GUI & DATABASE

Arranged By:

Rajendra Rakha Arya Prabaswara

(1941720080/21)

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

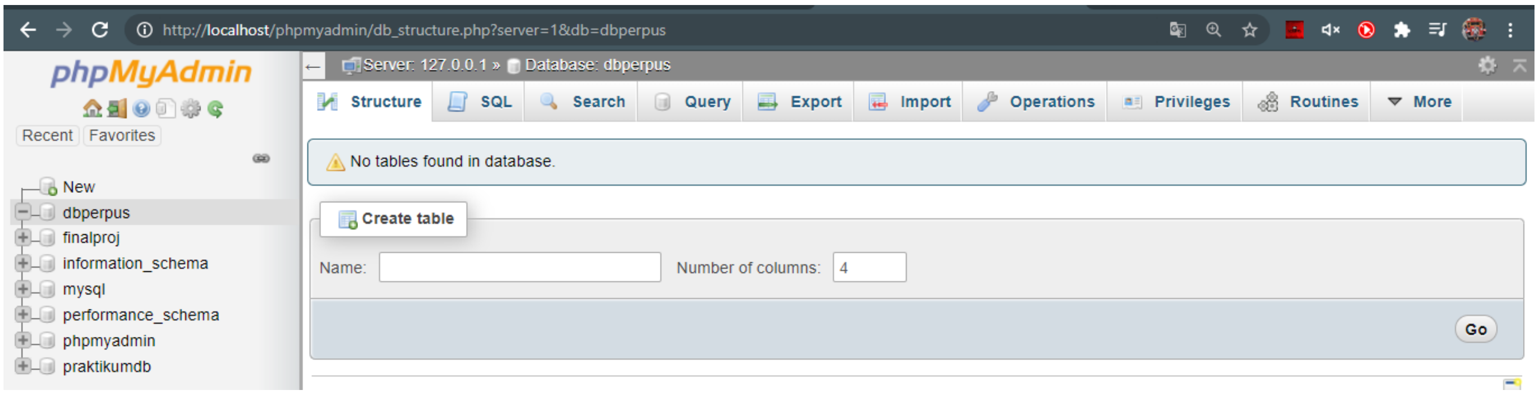
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

Experiment 1

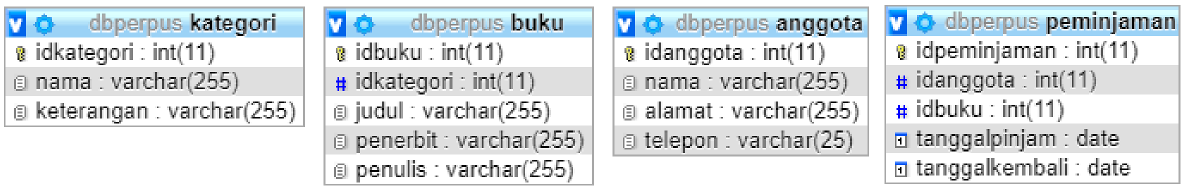
Creating a database.

1. The first step of this experiment is to create a database. Install XAMPP, open phpMyAdmin, create dbperpus database, and the tables:



No tables found in database.

Name: Number of columns: 4



dbperpus kategori
- idkategori : int(11)
- nama : varchar(255)
- keterangan : varchar(255)

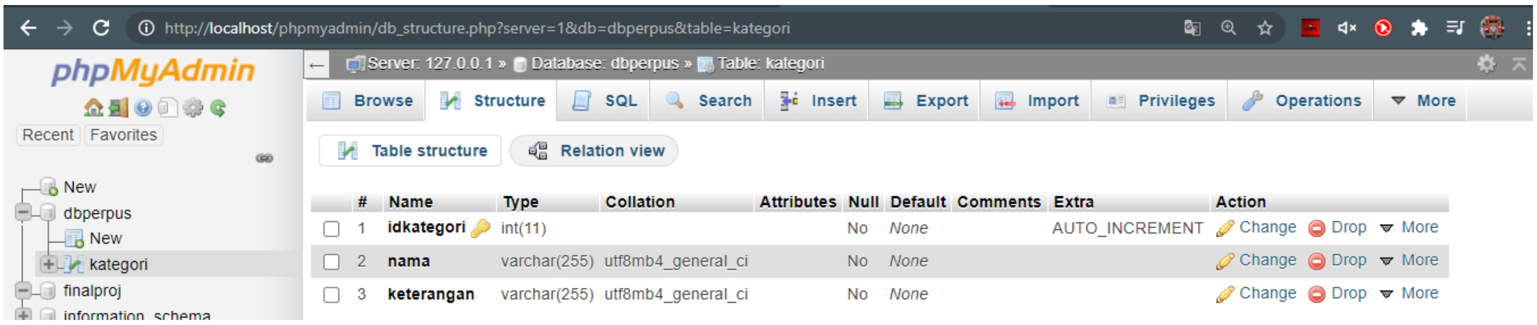
dbperpus buku
- idbuku : int(11)
- idkategori : int(11)
- judul : varchar(255)
- penerbit : varchar(255)
- penulis : varchar(255)

dbperpus anggota
- idanggota : int(11)
- nama : varchar(255)
- alamat : varchar(255)
- telepon : varchar(25)

dbperpus peminjaman
- idpeminjaman : int(11)
- idanggota : int(11)
- idbuku : int(11)
- tanggalpinjam : date
- tanggalkembali : date

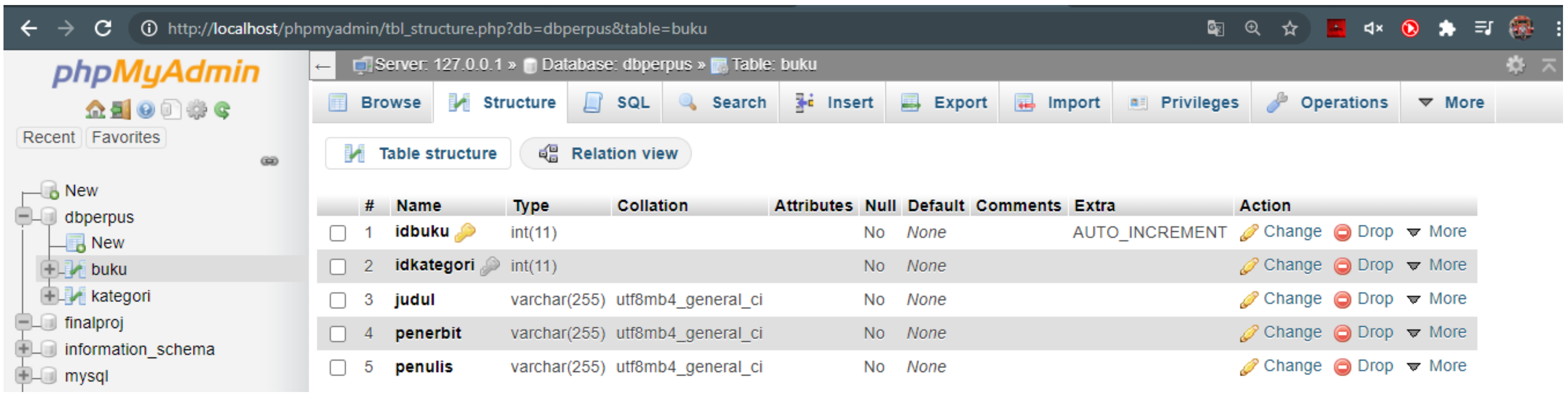
Set all id primary key in each table (idanggota, idkategori, idpeminjaman, idbuku) with Auto Increment.

➢ **Kategori**



#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idkategori	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	keterangan	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

➢ **Buku**

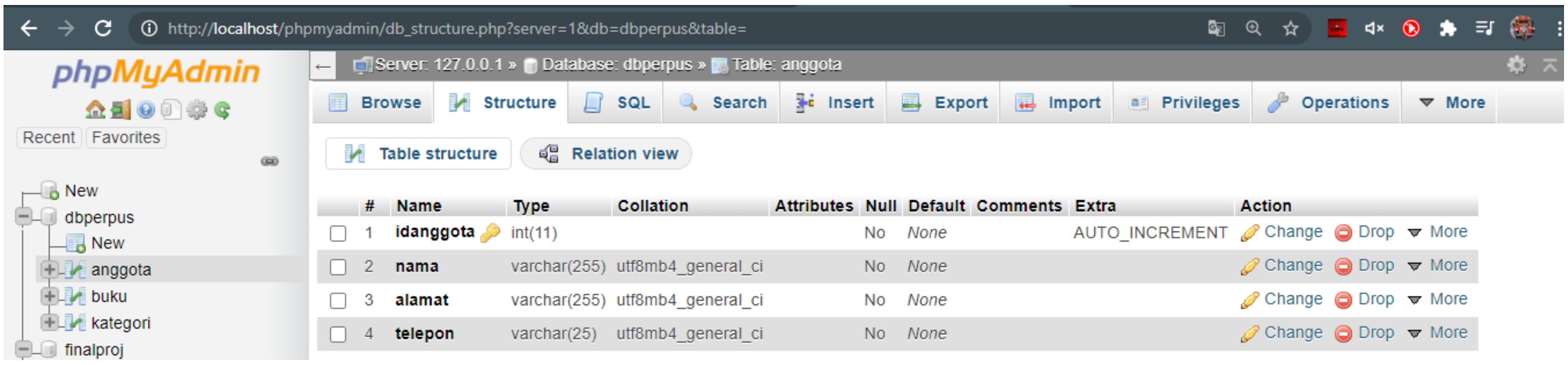


http://localhost/phpmyadmin/tbl_structure.php?db=dbperpus&table=buku

Server: 127.0.0.1 » Database: dbperpus » Table: buku

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idbuku	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	idkategori	int(11)			No	None			Change Drop More
3	judul	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	penerbit	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
5	penulis	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More

