MODUL 13.3
“KONEKSI DATABASE, DAO, DAO IMPLEMENT (MVC)”

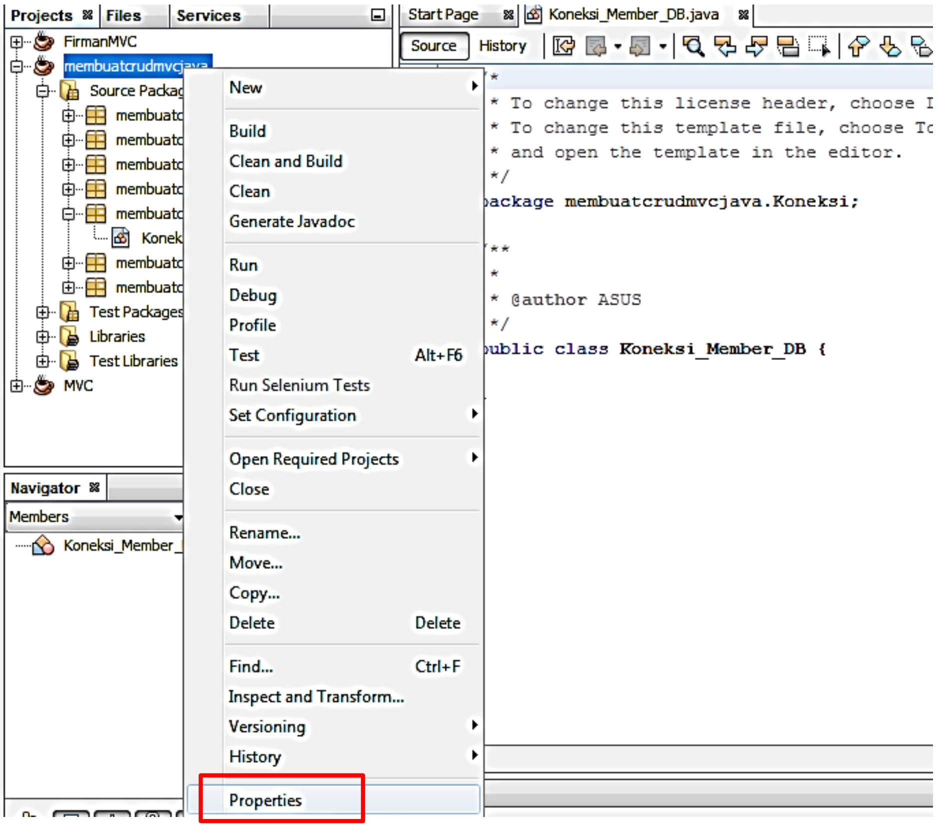
Rekayasa Perangkat Lunak
(Software Engineering)
SMKN 2 TRENGGALEK

Oleh

SAFIRA MAYA SHOVIE, S.Pd

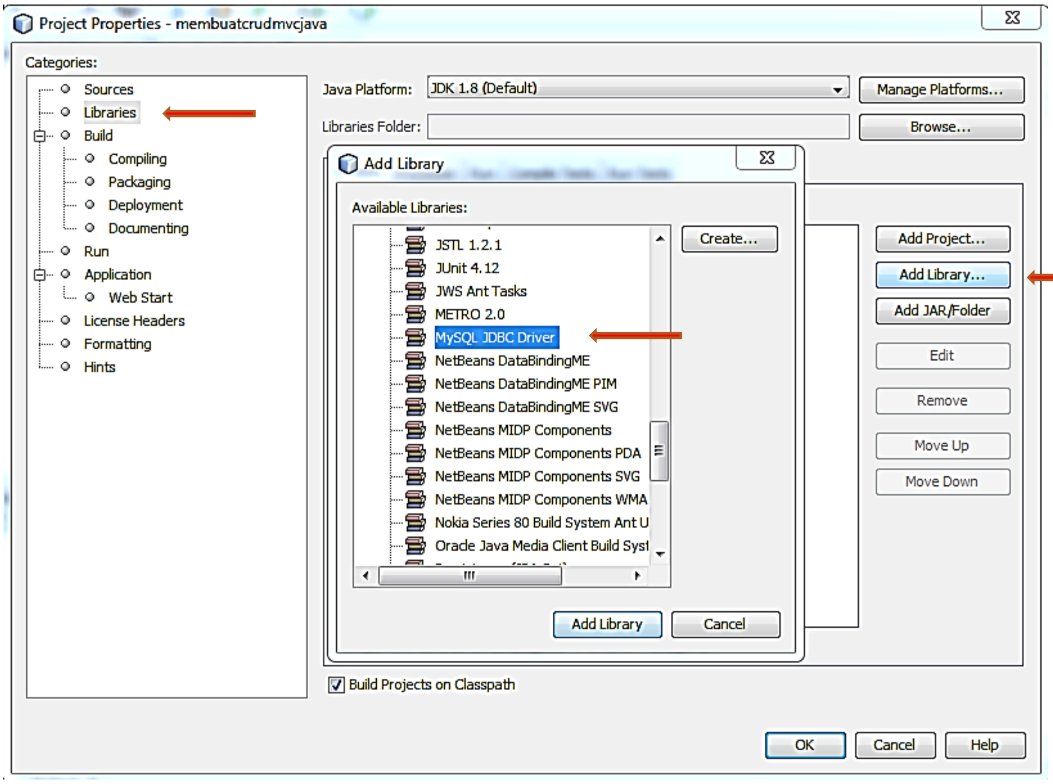
A. KONEKSI DATABASE

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



```
 *
 * To change this license header, choose I
 * To change this template file, choose Tc
 * and open the template in the editor.
 */
package membuatcrudmvcjava.Koneksi;

/**
 *
 * @author ASUS
 */
public class Koneksi_Member_DB {
```

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



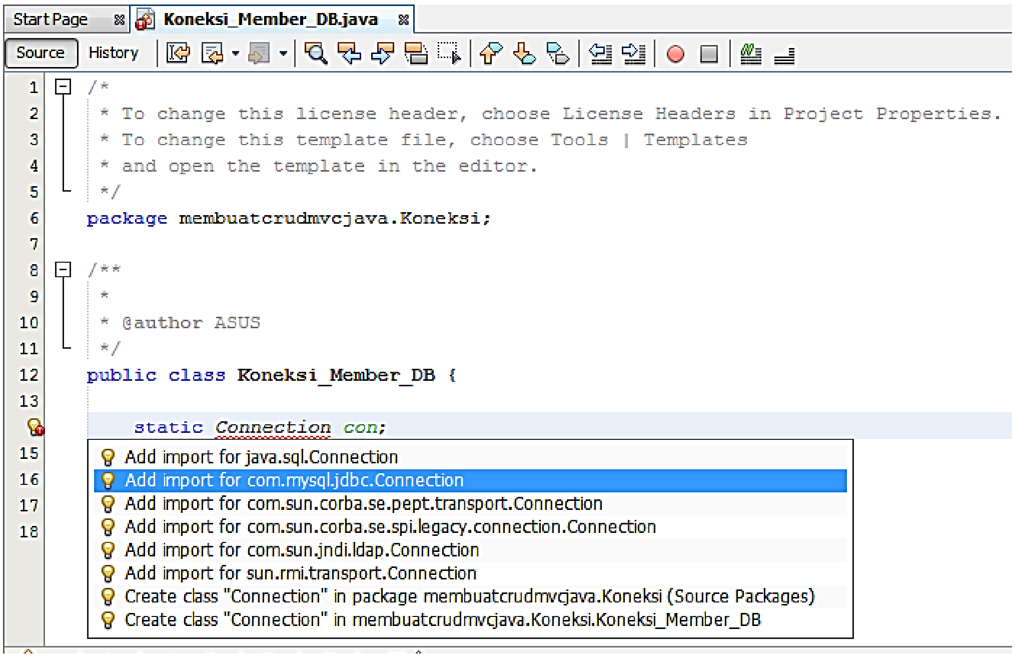
Java Platform: JDK 1.8 (Default)

Libraries Folder:

☑ Build Projects on Classpath

3. Masukkan Syntax pada Source Package membuatcrudmvcjava.Koneksi

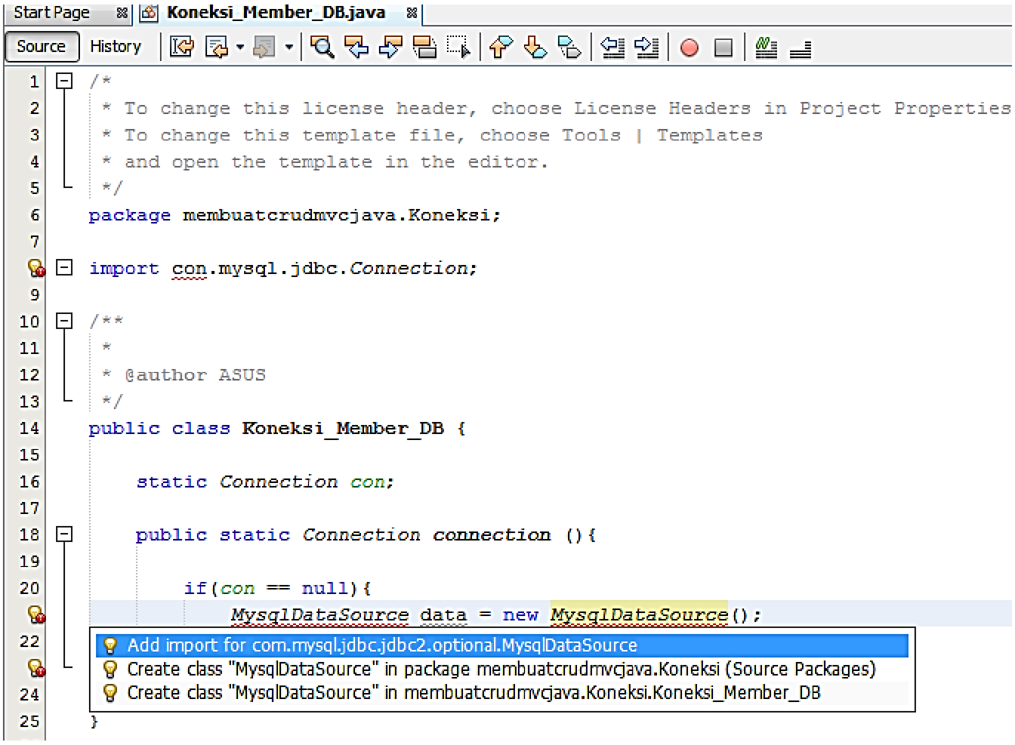
   Pada bagian yang terdapat tanda pentung merah, silahkan di klik dan dibenarkan seperti berikut



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package membuatcrudmvcjava.Koneksi;

