
63          try{
64              statement = connection.prepareStatement(update);
65              statement.setString(1, b.getNama());
66              statement.setString(2, b.getNo_telp());
67              statement.setString(3, b.getAlamat());
68              statement.setString(4, b.getPaket());
69              statement.setInt(5, b.getId());
70              statement.executeUpdate();
71          } catch (SQLException ex) {
72              ex.printStackTrace();
73          } finally {
74              try{
75                  statement.close();
76              } catch (SQLException ex) {
77                  ex.printStackTrace();
78              }
79          }
80      }
```

10. Selanjutnya meng *@Override Delete* seperti berikut

```
82      @Override
83      public void delete(int id) {
84          PreparedStatement statement = null;
85          try{
86              statement = connection.prepareStatement(delete);
87
88              statement.setInt(1, id);
89              statement.executeUpdate();
90          } catch (SQLException ex) {
91              ex.printStackTrace();
92          } finally {
93              try{
94                  statement.close();
95              } catch (SQLException ex) {
96                  ex.printStackTrace();
97              }
98          }
99      }
```

11. Selanjutnya meng **@Override** *getALL* seperti berikut. Pada syntax *lb = new ArrayList<Model_Member>()*; Add import for *java.util.ArrayList*

```
100
101  @Override
102  public java.util.List<Model_Member> getALL() {
103      java.util.List<Model_Member> lb = null;
104      try{
105          lb = new ArrayList<Model_Member>();
106      } finally {
107          try{
108              statement.close();
109          } catch (SQLException ex) {
110              ex.printStackTrace();
111          }
112      }
113  }
114
115
116
117
118 }
```

Add import for java.util.ArrayList
Create class "ArrayList" in package membuatcrudmvcjava.DAO (Source Packages)
Create class "ArrayList" in membuatcrudmvcjava.DAO.DAO_Member

12. Selanjutnya pada syntax *Statement st = connection.createStatement()*; Add import for *java.sql.Statement*

Start Page | Implement_Member.java | DAO_Member.java

Source | History

```
94      try{
95          statement.close();
96      } catch (SQLException ex) {
97          ex.printStackTrace();
98      }
99  }
100
101
102  @Override
103  public java.util.List<Model_Member> getALL() {
104      java.util.List<Model_Member> lb = null;
105      try{
106          lb = new ArrayList<Model_Member>();
107          Statement st = connection.createStatement();
108      }
109  }
110
111
112
113 }
```

Add import for java.sql.Statement
Add import for com.mysql.jdbc.Statement
Add import for java.beans.Statement
Add import for jdk.nashorn.internal.ir.Statement
Create class "Statement" in package membuatcrudmvcjava.DAO (Source Packages)
Create class "Statement" in membuatcrudmvcjava.DAO.DAO_Member

13. Berikut ini adalah hasil *@Override getALL*

```
103      @Override
104      public java.util.List<Model_Member> getALL(){
105          java.util.List<Model_Member> lb = null;
106          try{
107              lb = new ArrayList<Model_Member>();
108              Statement st = connection.createStatement();
109              ResultSet rs = st.executeQuery(select);
110              while (rs.next()){
111                  Model_Member b = new Model_Member();
112                  b.setId(rs.getInt("id"));
113                  b.setNama(rs.getString("no_telp"));
114                  b.setNama(rs.getString("alamat"));
115                  b.setNama(rs.getString("paket"));
116                  lb.add(b);
117              }
118          } catch (SQLException ex) {
119              Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
120          }
121          return lb;
122      }
123  }
```

14. Selanjutnya meng *@Override getCariNama* seperti berikut

```
124      @Override
125      public java.util.List<Model_Member> getCariNama(String nama){
126          java.util.List<Model_Member> lb = null;
127          try{
128              lb = new ArrayList<Model_Member>();
129              PreparedStatement st = connection.prepareStatement(carinama);
130              st.setString(1, "%" + nama + "%");
131              ResultSet rs = st.executeQuery();
132              while (rs.next()){
133                  Model_Member b = new Model_Member();
134                  b.setId(rs.getInt("id"));
135                  b.setNama(rs.getString("no_telp"));
136                  b.setNama(rs.getString("alamat"));
137                  b.setNama(rs.getString("paket"));
138                  lb.add(b);
139              }
140          } catch (SQLException ex) {
141              Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
142          }
143          return lb;
144      }
145  }
```

15. Jika syntax program diatas sudah dikerjakan semua, maka hasil akhir dari syntax program

DAO sebagai berikut