➢ **Anggota**

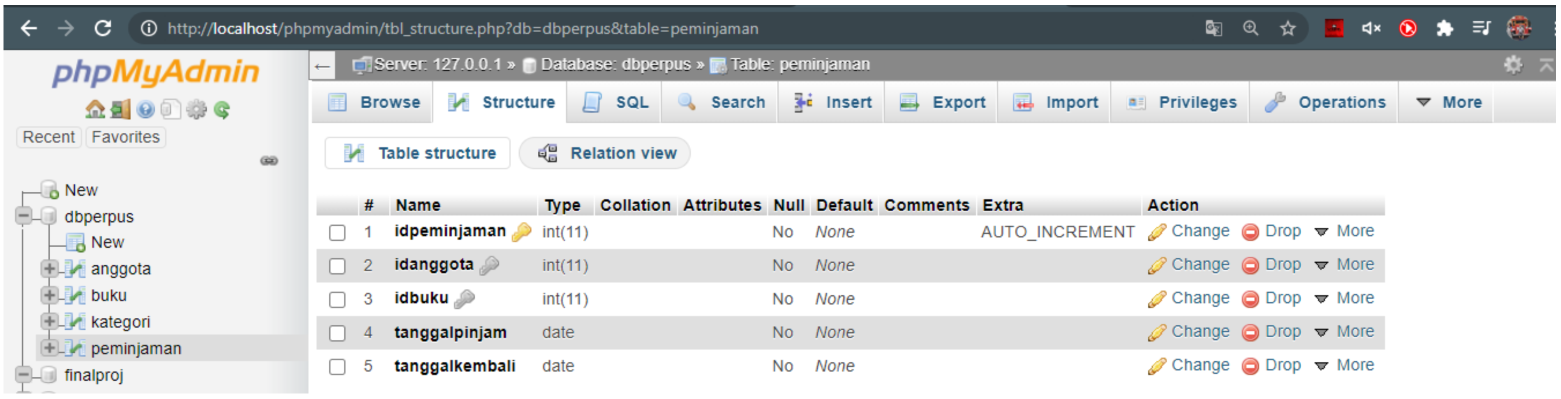


http://localhost/phpmyadmin/db_structure.php?server=1&db=dbperpus&table=

Server: 127.0.0.1 » Database: dbperpus » Table: anggota

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idanggota	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	nama	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
3	alamat	varchar(255)	utf8mb4_general_ci		No	None			Change Drop More
4	telepon	varchar(25)	utf8mb4_general_ci		No	None			Change Drop More

➢ **Peminjaman**



http://localhost/phpmyadmin/tbl_structure.php?db=dbperpus&table=peminjaman

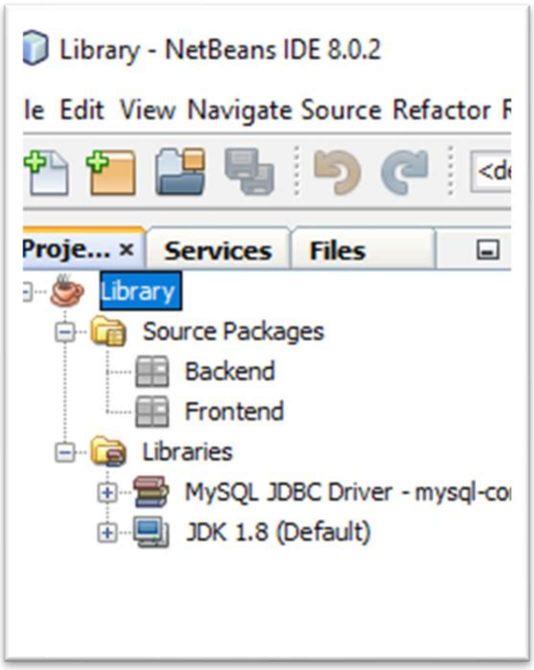
Server: 127.0.0.1 » Database: dbperpus » Table: peminjaman

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	idpeminjaman	int(11)			No	None		AUTO_INCREMENT	Change Drop More
2	idanggota	int(11)			No	None			Change Drop More
3	idbuku	int(11)			No	None			Change Drop More
4	tanggalpinjam	date			No	None			Change Drop More
5	tanggalkembali	date			No	None			Change Drop More

Experiment 2

Preparing project.

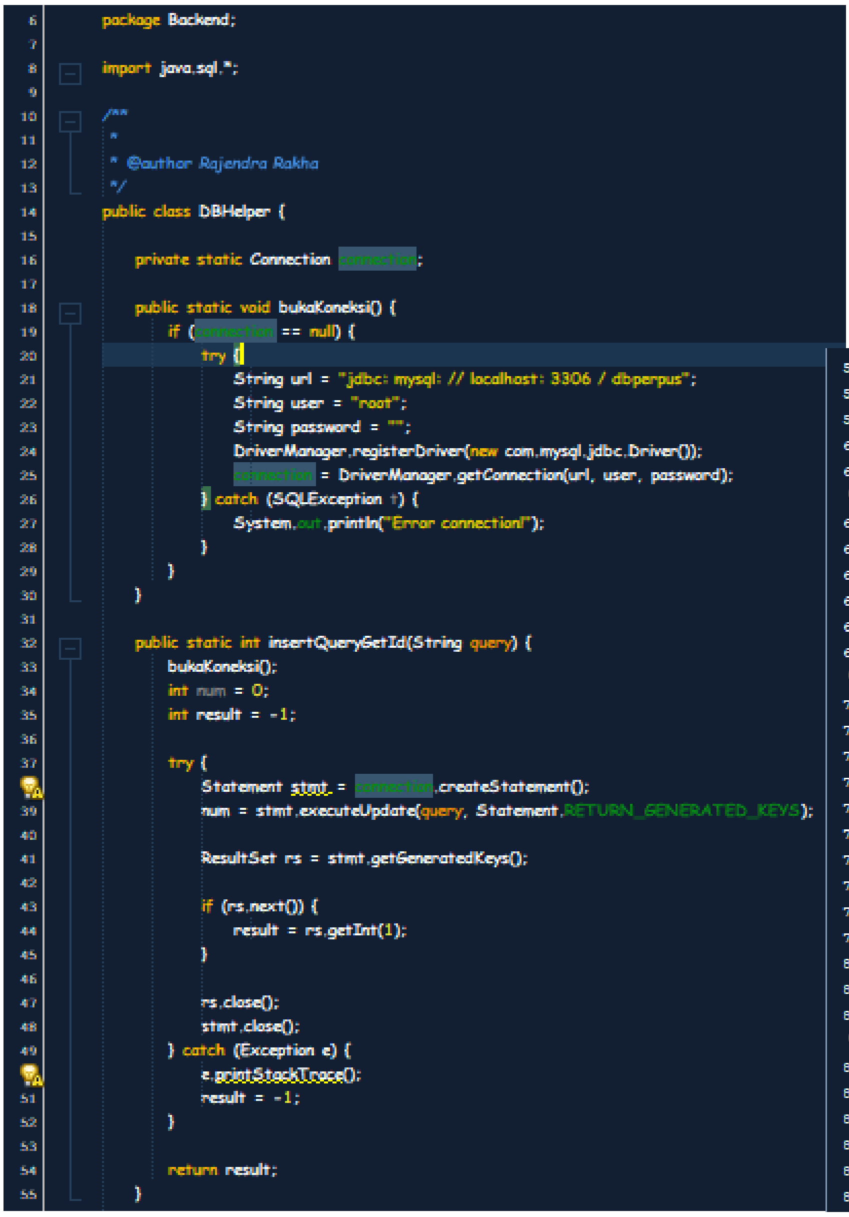
1. Create a new project, named Library.
2. In the project explorer, right click on Libraries→ Add Library, select the MySQL JDBC Driver.
3. Create package **frontend** and **backend**. How to create a package is, in the project explorer, right click on the Source Packages→ New → Java Package, give her the package name (frontend, backend).


Library - NetBeans IDE 8.0.2
le Edit View Navigate Source Refactor
Proje... ×
Services
Files
Library
Source Packages
Backend
Frontend
Libraries
MySQL JDBC Driver - mysql-co
JDK 1.8 (Default)

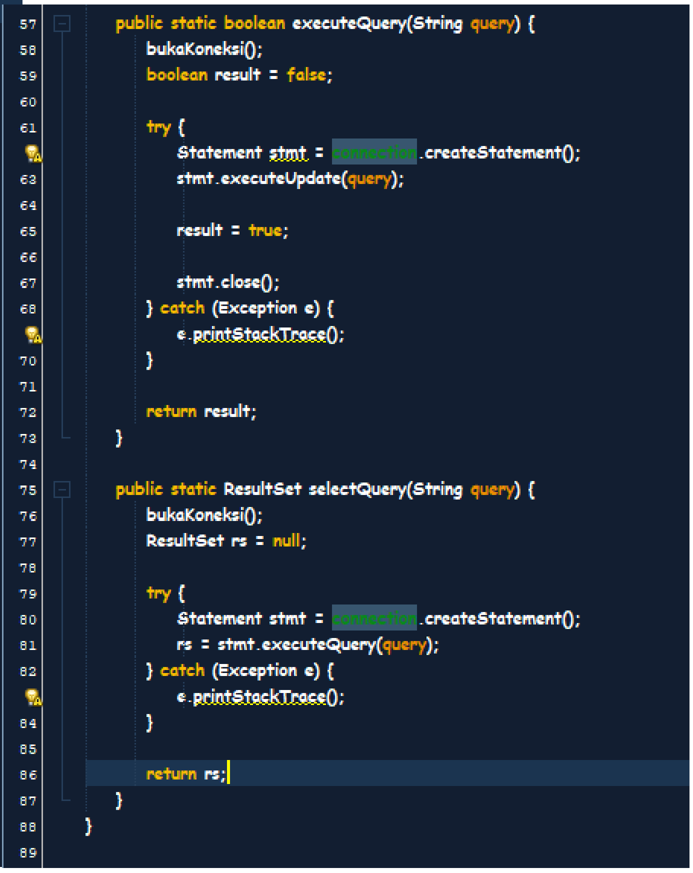
Experiment 3

Creating helper classes to execute SQL queries.