/**
 *
 * @author ASUS
 */
public class Koneksi_Member_DB {

    static Connection con;
```

- Add import for java.sql.Connection
- Add import for com.mysql.jdbc.Connection
- Add import for com.sun.corba.se.pept.transport.Connection
- Add import for com.sun.corba.se.spi.legacy.connection.Connection
- Add import for com.sun.jndi.ldap.Connection
- Add import for sun.rmi.transport.Connection
- Create class "Connection" in package membuatcrudmvcjava.Koneksi (Source Packages)
- 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

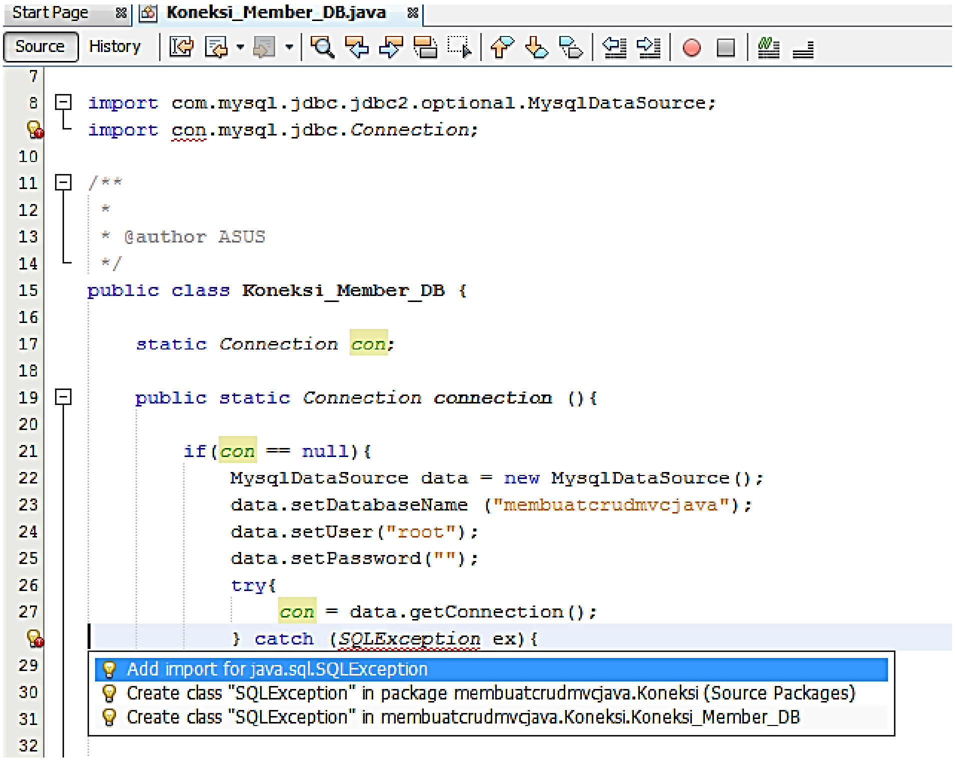


```
/*
 * To change this license header, choose License Headers in Project Properties
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package membuatcrudmvcjava.Koneksi;

import con.mysql.jdbc.Connection;

/**
 *
 * @author ASUS
 */
public class Koneksi_Member_DB {

    static Connection con;

    public static Connection connection (){

        if(con == null){
            MysqlDataSource data = new MysqlDataSource();
```

- Add import for com.mysql.jdbc.jdbc2.optional.MysqlDataSource
- Create class "MysqlDataSource" in package membuatcrudmvcjava.Koneksi (Source Packages)
- Create class "MysqlDataSource" in membuatcrudmvcjava.Koneksi.Koneksi_Member_DB

```
}
```

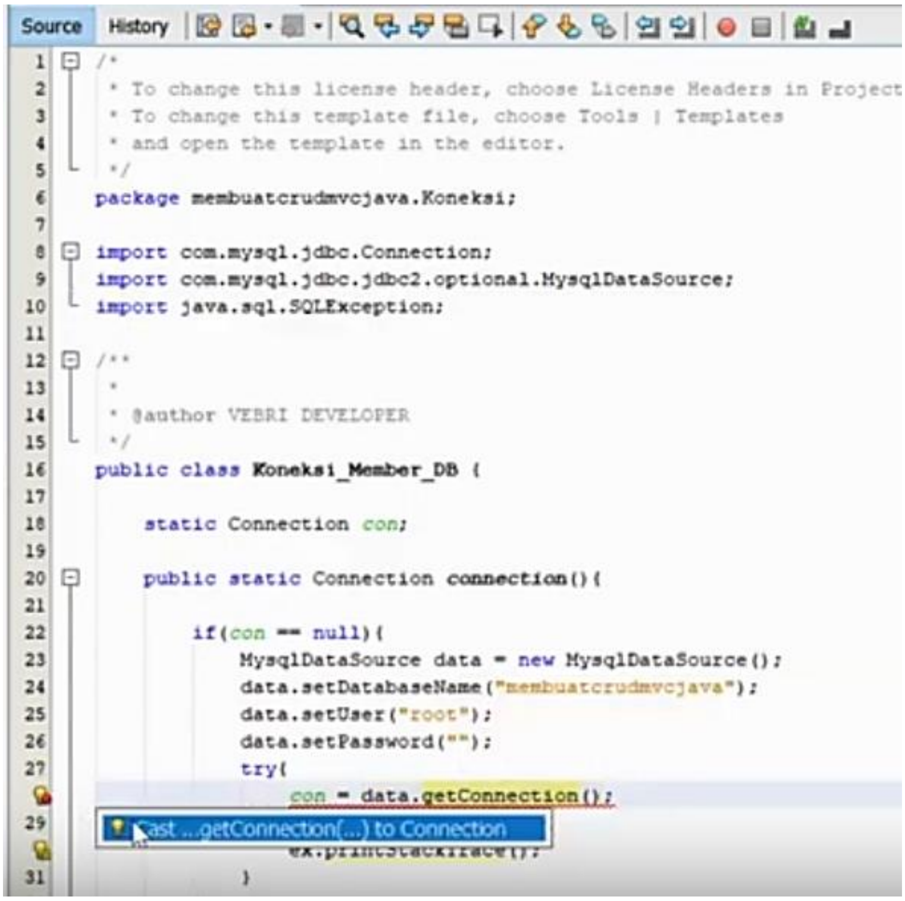
Pada bagian yang terdapat tanda pentung merah/ ***} catch (SQLException ex){***, silahkan di klik dan dibenarkan seperti berikut



```
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
import con.mysql.jdbc.Connection;

/**
 *
 * @author ASUS
 */
public class Koneksi_Member_DB {

    static Connection con;

    public static Connection connection (){

        if(con == null){
            MysqlDataSource data = new MysqlDataSource();
            data.setDatabaseName ("membuatcrudmvcjava");
            data.setUser("root");
            data.setPassword("");
            try{
                con = data.getConnection();
            } catch (SQLException ex){
```

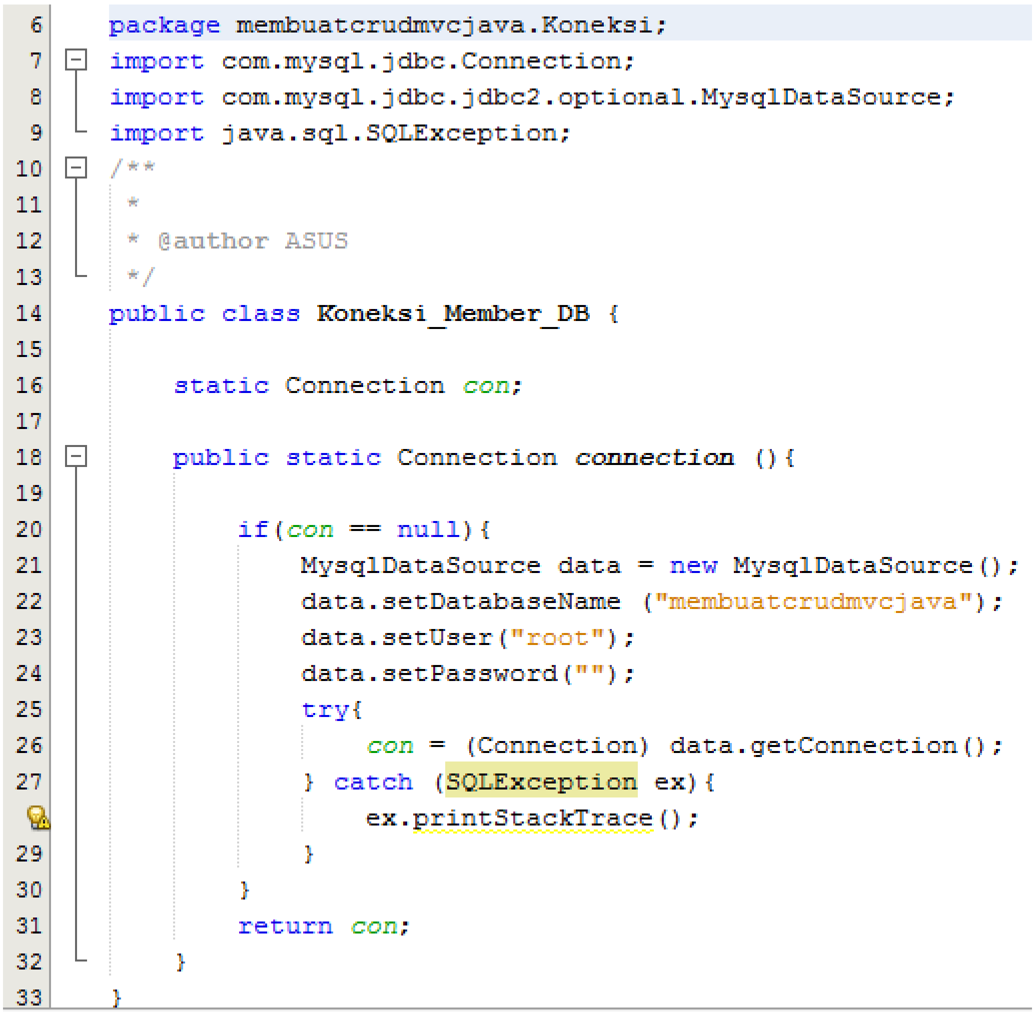
```
Add import for java.sql.SQLException
Create class "SQLException" in package membuatcrudmvcjava.Koneksi (Source Packages)
Create class "SQLException" in membuatcrudmvcjava.Koneksi.Koneksi_Member_DB
```

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



```
/*
 * To change this license header, choose License Headers in Project
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package membuatcrudmvcjava.Koneksi;

import com.mysql.jdbc.Connection;
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
import java.sql.SQLException;

/**
 *
 * @author VEBRI DEVELOPER
 */
public class Koneksi_Member_DB {

    static Connection con;

    public static Connection connection(){

        if(con == null){
            MysqlDataSource data = new MysqlDataSource();
            data.setDatabaseName("membuatcrudmvcjava");
            data.setUser("root");
            data.setPassword("");
            try{
                con = data.getConnection();
Cast ...getConnection(...) to Connection
                ex.printStackTrace();
            }
```

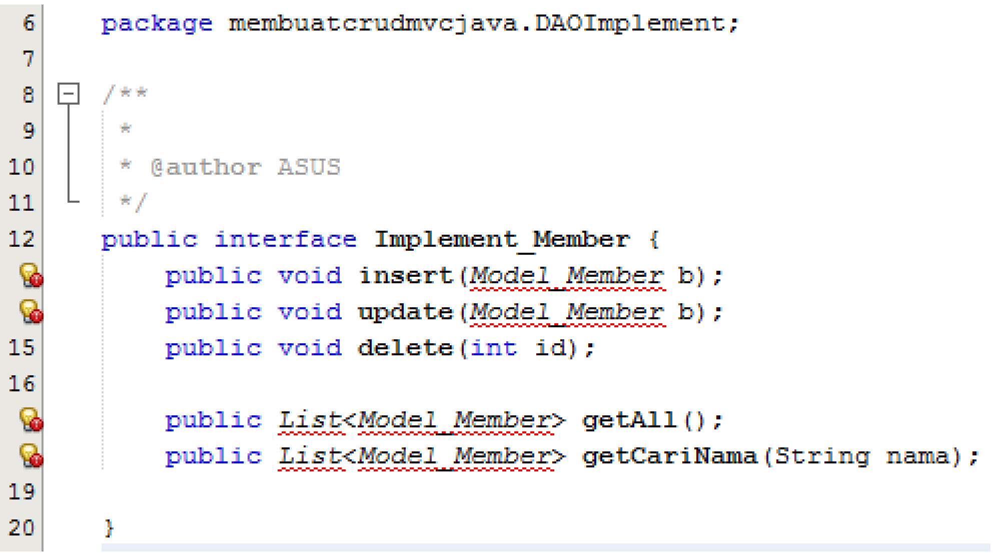
4. Lanjutkan selesaikan penulisan syntax seperti berikut. Kemudian run untuk mengetahui program berhasil atau belum.