```

6      package membuatcrudmvcjava.DAO;
7
8      import java.sql.Connection;
9      import java.sql.PreparedStatement;
10     import java.sql.ResultSet;
11     import java.sql.SQLException;
12     import java.sql.Statement;
13     import java.util.ArrayList;
14     import java.util.logging.Level;
15     import java.util.logging.Logger;
16     import membuatcrudmvcjava.DAOImplement.Implement_Member;
17     import membuatcrudmvcjava.Koneksi.Koneksi_Member_DB;
18     import membuatcrudmvcjava.Model.Model_Member;
19
20     /**
21      *
22      * @author SAFIRA
23      */
24     public class DAO_Member implements Implement_Member{
25         Connection connection;
26         final String insert = "INSERT INTO tbl_member_toko (nama, no_telp,alamat, paket) VALUES (?, ?, ?, ?)";
27         final String update = "UPDATE tbl_member_toko set nama=?, no_telp=?, alamat=?, paket=? where id=? ";
28         final String delete = "DELETE FROM tbl_member_toko where id=? ";
29         final String select = "SELECT * FROM tbl_member_toko";
30         final String carinama = "SELECT * FROM tbl_member_toko where nama like ?";
31
32     public DAO_Member() {
33         connection = Koneksi_Member_DB.connection();
34     }
35
36     @Override
37     public void insert(Model_Member b) {
38         PreparedStatement statement = null;
39         try{
40             statement = connection.prepareStatement(insert);
41             statement.setString(1, b.getNama());
42             statement.setString(2, b.getNo_telp());
43             statement.setString(3, b.getAlamat());
44             statement.setString(4, b.getPaket());
45             statement.executeUpdate();
46             ResultSet rs = statement.getGeneratedKeys();
47             while(rs.next()){
48                 b.setId(rs.getInt(1));
49             }
50         } catch (SQLException ex) {
51             ex.printStackTrace();
52         } finally {
53             try{
54                 statement.close();
55             } catch (SQLException ex) {
56                 ex.printStackTrace();
57             }
58         }
59     }
60 }

```



```

62      @Override
63      public void update(Model_Member b) {
64          PreparedStatement statement = null;
65          try{
66              statement = connection.prepareStatement(update);
67              statement.setString(1, b.getNama());
68              statement.setString(2, b.getNo_telp());
69              statement.setString(3, b.getAlamat());
70              statement.setString(4, b.getPaket());
71              statement.setInt(5, b.getId());
72              statement.executeUpdate();
73          } catch (SQLException ex) {
74              ex.printStackTrace();
75          } finally {
76              try{
77                  statement.close();
78              }catch(SQLException ex){
79                  ex.printStackTrace();
80              }
81          }
82      }

84      @Override
85      public void delete(int id) {
86          PreparedStatement statement = null;
87          try{
88              statement = connection.prepareStatement(delete);
89
90              statement.setInt(1, id);
91              statement.executeUpdate();
92          } catch (SQLException ex) {
93              ex.printStackTrace();
94          } finally {
95              try{
96                  statement.close();
97              }catch(SQLException ex){
98                  ex.printStackTrace();
99              }
100          }
101      }

103      @Override
104      public java.util.List<Model_Member> getAll(){
105          java.util.List<Model_Member> lb = null;
106          try{
107              lb = new ArrayList<Model_Member>();
108              Statement st = connection.createStatement();
109              ResultSet rs = st.executeQuery(select);
110              while (rs.next()){
111                  Model_Member b = new Model_Member();
112                  b.setId(rs.getInt("id"));
113                  b.setNama(rs.getString("no_telp"));
114                  b.setNama(rs.getString("alamat"));
115                  b.setNama(rs.getString("paket"));
116                  lb.add(b);
117              }

```



```

118         } catch (SQLException ex) {
119             Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
120         }
121         return lb;
122     }
123
124     @Override
125     public java.util.List<Model_Member> getCariNama(String nama){
126         java.util.List<Model_Member> lb = null;
127         try{
128             lb = new ArrayList<Model_Member>();
129             PreparedStatement st = connection.prepareStatement(carinama);
130             st.setString(1, "%" + nama + "%");
131             ResultSet rs = st.executeQuery();
132             while (rs.next()) {
133                 Model_Member b = new Model_Member();
134                 b.setId(rs.getInt("id"));
135                 b.setNama(rs.getString("no_telp"));
136                 b.setNama(rs.getString("alamat"));
137                 b.setNama(rs.getString("paket"));
138                 lb.add(b);
139             }
140         } catch (SQLException ex) {
141             Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
142         }
143         return lb;
144     }
145 }
146

```

Catatan :

1. Silahkan mengerjakan semua latihan yang terdapat diatas, bisa didiskusikan dengan teman kalian. Tetapi untuk pengerjaan tugasnya tetap individu.
2. Silahkan segera diselesaikan, batas waktu 3 hari. Untuk selanjutnya akan saya berikan modul lagi yang mana pada modul berikutnya adalah lanjutan dari project yang sudah kalian buat seperti latihan di atas. Jika project diatas belum terselesaikan, maka tidak bisa melanjutkan mengerjakan pada modul selanjutnya.

SELAMAT MENGERJAKAN ^_^