1. On the backend package, create DBHelper class.
2. Import java.sql. *
3. In this class there are those methods include:
    a. **bukaKoneksi ()**, To open a connection to the database
    b. insertQueryGetId (String query), to insert into the table and returns the ID generated by the database (yield Auto Increment).
    c. **executeQuery (String query)**, To execute a query that does not return a value (eg: insert, update, delete).
    d. **selectQuery (String query)**, To execute a select query which returns the query results.



```java
package Backend;

import java.sql.*;

/**
 *
 * @author Rajendra Rakha
 */
public class DBHelper {

    private static Connection connection;

    public static void bukaKoneksi() {
        if (connection == null) {
            try {
                String url = "jdbc: mysql: // localhost: 3306 / dbperpus";
                String user = "root";
                String password = "";
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                connection = DriverManager.getConnection(url, user, password);
            } catch (SQLException t) {
                System.out.println("Error connection!");
            }
        }
    }

    public static int insertQueryGetId(String query) {
        bukaKoneksi();
        int num = 0;
        int result = -1;

        try {
            Statement stmt = connection.createStatement();
            num = stmt.executeUpdate(query, Statement.RETURN_GENERATED_KEYS);

            ResultSet rs = stmt.getGeneratedKeys();

            if (rs.next()) {
                result = rs.getInt(1);
            }

            rs.close();
            stmt.close();
        } catch (Exception e) {
            e.printStackTrace();
            result = -1;
        }

        return result;
    }
```

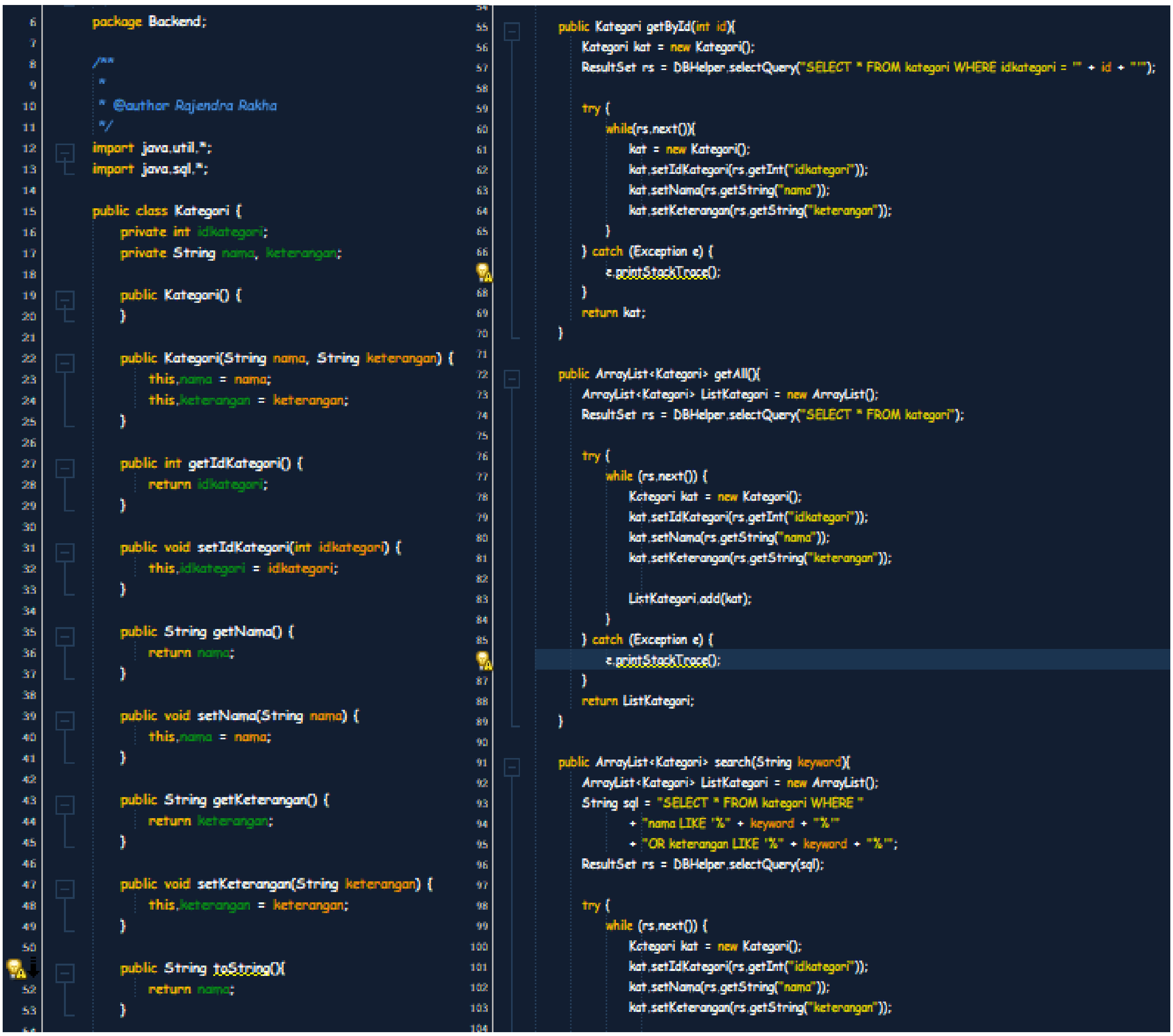


```java
    public static boolean executeQuery(String query) {
        bukaKoneksi();
        boolean result = false;

        try {
            Statement stmt = connection.createStatement();
            stmt.executeUpdate(query);

            result = true;

            stmt.close();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return result;
    }

    public static ResultSet selectQuery(String query) {
        bukaKoneksi();
        ResultSet rs = null;

        try {
            Statement stmt = connection.createStatement();
            rs = stmt.executeQuery(query);
        } catch (Exception e) {
            e.printStackTrace();
        }

        return rs;
    }
}
```

Experiment 4

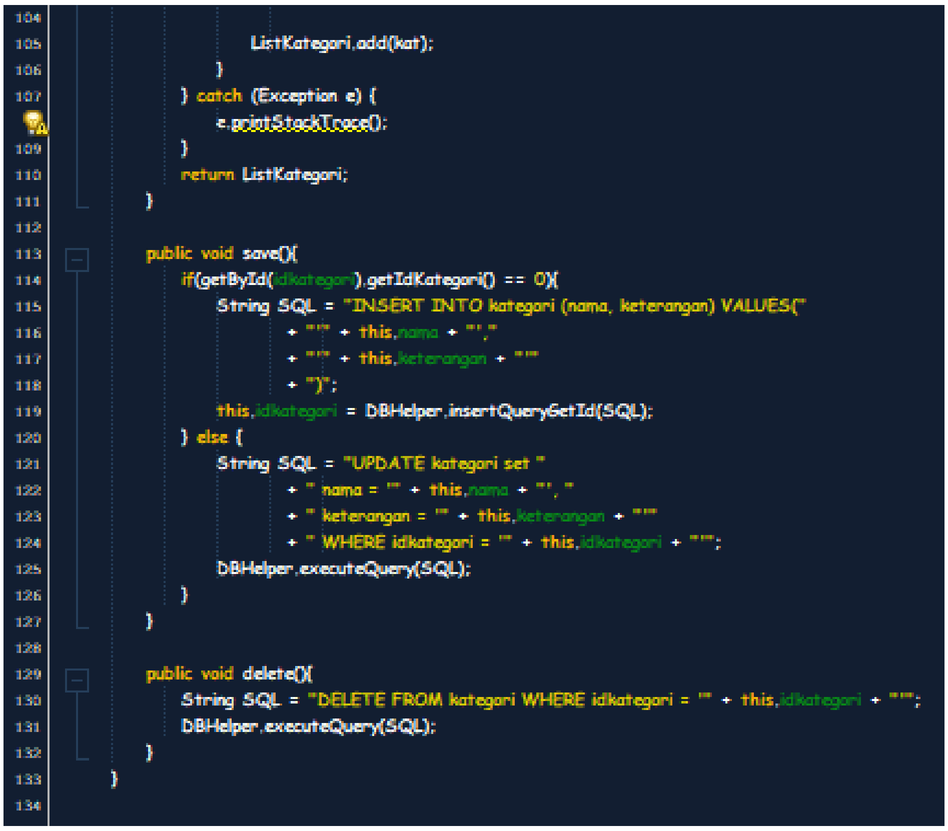
Creating a class to handle CRUD class in table **kategori**.

1. On the backend package, create a new class **Kategori**.
2. Add import java.util.ArrayList and java.sql. *
3. Add attributes corresponding fields in the tables Kategori.
4. Add getters setters for each attribute. You can use the NetBeans Code Insert facility. To do this, right-click anywhere in the editor, choose Insert Code, select Setter and Getter, check all of the attributes (idkategori, name, description).
5. Add custom default constructor and a constructor, which is used to set the name and description attributes. Idkategori attribute can not be set, because this id will be generated automatically through AutoIncrement features in MySQL.
6. Add method getById () to get a class object in the database by its id.
7. Add method getAll () to get all the data categories in the database, and accommodated to the ArrayList <Kategori>.
8. Add method search () in order to perform data searches. This method is similar to the method getAll () but different querynya.
9. Add method save (). This method has two functions, namely the insert and update. If the data is entered yet (idkategori = 0) then it will automatically insert. If the input data already exists, it automatically updates.
10. Add method delete () to do the removal operation on the table in the database Kategori.



```java
package Backend;

/**
 *
 * @author Rajendra Rakha
 */
import java.util.*;
import java.sql.*;

public class Kategori {
    private int idkategori;
    private String nama, keterangan;

    public Kategori() {
    }

    public Kategori(String nama, String keterangan) {
        this.nama = nama;
        this.keterangan = keterangan;
    }

    public int getIdKategori() {
        return idkategori;
    }

    public void setIdKategori(int idkategori) {
        this.idkategori = idkategori;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getKeterangan() {
        return keterangan;
    }

    public void setKeterangan(String keterangan) {
        this.keterangan = keterangan;
    }

    public String toString(){
        return nama;
    }
```