```java
package membuatcrudmvcjava.Koneksi;
import com.mysql.jdbc.Connection;
import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
import java.sql.SQLException;
/**
 *
 * @author ASUS
 */
public class Koneksi_Member_DB {

    static Connection con;

    public static Connection connection (){

        if(con == null){
            MysqlDataSource data = new MysqlDataSource();
            data.setDatabaseName ("membuatcrudmvcjava");
            data.setUser("root");
            data.setPassword("");
            try{
                con = (Connection) data.getConnection();
            } catch (SQLException ex){
                ex.printStackTrace();
            }
        }
        return con;
    }
}
```

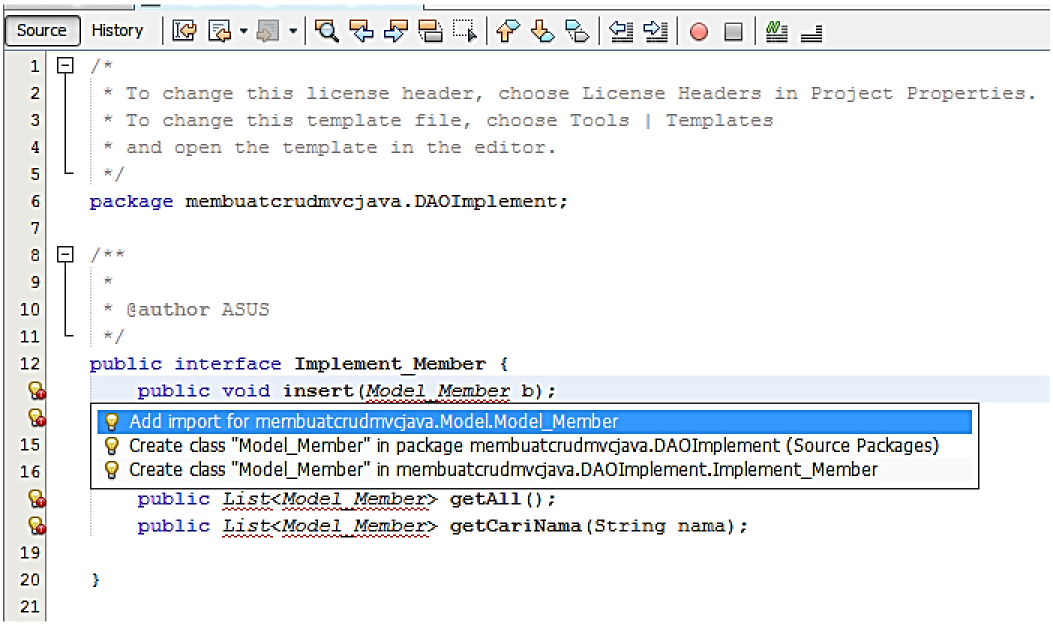
B. MENGISI SYNTAX DAO IMPLEMENT

1. Tuliskan syntax berikut



```java
package membuatcrudmvcjava.DAOImplement;

/**
 *
 * @author ASUS
 */
public interface Implement_Member {
    public void insert(Model_Member b);
    public void update(Model_Member b);
    public void delete(int id);

    public List<Model_Member> getAll();
    public List<Model_Member> getCariNama(String nama);

}
```

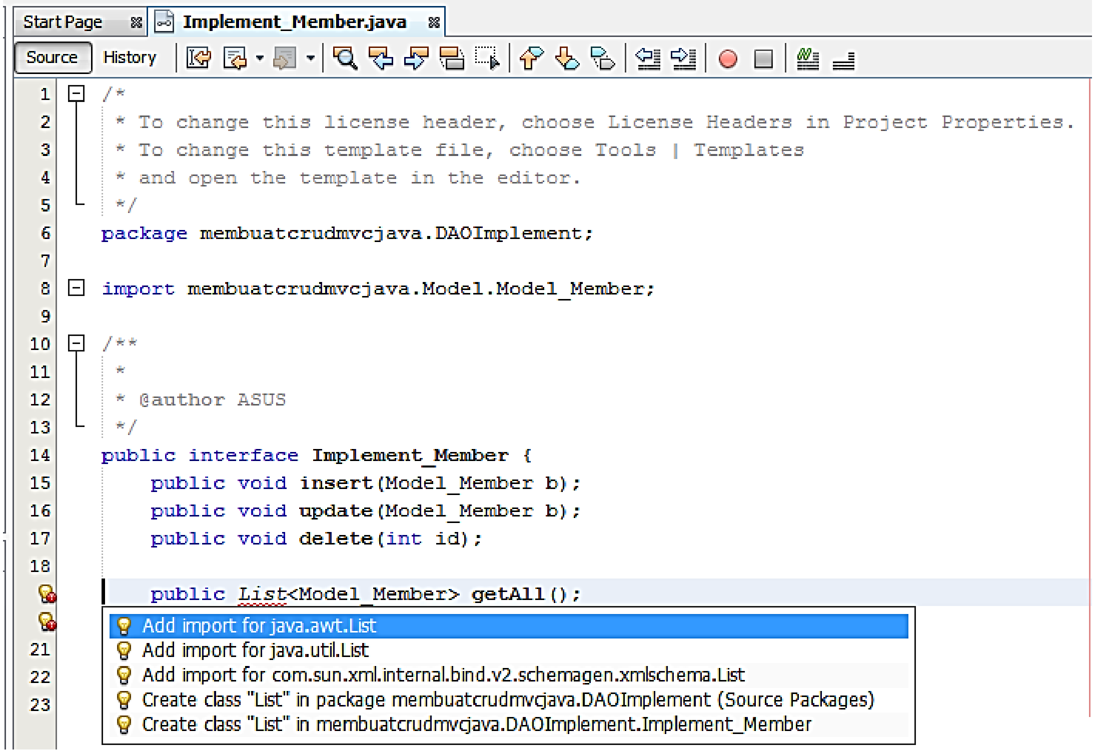
Karena ada eroor, silahkan lakukan pembenaran seperti berikut. Klik pada baris ***public void insert(Model_Member b);*** ,jika terdapat pilihan, pilihlah baris 1 ***Add import***.



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package membuatcrudmvcjava.DAOImplement;

/**
 *
 * @author ASUS
 */
public interface Implement_Member {
    public void insert(Model_Member b);
    public List<Model_Member> getAll();
    public List<Model_Member> getCariNama(String nama);

}
```

```
Add import for membuatcrudmvcjava.Model.Model_Member
Create class "Model_Member" in package membuatcrudmvcjava.DAOImplement (Source Packages)
Create class "Model_Member" in membuatcrudmvcjava.DAOImplement.Implement_Member
```

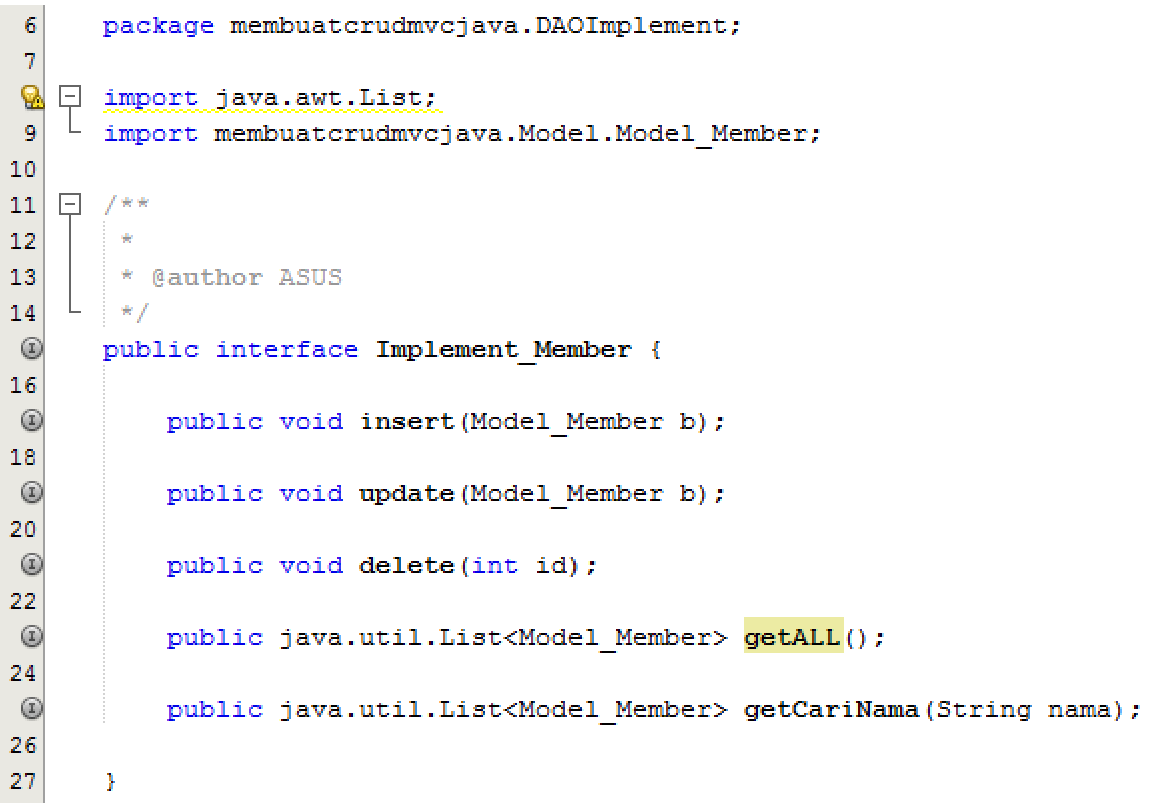
Selanjutnya pada bagian public ***List<Model_Member> getAll();*** , silahkan diganti ***Add import for java.awt.List***



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package membuatcrudmvcjava.DAOImplement;

import membuatcrudmvcjava.Model.Model_Member;

/**
 *
 * @author ASUS
 */
public interface Implement_Member {
    public void insert(Model_Member b);
    public void update(Model_Member b);
    public void delete(int id);

    public List<Model_Member> getAll();
```

```
Add import for java.awt.List
Add import for java.util.List
Add import for com.sun.xml.internal.bind.v2.schemagen.xmlschema.List
Create class "List" in package membuatcrudmvcjava.DAOImplement (Source Packages)
Create class "List" in membuatcrudmvcjava.DAOImplement.Implement_Member
```

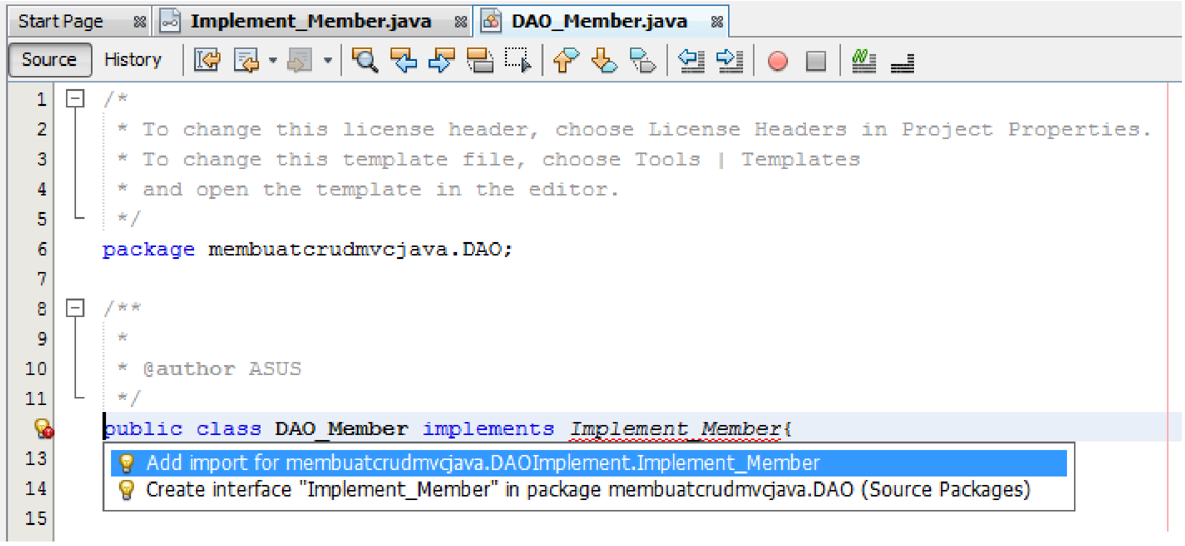
2. Samakan syntaxnya seperti berikut