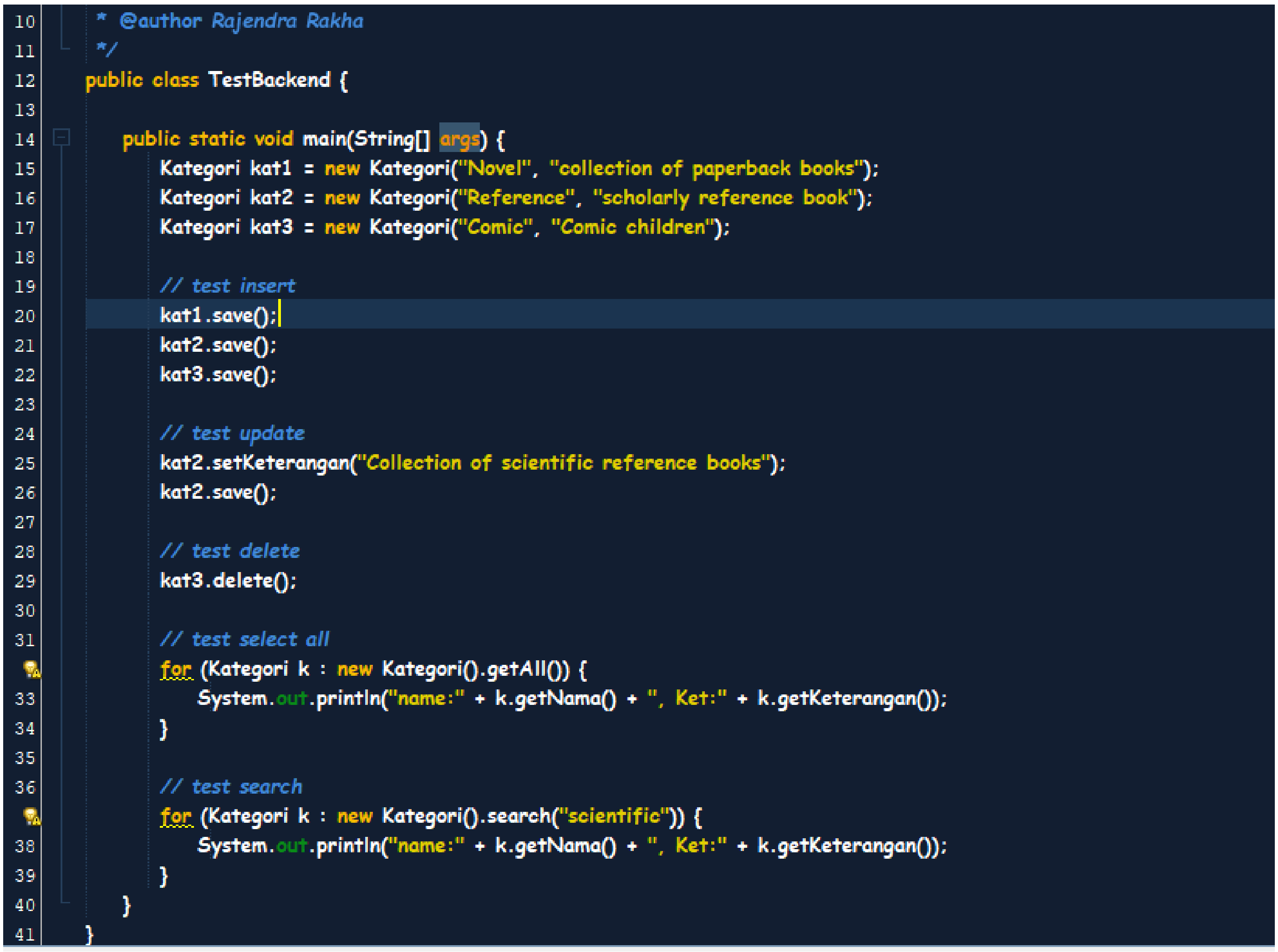
```java
    public Kategori getById(int id){
        Kategori kat = new Kategori();
        ResultSet rs = DBHelper.selectQuery("SELECT * FROM kategori WHERE idkategori = '" + id + "'");

        try {
            while(rs.next()){
                kat = new Kategori();
                kat.setIdKategori(rs.getInt("idkategori"));
                kat.setNama(rs.getString("nama"));
                kat.setKeterangan(rs.getString("keterangan"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return kat;
    }

    public ArrayList<Kategori> getAll(){
        ArrayList<Kategori> ListKategori = new ArrayList();
        ResultSet rs = DBHelper.selectQuery("SELECT * FROM kategori");

        try {
            while (rs.next()) {
                Kategori kat = new Kategori();
                kat.setIdKategori(rs.getInt("idkategori"));
                kat.setNama(rs.getString("nama"));
                kat.setKeterangan(rs.getString("keterangan"));

                ListKategori.add(kat);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ListKategori;
    }

    public ArrayList<Kategori> search(String keyword){
        ArrayList<Kategori> ListKategori = new ArrayList();
        String sql = "SELECT * FROM kategori WHERE "
                + "nama LIKE '%" + keyword + "%' "
                + "OR keterangan LIKE '%" + keyword + "%'";
        ResultSet rs = DBHelper.selectQuery(sql);

        try {
            while (rs.next()) {
                Kategori kat = new Kategori();
                kat.setIdKategori(rs.getInt("idkategori"));
                kat.setNama(rs.getString("nama"));
                kat.setKeterangan(rs.getString("keterangan"));
```



```
                ListKategori.add(kat);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ListKategori;
    }

    public void save(){
        if(getById(idkategori).getIdKategori() == 0){
            String SQL = "INSERT INTO kategori (nama, keterangan) VALUES("
                    + "'" + this.nama + "',"
                    + "'" + this.keterangan + "'"
                    + ")";
            this.idkategori = DBHelper.insertQueryGetId(SQL);
        } else {
            String SQL = "UPDATE kategori set "
                    + " nama = '" + this.nama + "', "
                    + " keterangan = '" + this.keterangan + "'"
                    + " WHERE idkategori = '" + this.idkategori + "'";
            DBHelper.executeQuery(SQL);
        }
    }

    public void delete(){
        String SQL = "DELETE FROM kategori WHERE idkategori = '" + this.idkategori + "'";
        DBHelper.executeQuery(SQL);
    }
}
```

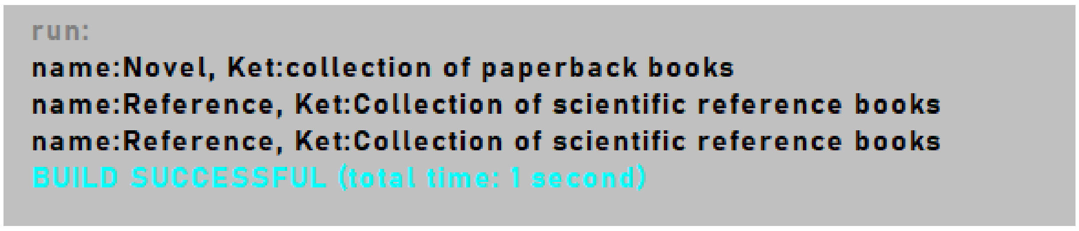
Experiment 5

Trying backed that has been made by operating through a text-based frontend (console). This experiment can you skip if you have been convinced that you have made the backend is functioning properly.



```java
 * @author Rajendra Rakha
 */
public class TestBackend {

    public static void main(String[] args) {
        Kategori kat1 = new Kategori("Novel", "collection of paperback books");
        Kategori kat2 = new Kategori("Reference", "scholarly reference book");
        Kategori kat3 = new Kategori("Comic", "Comic children");

        // test insert
        kat1.save();
        kat2.save();
        kat3.save();

        // test update
        kat2.setKeterangan("Collection of scientific reference books");
        kat2.save();

        // test delete
        kat3.delete();

        // test select all
        for (Kategori k : new Kategori().getAll()) {
            System.out.println("name:" + k.getNama() + ", Ket:" + k.getKeterangan());
        }

        // test search
        for (Kategori k : new Kategori().search("scientific")) {
            System.out.println("name:" + k.getNama() + ", Ket:" + k.getKeterangan());
        }
    }
}
```

OUTPUT

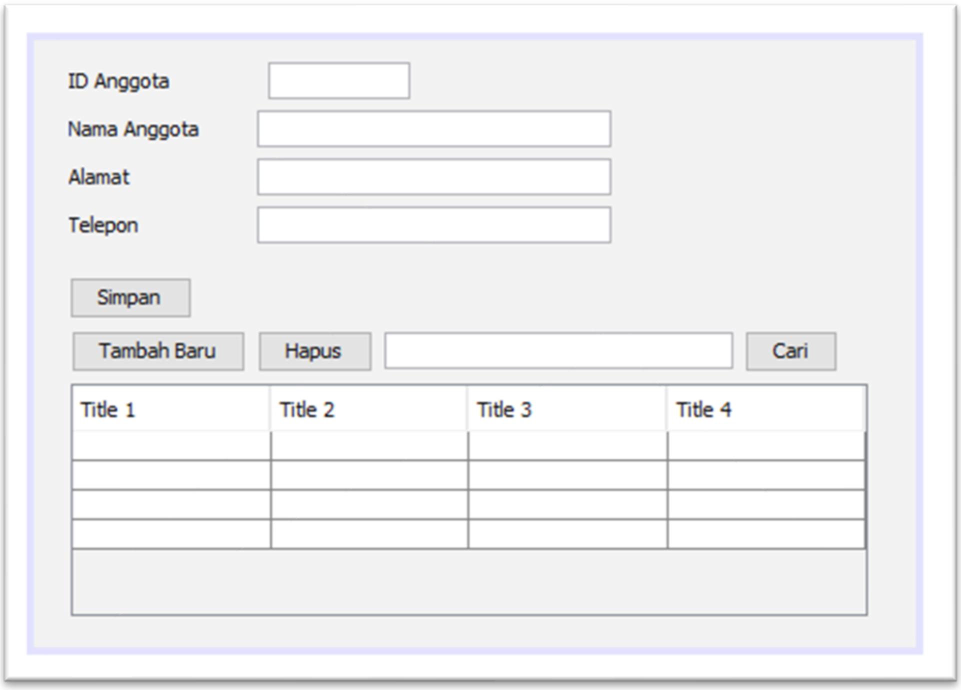


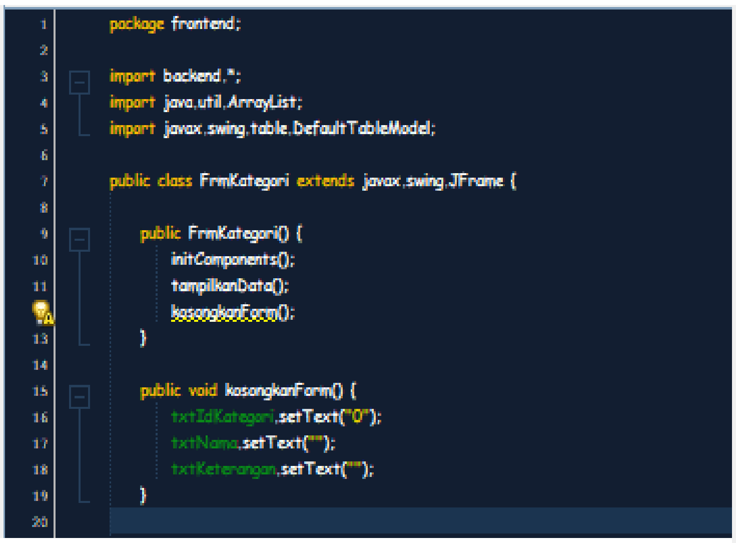
```
run:
name:Novel, Ket:collection of paperback books
name:Reference, Ket:Collection of scientific reference books
name:Reference, Ket:Collection of scientific reference books
BUILD SUCCESSFUL (total time: 1 second)
```