```
package membuatcrudmvcjava.DAOImplement;

import java.awt.List;
import membuatcrudmvcjava.Model.Model_Member;

/**
 *
 * @author ASUS
 */
public interface Implement_Member {

    public void insert(Model_Member b);

    public void update(Model_Member b);

    public void delete(int id);

    public java.util.List<Model_Member> getALL();

    public java.util.List<Model_Member> getCariNama(String nama);

}
```

C. MENGISI SYNTAX DAO

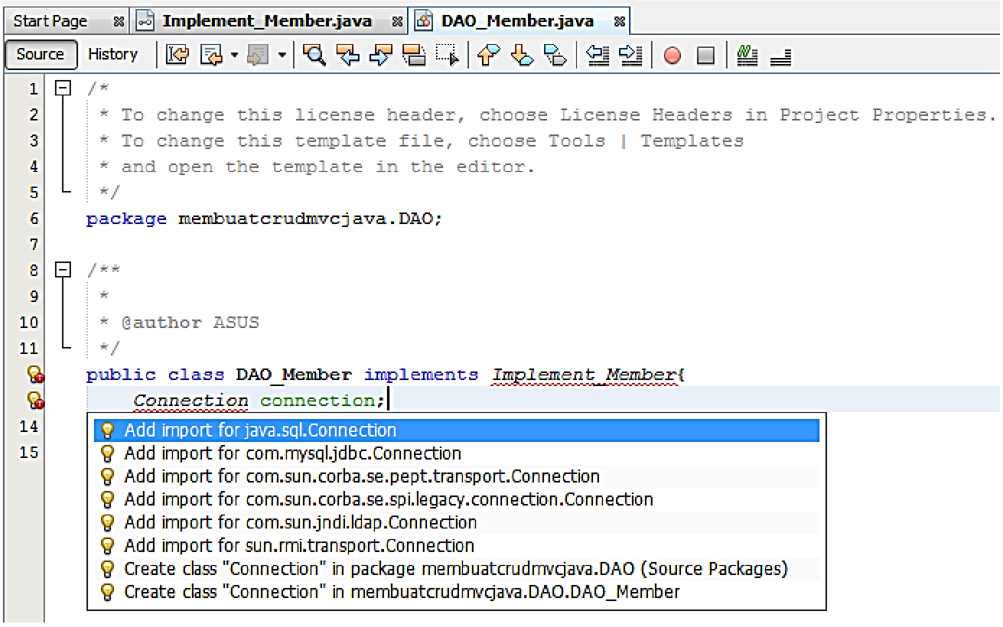
1. Tambahkan ***Implements Implement_Member{***. Kemudian ***Add import*** seperti berikut



```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package membuatcrudmvcjava.DAO;

/**
 *
 * @author ASUS
 */
public class DAO_Member implements Implement_Member{
```

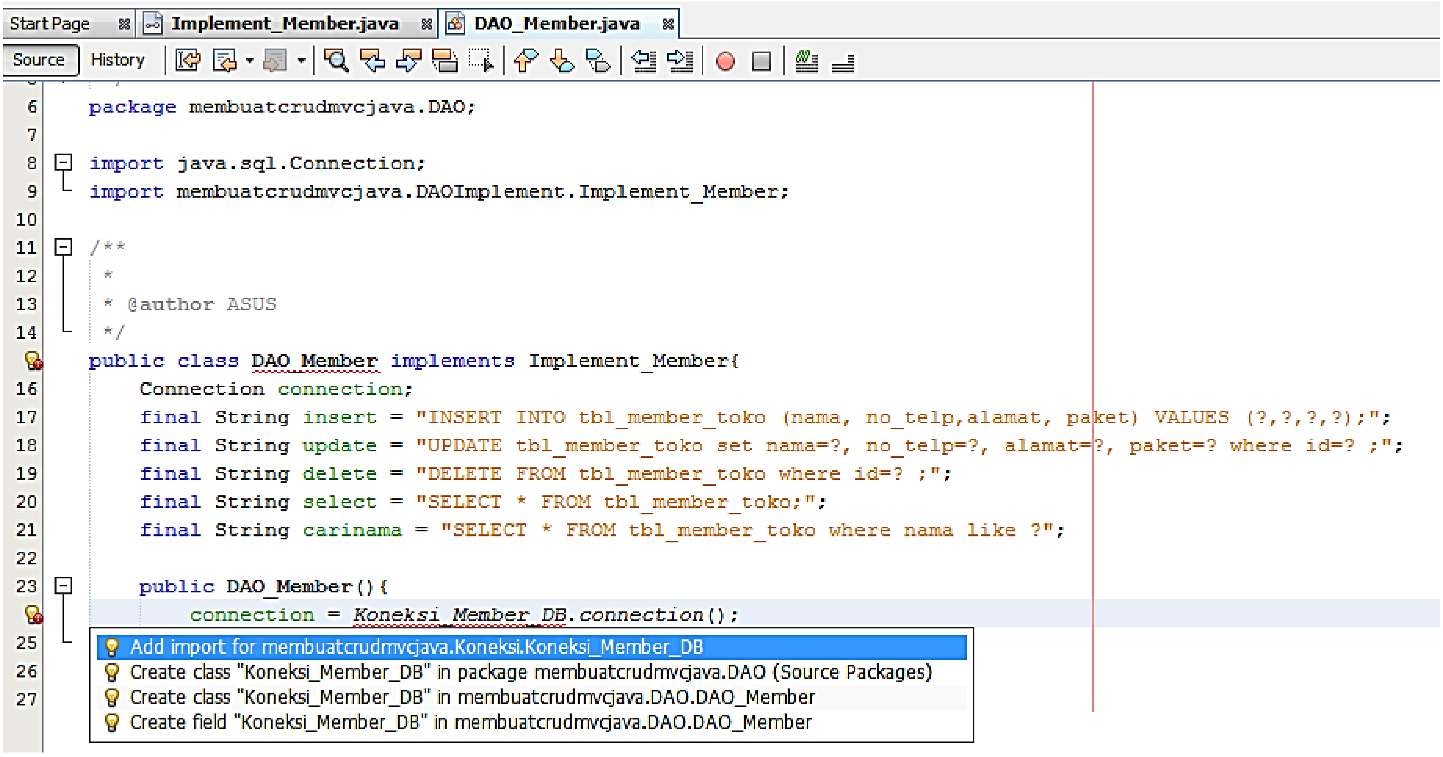
Add import for membuatcrudmvcjava.DAOImplement.Implement_Member
Create interface "Implement_Member" in package membuatcrudmvcjava.DAO (Source Packages)

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


Start Page
Implement_Member.java
DAO_Member.java
Source
History
Add import for java.sql.Connection
Add import for com.mysql.jdbc.Connection
Add import for com.sun.corba.se.pept.transport.Connection
Add import for com.sun.corba.se.spi.legacy.connection.Connection
Add import for com.sun.jndi.ldap.Connection
Add import for sun.rmi.transport.Connection
Create class "Connection" in package membuatcrudmvcjava.DAO (Source Packages)
Create class "Connection" in membuatcrudmvcjava.DAO.DAO_Member

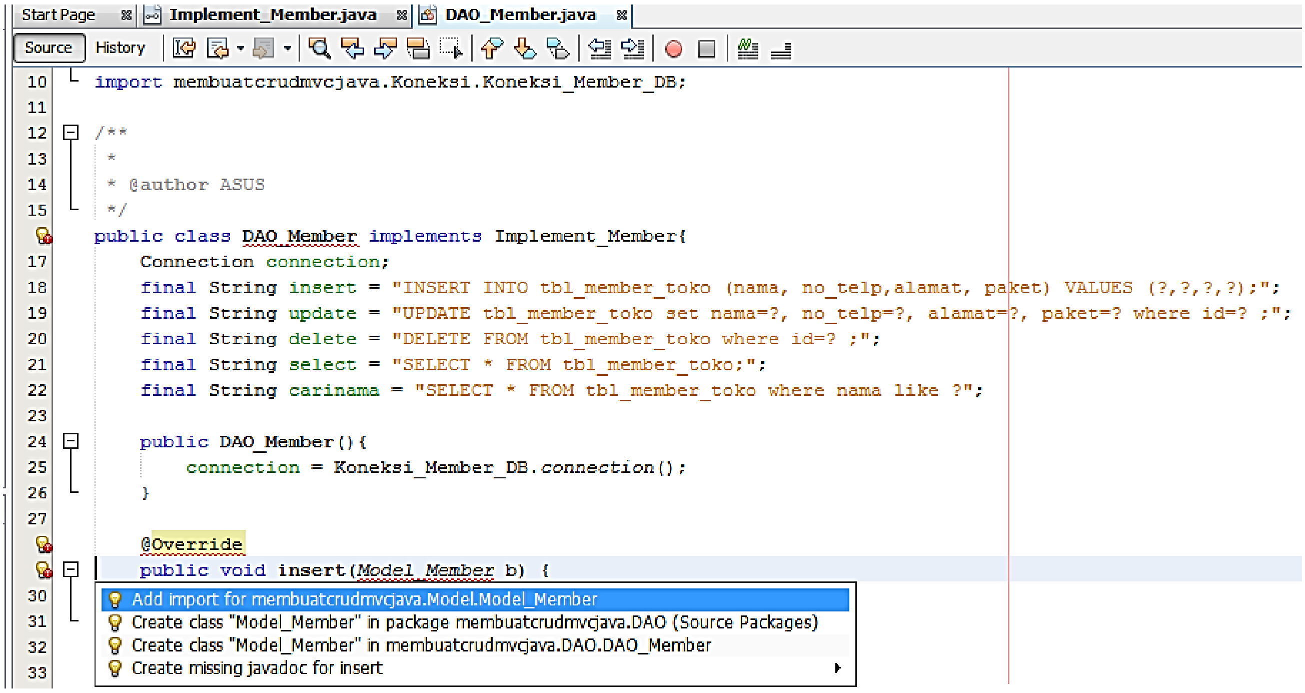
```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package membuatcrudmvcjava.DAO;

/**
 *
 * @author ASUS
 */
public class DAO_Member implements Implement_Member{
    Connection connection;
```

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


Start Page
Implement_Member.java
DAO_Member.java
Source
History
Add import for membuatcrudmvcjava.Koneksi.Koneksi_Member_DB
Create class "Koneksi_Member_DB" in package membuatcrudmvcjava.DAO (Source Packages)
Create class "Koneksi_Member_DB" in membuatcrudmvcjava.DAO.DAO_Member
Create field "Koneksi_Member_DB" in membuatcrudmvcjava.DAO.DAO_Member

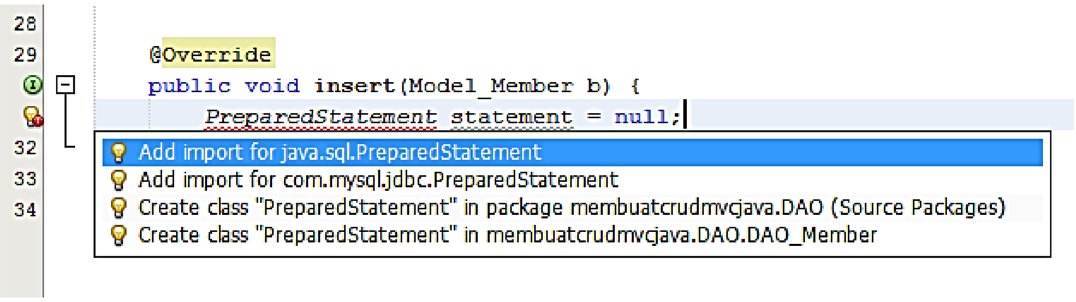
```java
package membuatcrudmvcjava.DAO;

import java.sql.Connection;
import membuatcrudmvcjava.DAOImplement.Implement_Member;

/**
 *
 * @author ASUS
 */
public class DAO_Member implements Implement_Member{
    Connection connection;
    final String insert = "INSERT INTO tbl_member_toko (nama, no_telp,alamat, paket) VALUES (?,?,?,?);";
    final String update = "UPDATE tbl_member_toko set nama=?, no_telp=?, alamat=?, paket=? where id=? ;";
    final String delete = "DELETE FROM tbl_member_toko where id=? ;";
    final String select = "SELECT * FROM tbl_member_toko;";
    final String carinama = "SELECT * FROM tbl_member_toko where nama like ?";

    public DAO_Member(){
        connection = Koneksi_Member_DB.connection();
```

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


Start Page
Implement_Member.java
DAO_Member.java
Source
History
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

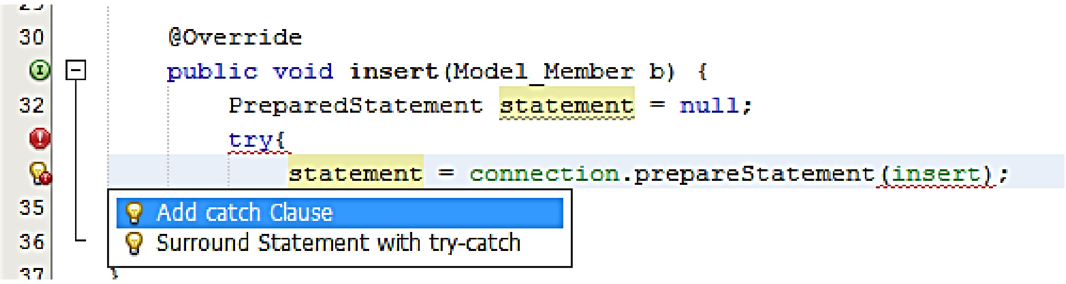
```
import membuatcrudmvcjava.Koneksi.Koneksi_Member_DB;

/**
 *
 * @author ASUS
 */
public class DAO_Member implements Implement_Member{
    Connection connection;
    final String insert = "INSERT INTO tbl_member_toko (nama, no_telp,alamat, paket) VALUES (?,?,?,?);";
    final String update = "UPDATE tbl_member_toko set nama=?, no_telp=?, alamat=?, paket=? where id=? ;";
    final String delete = "DELETE FROM tbl_member_toko where id=? ;";
    final String select = "SELECT * FROM tbl_member_toko;";
    final String carinama = "SELECT * FROM tbl_member_toko where nama like ?";

    public DAO_Member(){
        connection = Koneksi_Member_DB.connection();
    }

    @Override
    public void insert(Model_Member b) {
```

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

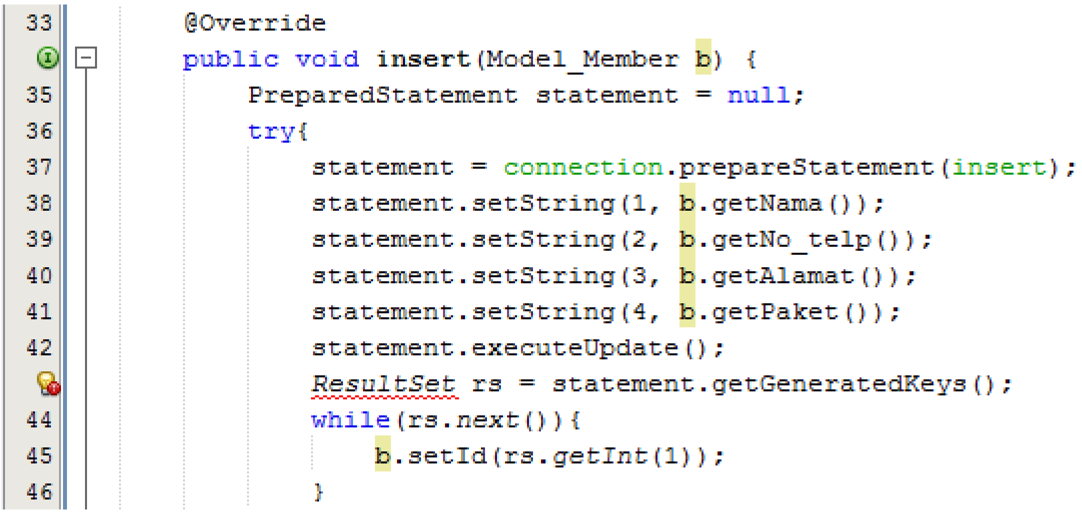

Add import for java.sql.PreparedStatement
Add import for com.mysql.jdbc.PreparedStatement
Create class "PreparedStatement" in package membuatcrudmvcjava.DAO (Source Packages)
Create class "PreparedStatement" in membuatcrudmvcjava.DAO.DAO_Member

```
    @Override
    public void insert(Model_Member b) {
        PreparedStatement statement = null;
```

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

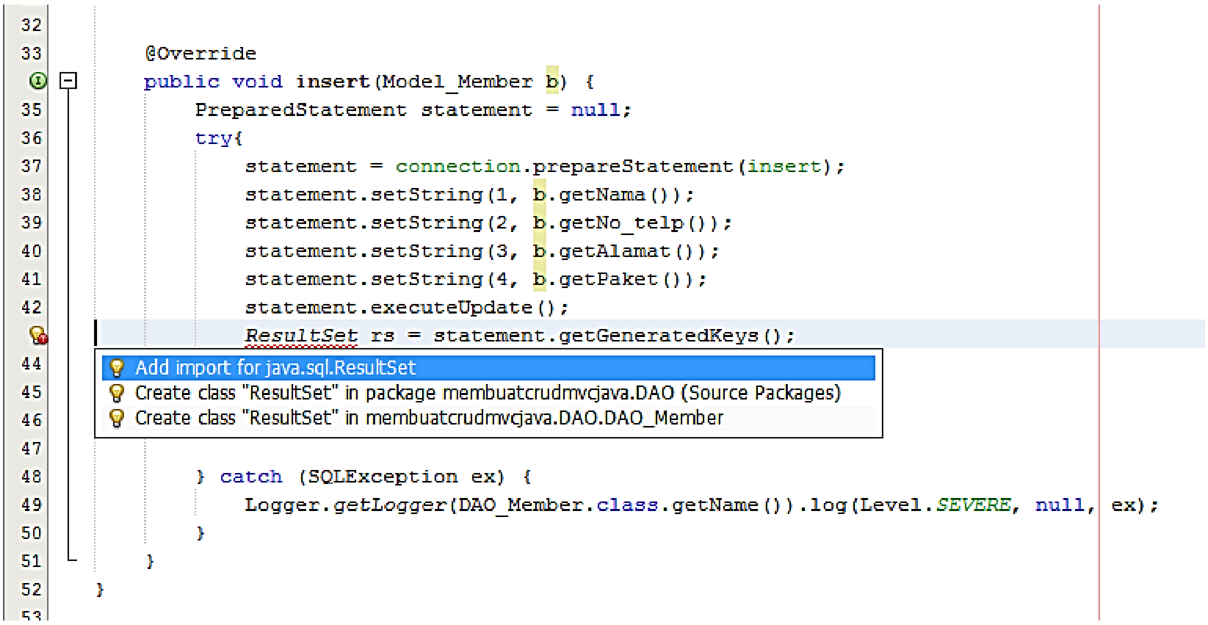


```
    @Override
    public void insert(Model_Member b) {
        PreparedStatement statement = null;
        try{
            statement = connection.prepareStatement(insert);
```



```
    @Override
    public void insert(Model_Member b) {
        PreparedStatement statement = null;
        try{
            statement = connection.prepareStatement(insert);
            statement.setString(1, b.getNama());
            statement.setString(2, b.getNo_telp());
            statement.setString(3, b.getAlamat());
            statement.setString(4, b.getPaket());
            statement.executeUpdate();
            ResultSet rs = statement.getGeneratedKeys();
            while(rs.next()){
                b.setId(rs.getInt(1));
            }
```

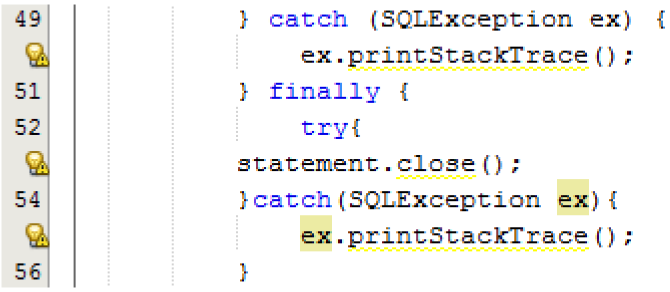
7. Pada syntax ***ResultSet rs = statement.getGeneratedKeys();*** selanjutnya ***Add import for java.sql.ResultSet***



```
    @Override
    public void insert(Model_Member b) {
        PreparedStatement statement = null;
        try{
            statement = connection.prepareStatement(insert);
            statement.setString(1, b.getNama());
            statement.setString(2, b.getNo_telp());
            statement.setString(3, b.getAlamat());
            statement.setString(4, b.getPaket());
            statement.executeUpdate();
            ResultSet rs = statement.getGeneratedKeys();