Experiment 6

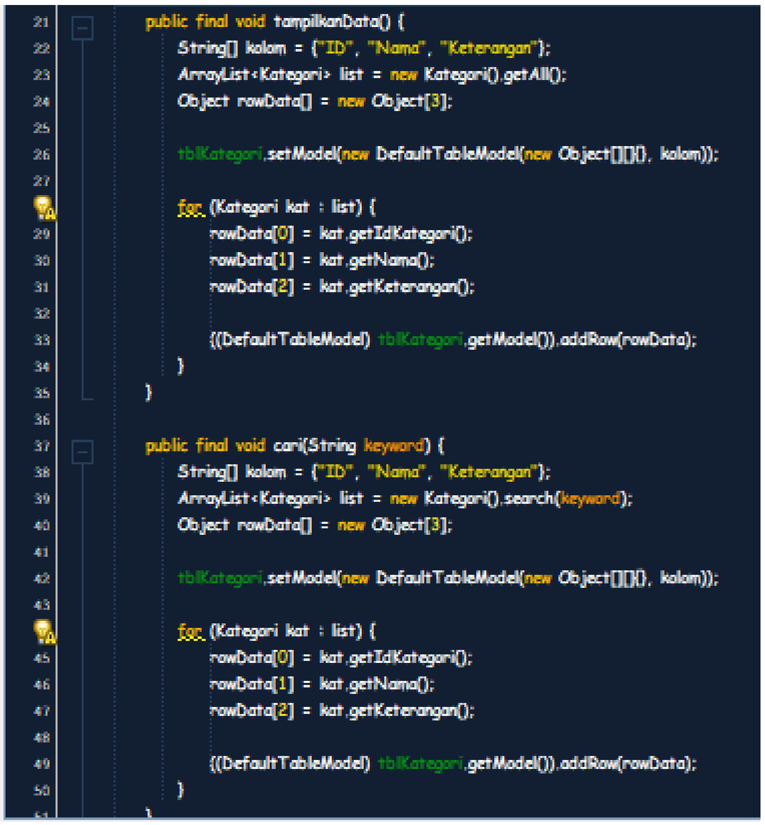
In this experiment we will create a GUI interface for the class Kategori.

1. In the frontend package, create a JFrame with FrmKategori name. To do this, right click on the package frontend→ New →JFrame Form.

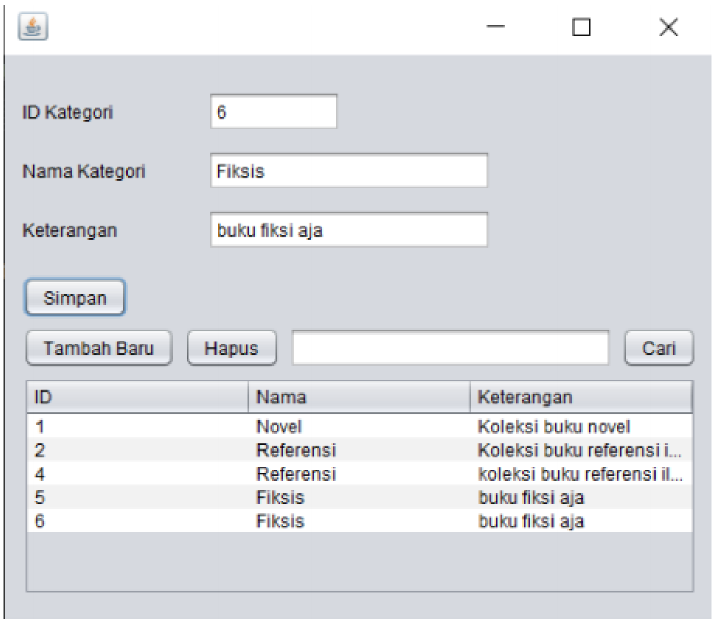




```java
package frontend;

import backend.*;
import java.util.ArrayList;
import javax.swing.table.DefaultTableModel;

public class FrmKategori extends javax.swing.JFrame {

    public FrmKategori() {
        initComponents();
        tampilkanData();
        kosongkanForm();
    }

    public void kosongkanForm() {
        txtIdKategori.setText("0");
        txtNama.setText("");
        txtKeterangan.setText("");
    }

```

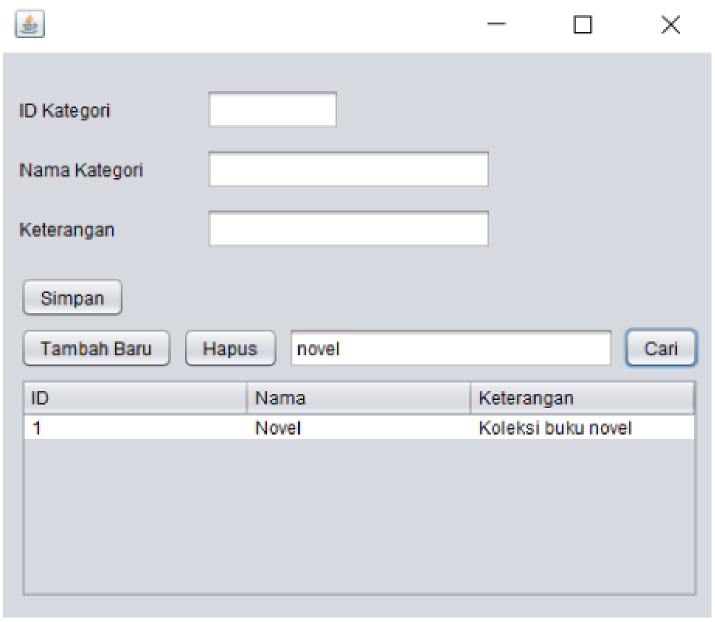


```java
    public final void tampilkanData() {
        String[] kolom = {"ID", "Nama", "Keterangan"};
        ArrayList<Kategori> list = new Kategori().getAll();
        Object rowData[] = new Object[3];

        tblKategori.setModel(new DefaultTableModel(new Object[][]{}, kolom));

        for (Kategori kat : list) {
            rowData[0] = kat.getIdKategori();
            rowData[1] = kat.getNama();
            rowData[2] = kat.getKeterangan();

            ((DefaultTableModel) tblKategori.getModel()).addRow(rowData);
        }
    }

    public final void cari(String keyword) {
        String[] kolom = {"ID", "Nama", "Keterangan"};
        ArrayList<Kategori> list = new Kategori().search(keyword);
        Object rowData[] = new Object[3];

        tblKategori.setModel(new DefaultTableModel(new Object[][]{}, kolom));

        for (Kategori kat : list) {
            rowData[0] = kat.getIdKategori();
            rowData[1] = kat.getNama();
            rowData[2] = kat.getKeterangan();

            ((DefaultTableModel) tblKategori.getModel()).addRow(rowData);
        }
    }
```



ID Kategori	6
Nama Kategori	Fiksis
Keterangan	buku fiksi aja

ID	Nama	Keterangan
1	Novel	Koleksi buku novel
2	Referensi	Koleksi buku referensi i...
4	Referensi	koleksi buku referensi il...
5	Fiksis	buku fiksi aja
6	Fiksis	buku fiksi aja



ID Kategori	
Nama Kategori	
Keterangan	

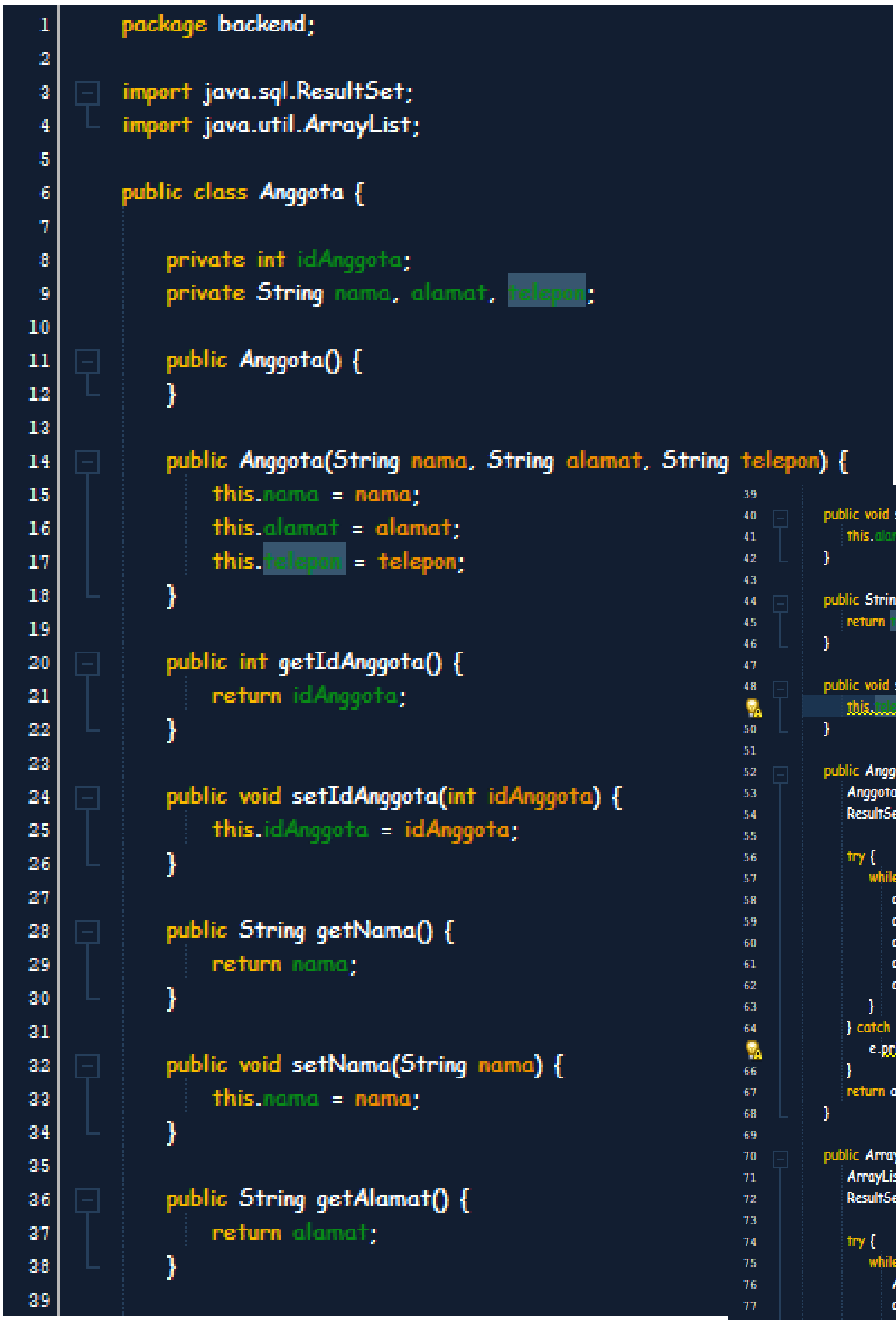
ID	Nama	Keterangan
1	Novel	Koleksi buku novel