        } catch (SQLException ex) {
            Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
}
```

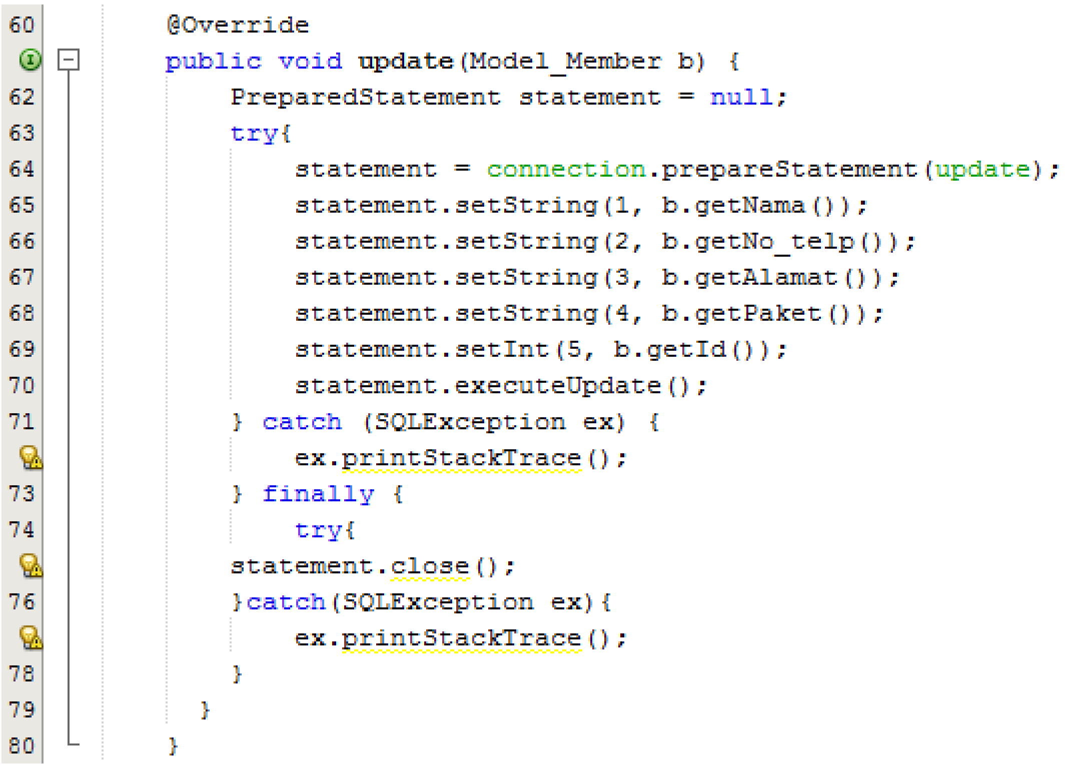
Add import for java.sql.ResultSet
Create class "ResultSet" in package membuatcrudmvcjava.DAO (Source Packages)
Create class "ResultSet" in membuatcrudmvcjava.DAO.DAO_Member

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



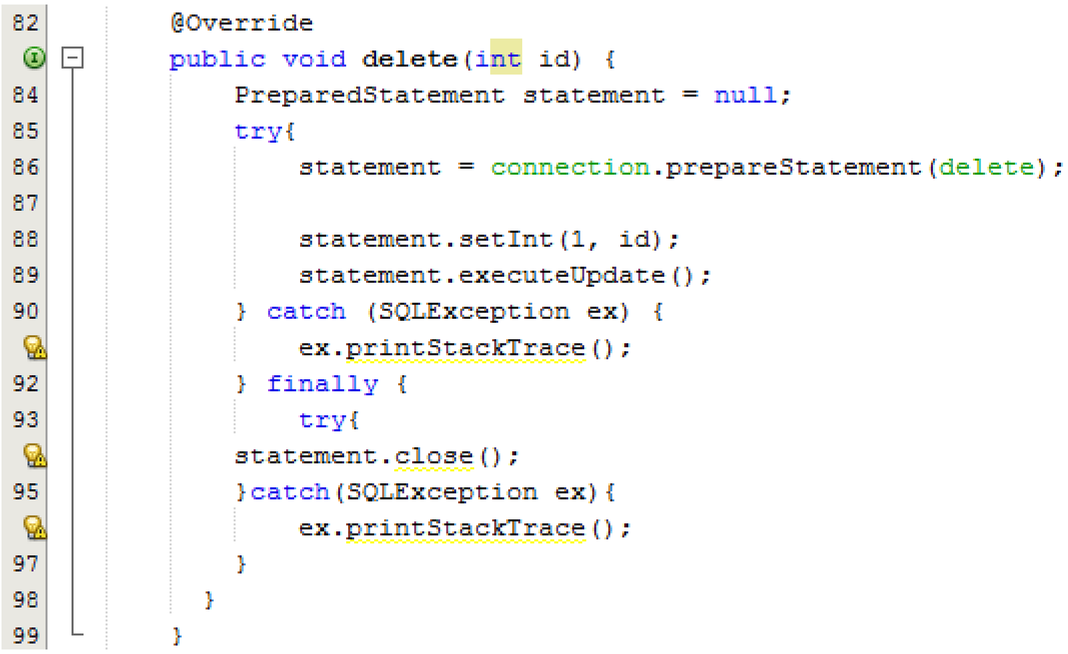
```
        } catch (SQLException ex) {
            ex.printStackTrace();
        } finally {
            try{
        statement.close();
        }catch(SQLException ex){
            ex.printStackTrace();
        }
```

9. Selanjutnya meng ***@Override Update*** seperti berikut



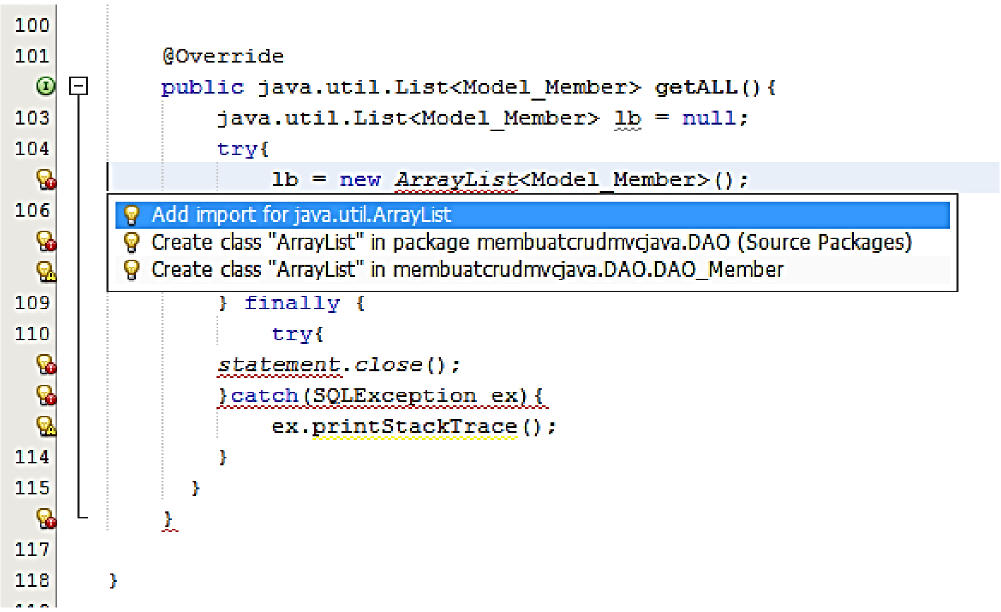
```
@Override
public void update(Model_Member b) {
    PreparedStatement statement = null;
    try{
        statement = connection.prepareStatement(update);
        statement.setString(1, b.getNama());
        statement.setString(2, b.getNo_telp());
        statement.setString(3, b.getAlamat());
        statement.setString(4, b.getPaket());
        statement.setInt(5, b.getId());
        statement.executeUpdate();
    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try{
    statement.close();
    }catch(SQLException ex){
        ex.printStackTrace();
    }
  }
}
```

10. Selanjutnya meng ***@Override Delete*** seperti berikut



```
@Override
public void delete(int id) {
    PreparedStatement statement = null;
    try{
        statement = connection.prepareStatement(delete);

        statement.setInt(1, id);
        statement.executeUpdate();
    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try{
    statement.close();
    }catch(SQLException ex){
        ex.printStackTrace();
    }
  }
}
```

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

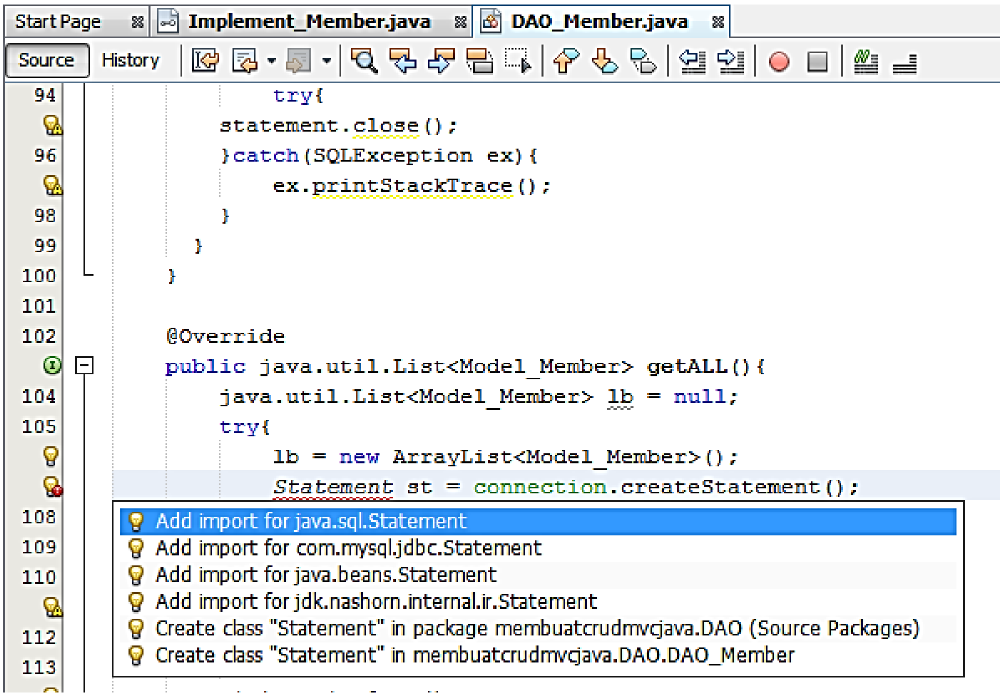


```
    @Override
    public java.util.List<Model_Member> getALL(){
        java.util.List<Model_Member> lb = null;
        try{
            lb = new ArrayList<Model_Member>();
```

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

```
        } finally {
            try{
        statement.close();
        }catch(SQLException ex){
            ex.printStackTrace();
        }
      }
    }

}
```

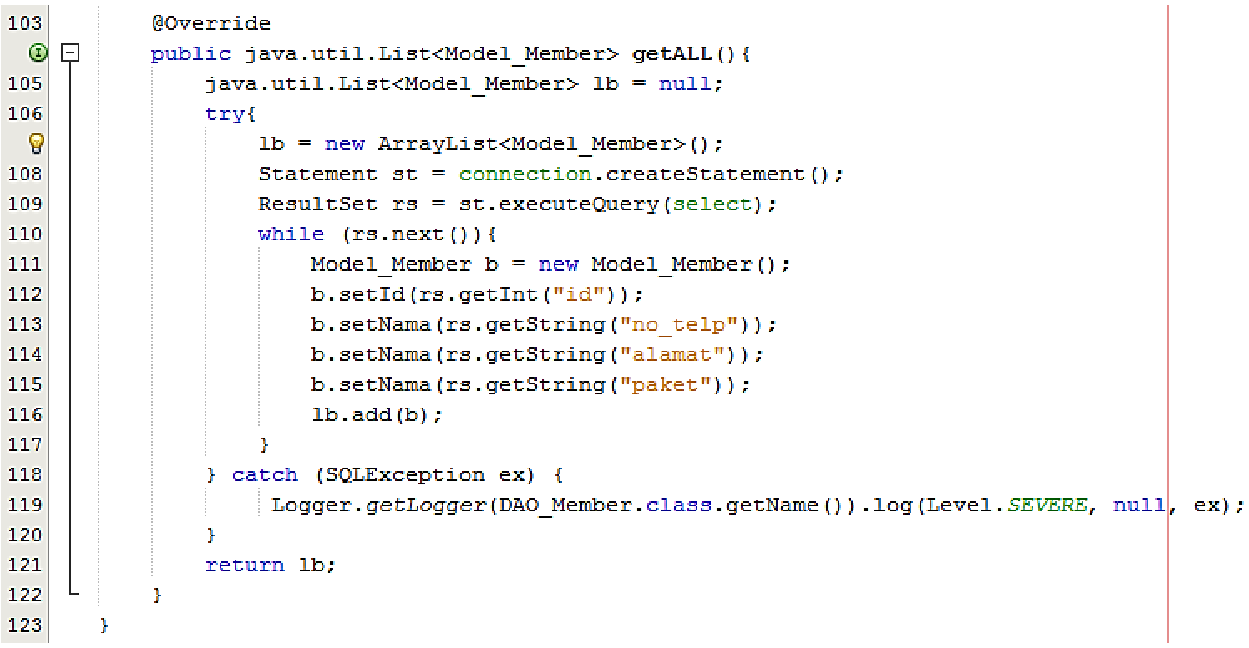
12. Selanjutnya pada syntax ***Statement st = connection.createStatement();*** Add ***import for java.sql.Statement***



```
            try{
        statement.close();
        }catch(SQLException ex){
            ex.printStackTrace();
        }
      }
    }

    @Override
    public java.util.List<Model_Member> getALL(){
        java.util.List<Model_Member> lb = null;
        try{
            lb = new ArrayList<Model_Member>();
            Statement st = connection.createStatement();
```

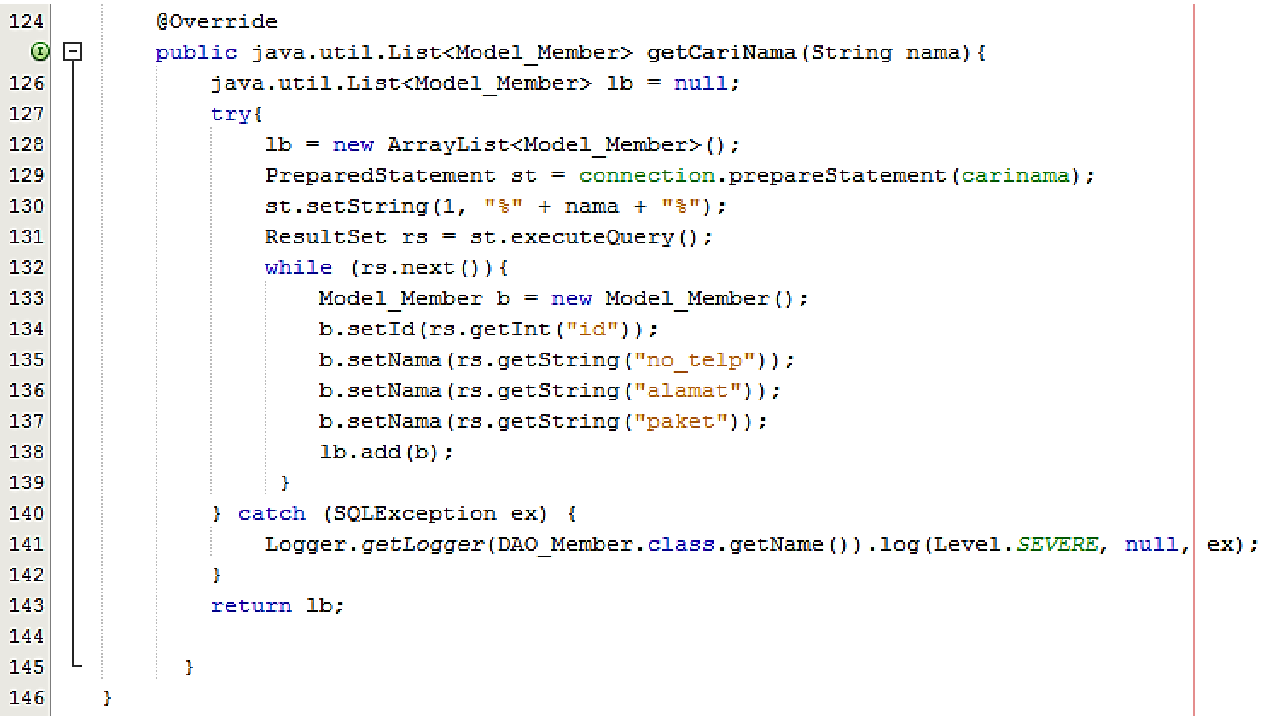
```
Add import for java.sql.Statement
Add import for com.mysql.jdbc.Statement
Add import for java.beans.Statement
Add import for jdk.nashorn.internal.ir.Statement
Create class "Statement" in package membuatcrudmvcjava.DAO (Source Packages)
Create class "Statement" in membuatcrudmvcjava.DAO.DAO_Member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

13. Berikut ini adalah hasil ***@Override getALL***



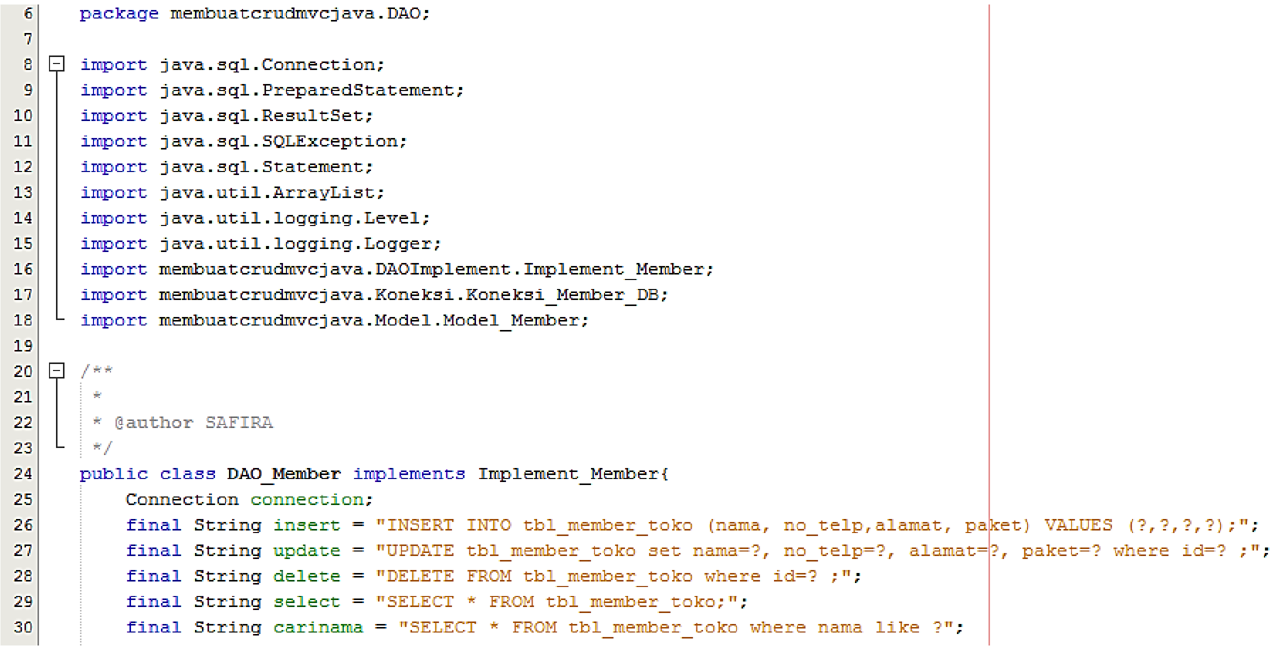
```
    @Override
    public java.util.List<Model_Member> getALL(){
        java.util.List<Model_Member> lb = null;
        try{
            lb = new ArrayList<Model_Member>();
            Statement st = connection.createStatement();
            ResultSet rs = st.executeQuery(select);
            while (rs.next()){
                Model_Member b = new Model_Member();
                b.setId(rs.getInt("id"));
                b.setNama(rs.getString("no_telp"));
                b.setNama(rs.getString("alamat"));
                b.setNama(rs.getString("paket"));
                lb.add(b);
            }
        } catch (SQLException ex) {
             Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
        }
        return lb;
    }
}
```

14. Selanjutnya meng ***@Override getCariNama*** seperti berikut

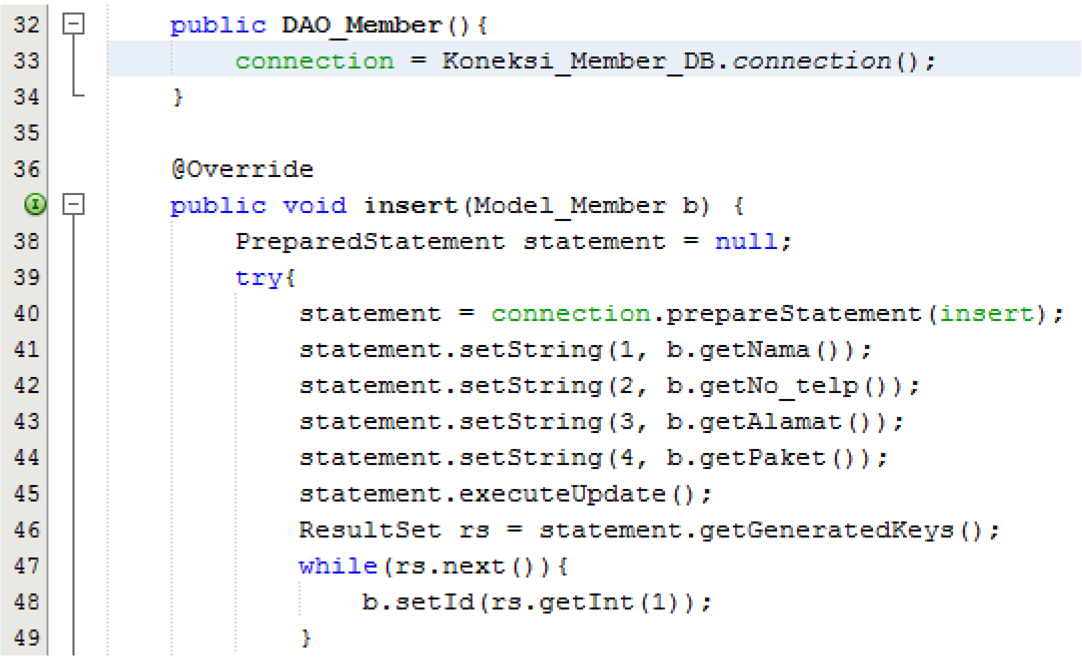


```
    @Override
    public java.util.List<Model_Member> getCariNama(String nama){
        java.util.List<Model_Member> lb = null;
        try{
            lb = new ArrayList<Model_Member>();
            PreparedStatement st = connection.prepareStatement(carinama);
            st.setString(1, "%" + nama + "%");
            ResultSet rs = st.executeQuery();
            while (rs.next()){
                Model_Member b = new Model_Member();
                b.setId(rs.getInt("id"));
                b.setNama(rs.getString("no_telp"));
                b.setNama(rs.getString("alamat"));
                b.setNama(rs.getString("paket"));
                lb.add(b);
             }
        } catch (SQLException ex) {
            Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
        }
        return lb;

      }
}
```

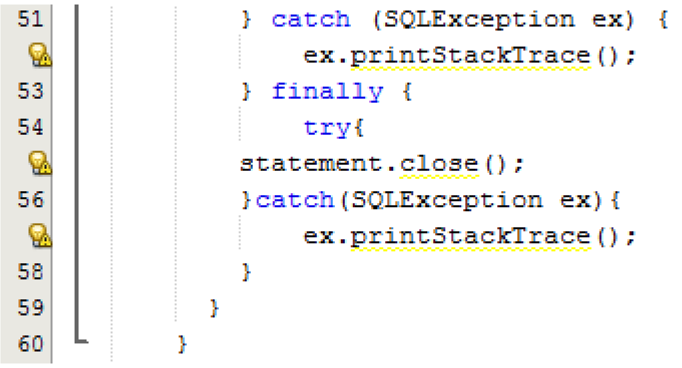
15. Jika syntax program diatas sudah dikerjakan semua, maka hasil akhir dari syntax program DAO sebagai berikut



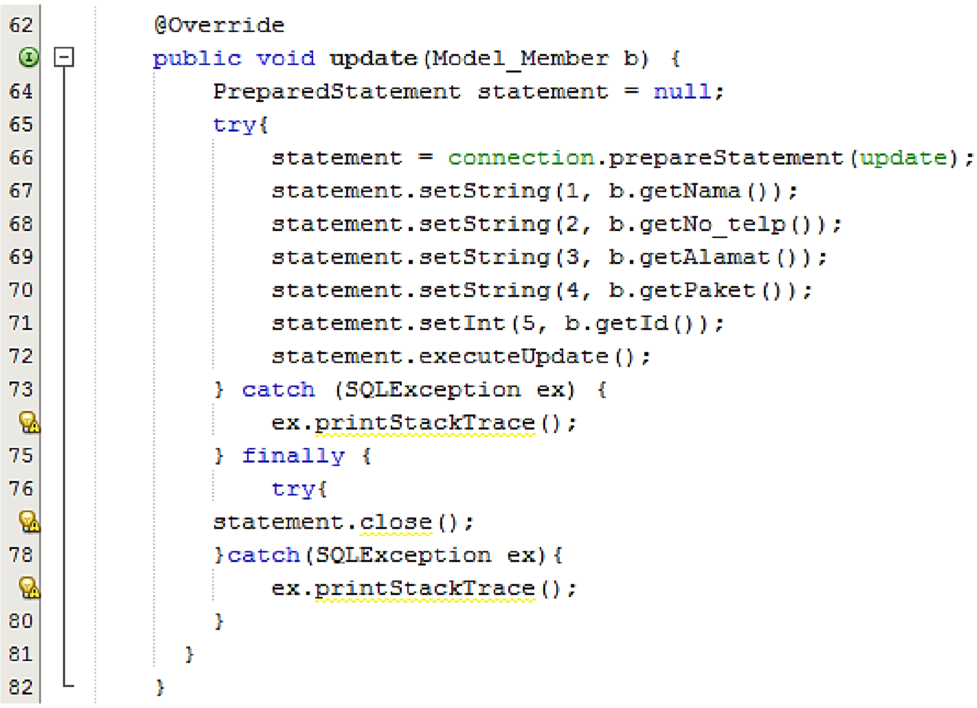
```java
package membuatcrudmvcjava.DAO;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import membuatcrudmvcjava.DAOImplement.Implement_Member;
import membuatcrudmvcjava.Koneksi.Koneksi_Member_DB;
import membuatcrudmvcjava.Model.Model_Member;

/**
 *
 * @author SAFIRA
 */
public class DAO_Member implements Implement_Member{
    Connection connection;
    final String insert = "INSERT INTO tbl_member_toko (nama, no_telp,alamat, paket) VALUES (?,?,?,?);";
    final String update = "UPDATE tbl_member_toko set nama=?, no_telp=?, alamat=?, paket=? where id=? ;";
    final String delete = "DELETE FROM tbl_member_toko where id=? ;";
    final String select = "SELECT * FROM tbl_member_toko;";
    final String carinama = "SELECT * FROM tbl_member_toko where nama like ?";
```