]

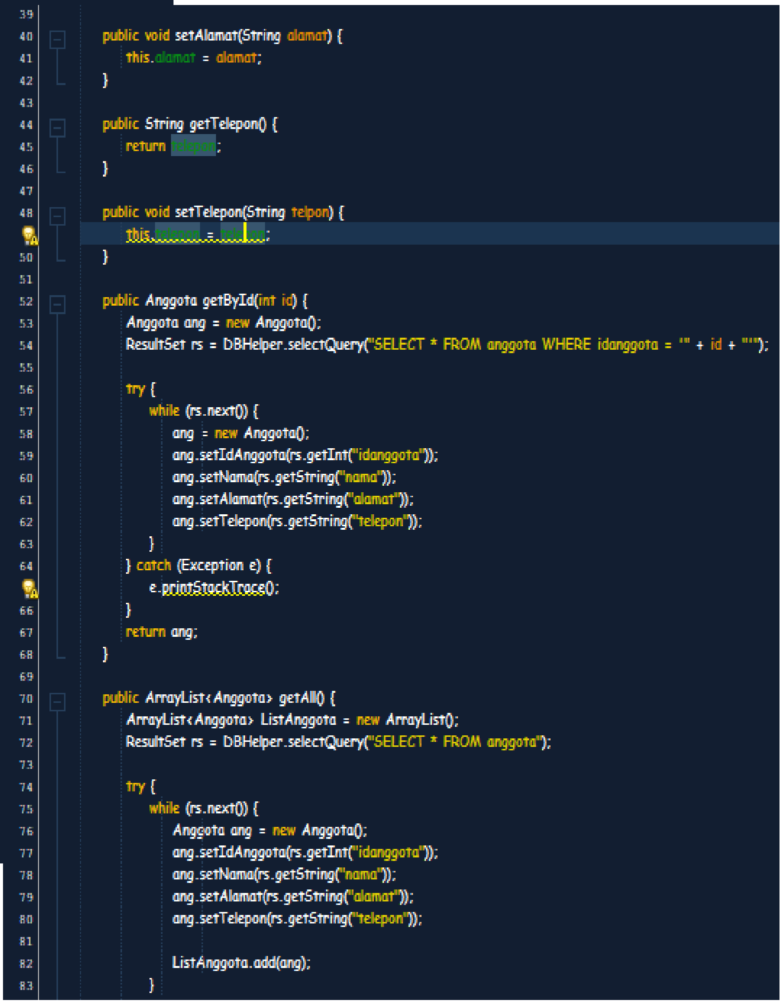
Experiment 6 Question

Do the same for Member data!

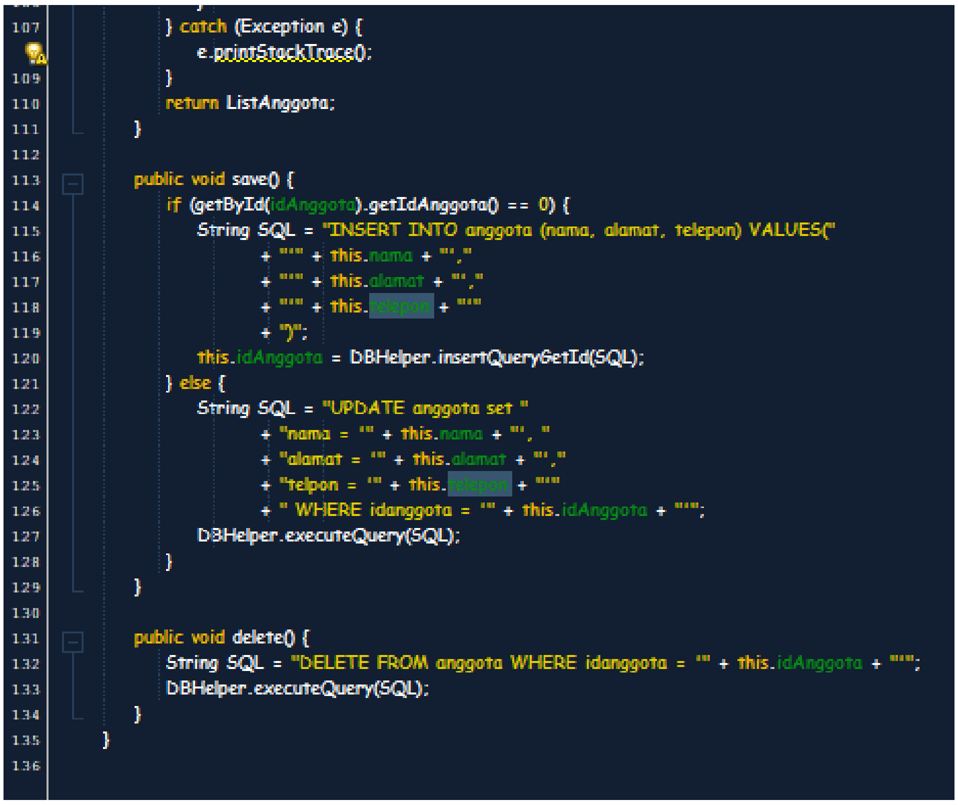
1. Create Member class on the backend package, complete its attributes and methods.
2. Perform test on TestBackend class on the frontend package.



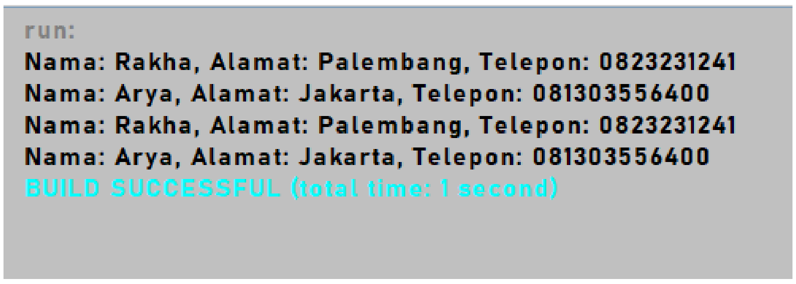
```java
package backend;

import java.sql.ResultSet;
import java.util.ArrayList;

public class Anggota {

    private int idAnggota;
    private String nama, alamat, telepon;

    public Anggota() {
    }

    public Anggota(String nama, String alamat, String telepon) {
        this.nama = nama;
        this.alamat = alamat;
        this.telepon = telepon;
    }

    public int getIdAnggota() {
        return idAnggota;
    }

    public void setIdAnggota(int idAnggota) {
        this.idAnggota = idAnggota;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public String getAlamat() {
        return alamat;
    }

```



```java

    public void setAlamat(String alamat) {
        this.alamat = alamat;
    }

    public String getTelepon() {
        return telepon;
    }

    public void setTelepon(String telpon) {
        this.telepon = telepon;
    }

    public Anggota getById(int id) {
        Anggota ang = new Anggota();
        ResultSet rs = DBHelper.selectQuery("SELECT * FROM anggota WHERE idanggota = '" + id + "'");

        try {
            while (rs.next()) {
                ang = new Anggota();
                ang.setIdAnggota(rs.getInt("idanggota"));
                ang.setNama(rs.getString("nama"));
                ang.setAlamat(rs.getString("alamat"));
                ang.setTelepon(rs.getString("telepon"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ang;
    }

    public ArrayList<Anggota> getAll() {
        ArrayList<Anggota> ListAnggota = new ArrayList();
        ResultSet rs = DBHelper.selectQuery("SELECT * FROM anggota");

        try {
            while (rs.next()) {
                Anggota ang = new Anggota();
                ang.setIdAnggota(rs.getInt("idanggota"));
                ang.setNama(rs.getString("nama"));
                ang.setAlamat(rs.getString("alamat"));
                ang.setTelepon(rs.getString("telepon"));

                ListAnggota.add(ang);
            }
```



```
        } catch (Exception e) {
            e.printStackTrace();
        }
        return ListAnggota;
    }

    public void save() {
        if (getById(idAnggota).getIdAnggota() == 0) {
            String SQL = "INSERT INTO anggota (nama, alamat, telepon) VALUES("
                    + "'" + this.nama + "',"
                    + "'" + this.alamat + "',"
                    + "'" + this.telepon + "'"
                    + ")";
            this.idAnggota = DBHelper.insertQueryGetId(SQL);
        } else {
            String SQL = "UPDATE anggota set "
                    + "nama = '" + this.nama + "', "
                    + "alamat = '" + this.alamat + "',"
                    + "telpon = '" + this.telepon + "'"
                    + " WHERE idanggota = '" + this.idAnggota + "'";
            DBHelper.executeQuery(SQL);
        }
    }

    public void delete() {
        String SQL = "DELETE FROM anggota WHERE idanggota = '" + this.idAnggota + "'";
        DBHelper.executeQuery(SQL);
    }
}
```

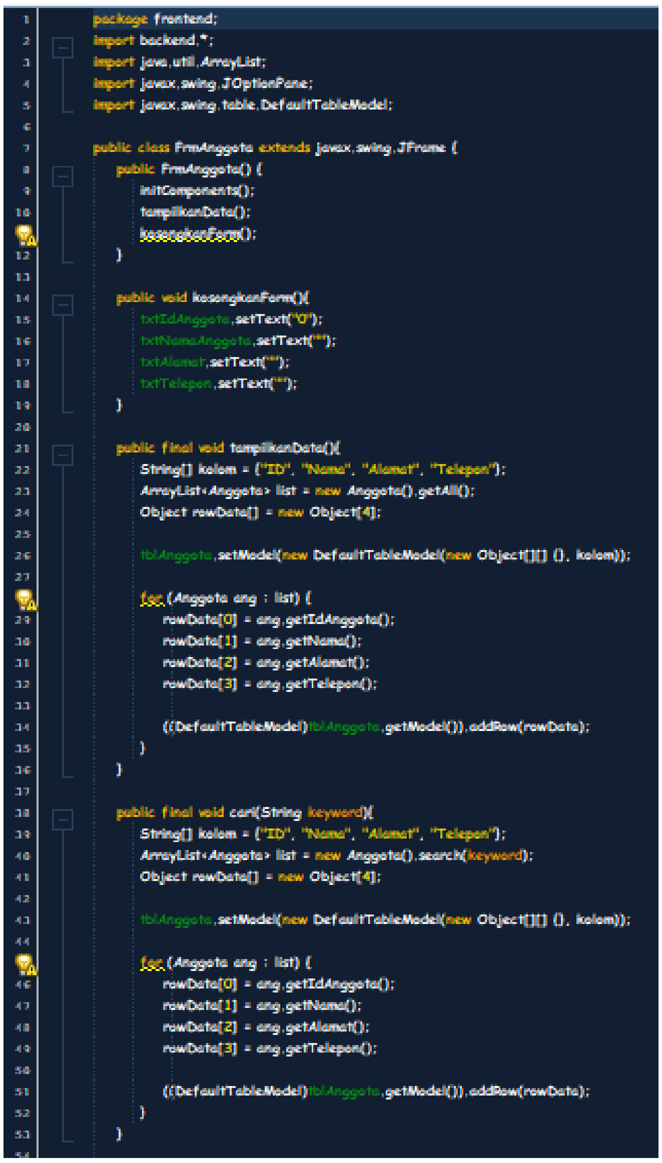


```
run:
Nama: Rakha, Alamat: Palembang, Telepon: 0823231241
Nama: Arya, Alamat: Jakarta, Telepon: 081303556400
Nama: Rakha, Alamat: Palembang, Telepon: 0823231241
Nama: Arya, Alamat: Jakarta, Telepon: 081303556400
BUILD SUCCESSFUL (total time: 1 second)
```