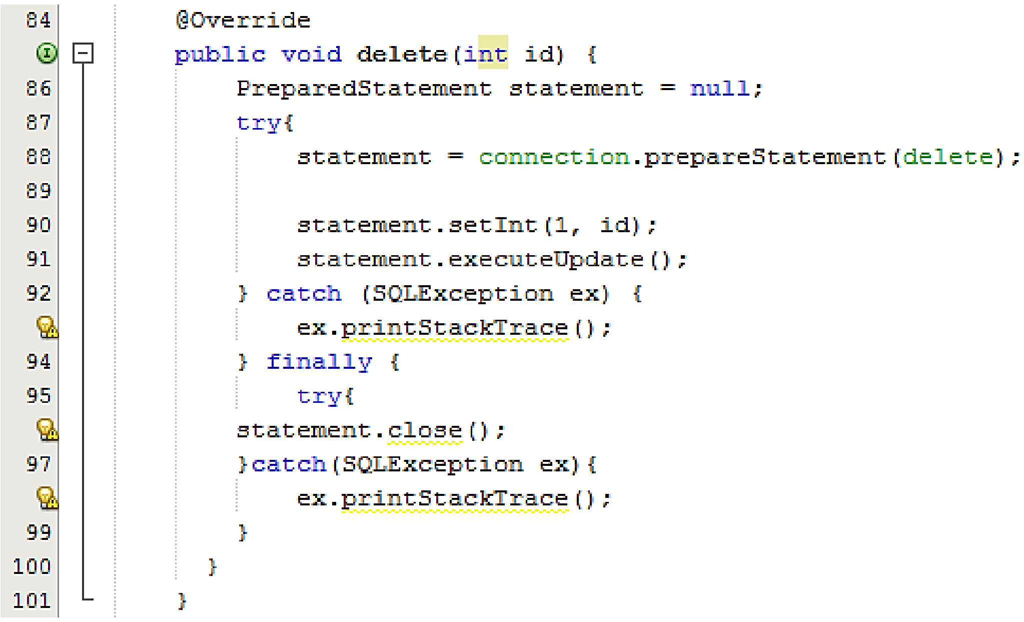
```java
    public DAO_Member(){
        connection = Koneksi_Member_DB.connection();
    }

    @Override
    public void insert(Model_Member b) {
        PreparedStatement statement = null;
        try{
            statement = connection.prepareStatement(insert);
            statement.setString(1, b.getNama());
            statement.setString(2, b.getNo_telp());
            statement.setString(3, b.getAlamat());
            statement.setString(4, b.getPaket());
            statement.executeUpdate();
            ResultSet rs = statement.getGeneratedKeys();
            while(rs.next()){
                b.setId(rs.getInt(1));
            }
```



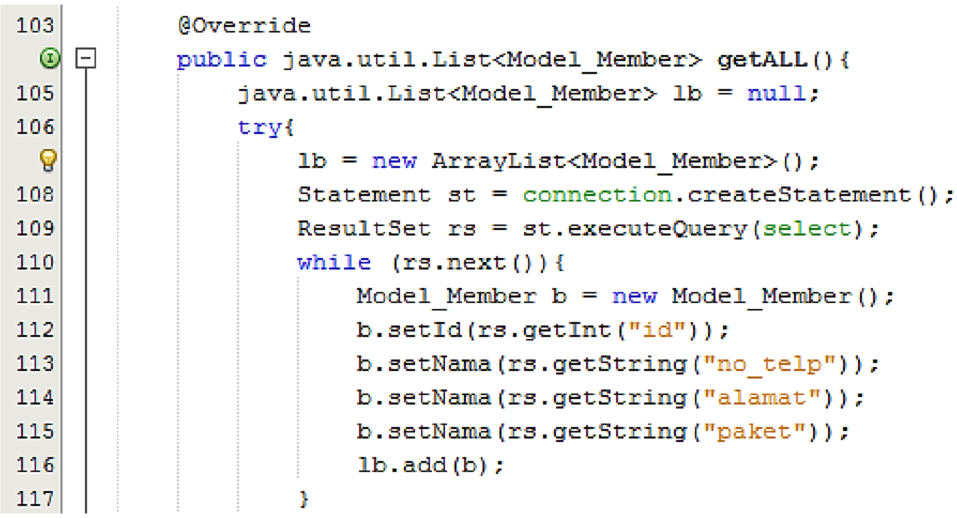
```java
        } catch (SQLException ex) {
            ex.printStackTrace();
        } finally {
            try{
        statement.close();
        }catch(SQLException ex){
            ex.printStackTrace();
        }
      }
    }
```



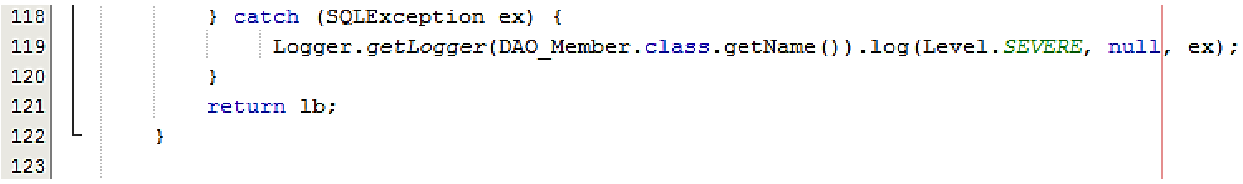
```
@Override
public void update(Model_Member b) {
    PreparedStatement statement = null;
    try{
        statement = connection.prepareStatement(update);
        statement.setString(1, b.getNama());
        statement.setString(2, b.getNo_telp());
        statement.setString(3, b.getAlamat());
        statement.setString(4, b.getPaket());
        statement.setInt(5, b.getId());
        statement.executeUpdate();
    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try{
    statement.close();
    }catch(SQLException ex){
        ex.printStackTrace();
    }
  }
}
```



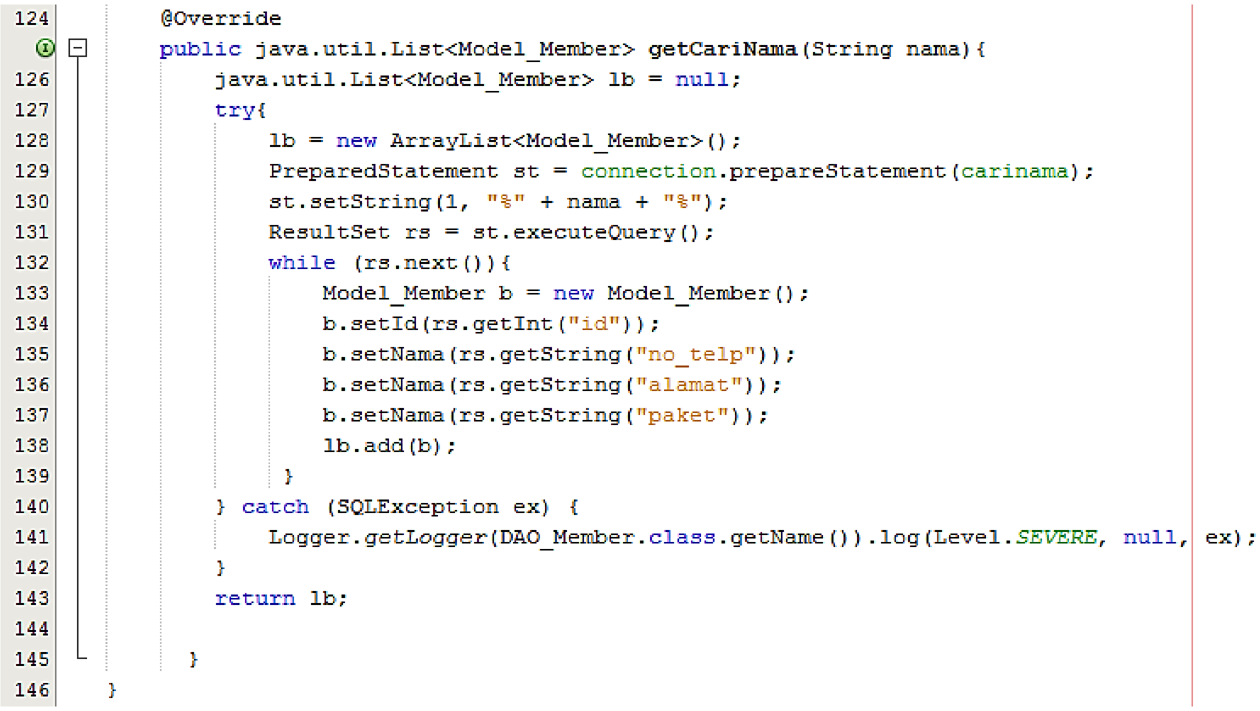
```
@Override
public void delete(int id) {
    PreparedStatement statement = null;
    try{
        statement = connection.prepareStatement(delete);

        statement.setInt(1, id);
        statement.executeUpdate();
    } catch (SQLException ex) {
        ex.printStackTrace();
    } finally {
        try{
    statement.close();
    }catch(SQLException ex){
        ex.printStackTrace();
    }
  }
}
```



```
@Override
public java.util.List<Model_Member> getALL(){
    java.util.List<Model_Member> lb = null;
    try{
        lb = new ArrayList<Model_Member>();
        Statement st = connection.createStatement();
        ResultSet rs = st.executeQuery(select);
        while (rs.next()){
            Model_Member b = new Model_Member();
            b.setId(rs.getInt("id"));
            b.setNama(rs.getString("no_telp"));
            b.setNama(rs.getString("alamat"));
            b.setNama(rs.getString("paket"));
            lb.add(b);
        }
```



```java
        } catch (SQLException ex) {
            Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
        }
        return lb;
    }

```



```java
    @Override
    public java.util.List<Model_Member> getCariNama(String nama){
        java.util.List<Model_Member> lb = null;
        try{
            lb = new ArrayList<Model_Member>();
            PreparedStatement st = connection.prepareStatement(carinama);
            st.setString(1, "%" + nama + "%");
            ResultSet rs = st.executeQuery();
            while (rs.next()){
                Model_Member b = new Model_Member();
                b.setId(rs.getInt("id"));
                b.setNama(rs.getString("no_telp"));
                b.setNama(rs.getString("alamat"));
                b.setNama(rs.getString("paket"));
                lb.add(b);
            }
        } catch (SQLException ex) {
            Logger.getLogger(DAO_Member.class.getName()).log(Level.SEVERE, null, ex);
        }
        return lb;

      }
}
```

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 ^_^