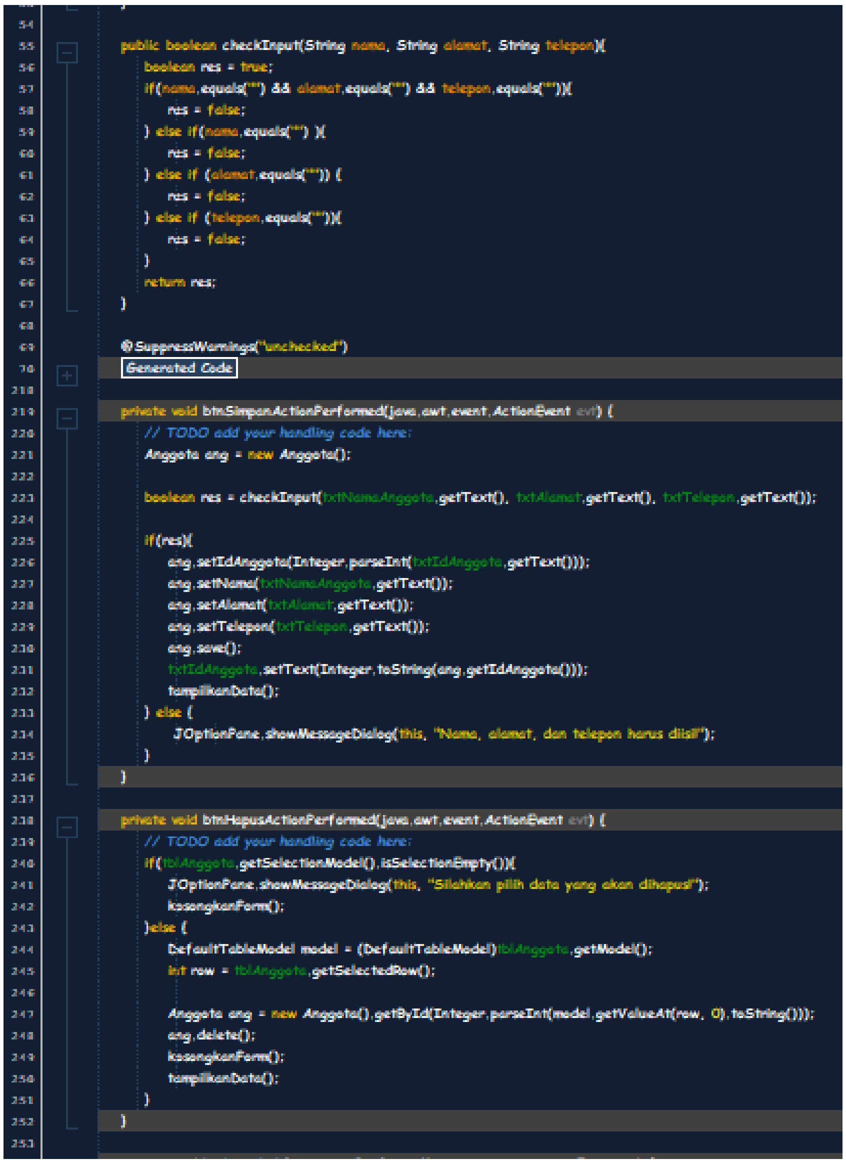
Experiment 7

Members create a form for data.

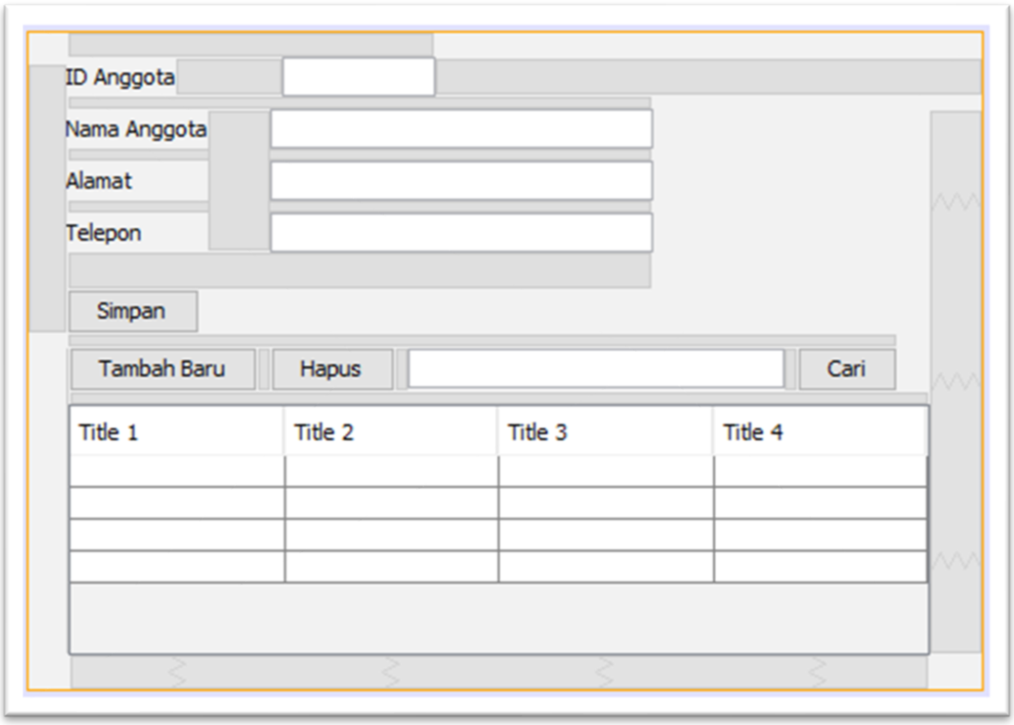
1. Create FrmAnggota on frontend and fill in the component package, the method and its events.

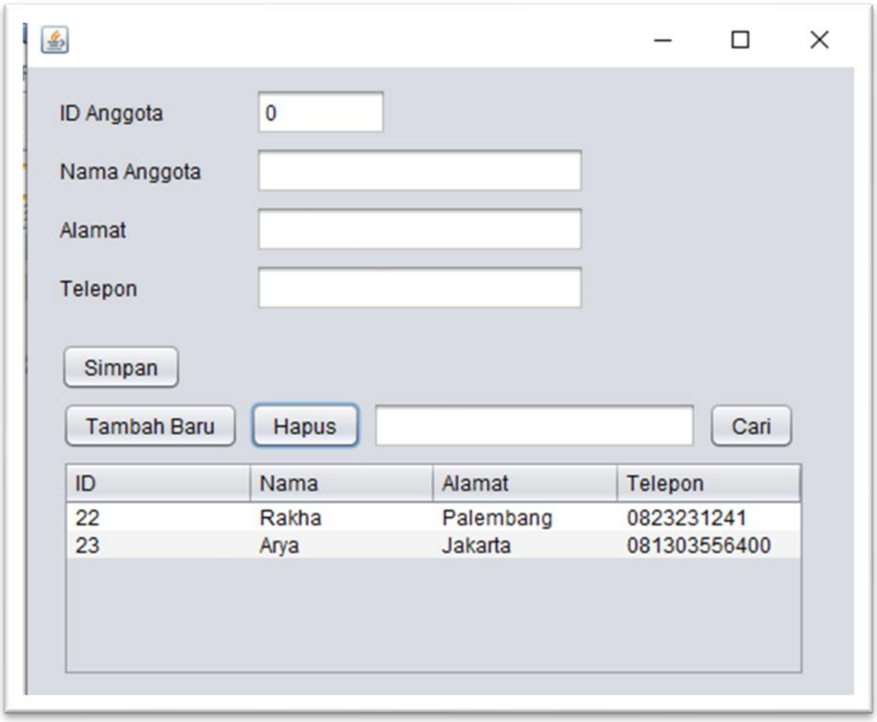


```java
package frontend;
import backend.*;
import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class FrmAnggota extends javax.swing.JFrame {
    public FrmAnggota() {
        initComponents();
        tampilkanData();
        kosongkanForm();
    }

    public void kosongkanForm(){
        txtIdAnggota.setText("0");
        txtNamaAnggota.setText("");
        txtAlamat.setText("");
        txtTelepon.setText("");
    }

    public final void tampilkanData(){
        String[] kolom = {"ID", "Nama", "Alamat", "Telepon"};
        ArrayList<Anggota> list = new Anggota().getAll();
        Object rowData[] = new Object[4];

        tblAnggota.setModel(new DefaultTableModel(new Object[][] {}, kolom));

        for (Anggota ang : list) {
            rowData[0] = ang.getIdAnggota();
            rowData[1] = ang.getNama();
            rowData[2] = ang.getAlamat();
            rowData[3] = ang.getTelepon();

            ((DefaultTableModel)tblAnggota.getModel()).addRow(rowData);
        }
    }

    public final void cari(String keyword){
        String[] kolom = {"ID", "Nama", "Alamat", "Telepon"};
        ArrayList<Anggota> list = new Anggota().search(keyword);
        Object rowData[] = new Object[4];

        tblAnggota.setModel(new DefaultTableModel(new Object[][] {}, kolom));

        for (Anggota ang : list) {
            rowData[0] = ang.getIdAnggota();
            rowData[1] = ang.getNama();
            rowData[2] = ang.getAlamat();
            rowData[3] = ang.getTelepon();

            ((DefaultTableModel)tblAnggota.getModel()).addRow(rowData);
        }
    }
```



```java
    }

    public boolean checkInput(String nama, String alamat, String telepon){
        boolean res = true;
        if(nama.equals("") && alamat.equals("") && telepon.equals("")){
            res = false;
        } else if(nama.equals("") ){
            res = false;
        } else if (alamat.equals("")) {
            res = false;
        } else if (telepon.equals("")){
            res = false;
        }
        return res;
    }

    @SuppressWarnings("unchecked")
    Generated Code

    private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Anggota ang = new Anggota();

        boolean res = checkInput(txtNamaAnggota.getText(), txtAlamat.getText(), txtTelepon.getText());

        if(res){
            ang.setIdAnggota(Integer.parseInt(txtIdAnggota.getText()));
            ang.setNama(txtNamaAnggota.getText());
            ang.setAlamat(txtAlamat.getText());
            ang.setTelepon(txtTelepon.getText());
            ang.save();
            txtIdAnggota.setText(Integer.toString(ang.getIdAnggota()));
            tampilkanData();
        } else {
             JOptionPane.showMessageDialog(this, "Nama, alamat, dan telepon harus diisi!");
        }
    }

    private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if(tblAnggota.getSelectionModel().isSelectionEmpty()){
            JOptionPane.showMessageDialog(this, "Silahkan pilih data yang akan dihapus!");
            kosongkanForm();
        }else {
            DefaultTableModel model = (DefaultTableModel)tblAnggota.getModel();
            int row = tblAnggota.getSelectedRow();

            Anggota ang = new Anggota().getById(Integer.parseInt(model.getValueAt(row, 0).toString()));
            ang.delete();
            kosongkanForm();
            tampilkanData();
        }
    }
```





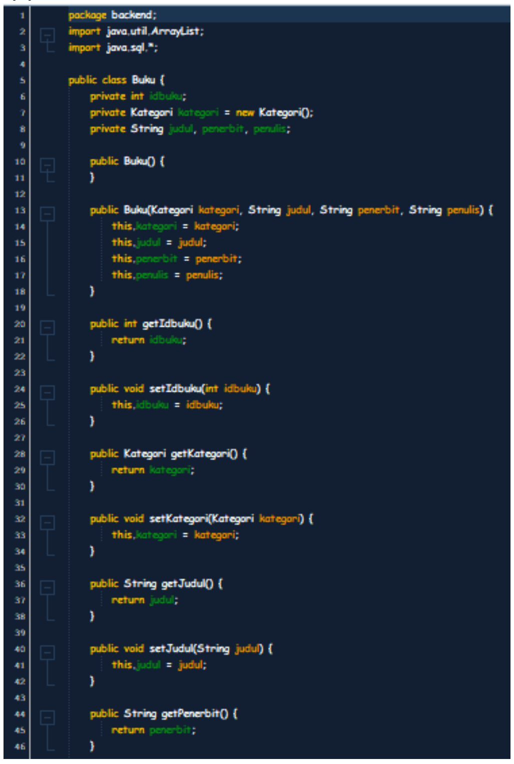
ID Anggota: 0

ID	Nama	Alamat	Telepon
22	Rakha	Palembang	0823231241
23	Arya	Jakarta	081303556400

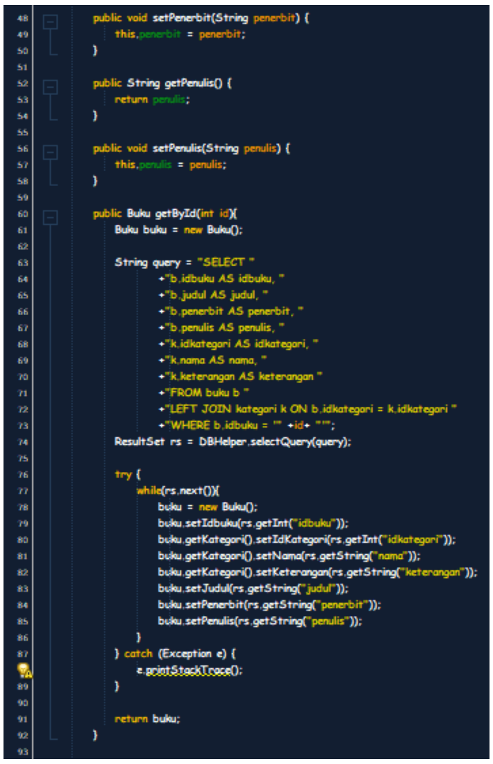
For Book data, approximately the same way as the data Kategori and members. Only different is:

a. Dialing getKategori (). GetIdKategori () on the query, insert and update to set idkategori in table book
b. SELECT queries involving join table on getById method (), getAll () and search ().

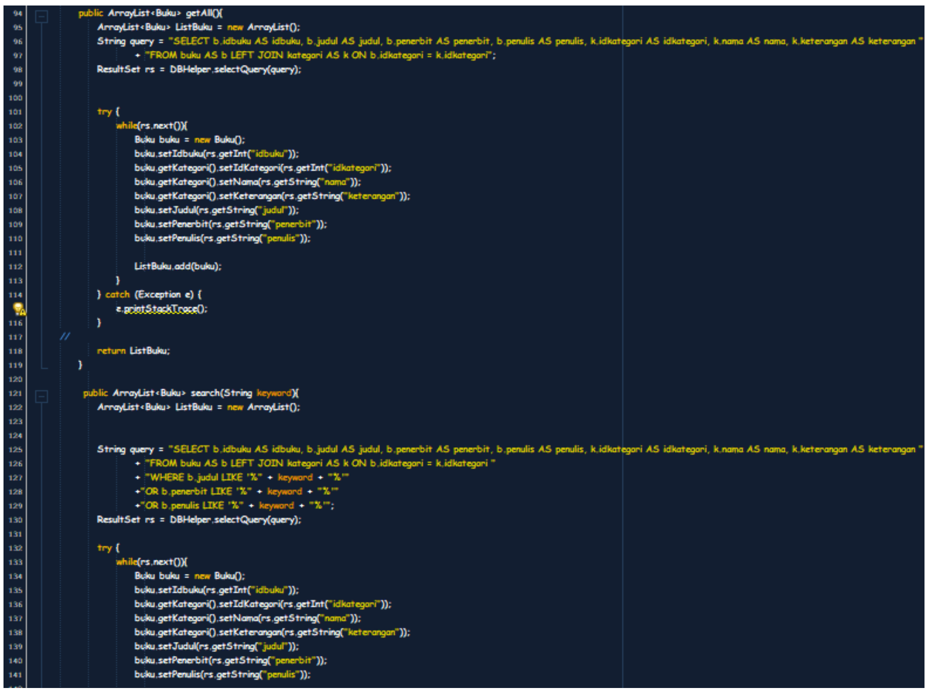
The complete code book class you can see in Appendix 1. To test book on the frontend, you can see in Appendix 2.



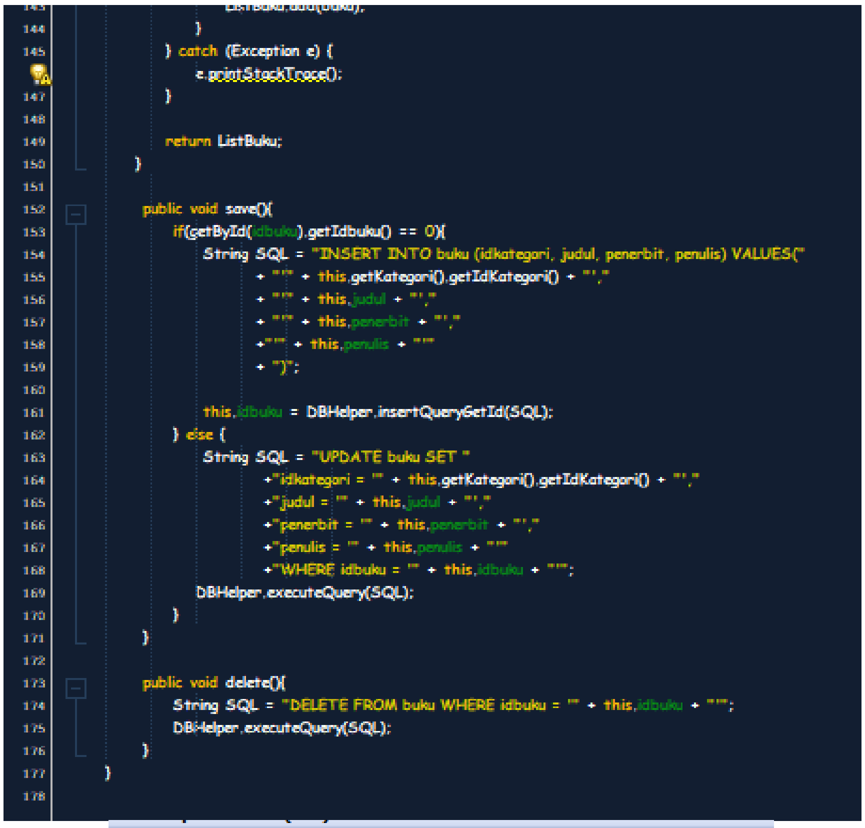
```java
package backend;
import java.util.ArrayList;
import java.sql.*;

public class Buku {
    private int idbuku;
    private Kategori kategori = new Kategori();
    private String judul, penerbit, penulis;

    public Buku() {
    }

    public Buku(Kategori kategori, String judul, String penerbit, String penulis) {
        this.kategori = kategori;
        this.judul = judul;
        this.penerbit = penerbit;
        this.penulis = penulis;
    }

    public int getIdbuku() {
        return idbuku;
    }

    public void setIdbuku(int idbuku) {
        this.idbuku = idbuku;
    }

    public Kategori getKategori() {
        return kategori;
    }

    public void setKategori(Kategori kategori) {
        this.kategori = kategori;
    }

    public String getJudul() {
        return judul;
    }

    public void setJudul(String judul) {
        this.judul = judul;
    }

    public String getPenerbit() {
        return penerbit;
    }
```



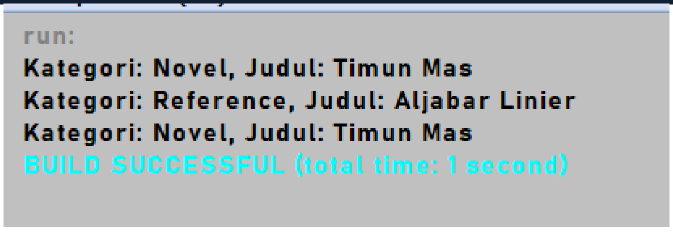
```java
    public void setPenerbit(String penerbit) {
        this.penerbit = penerbit;
    }

    public String getPenulis() {
        return penulis;
    }

    public void setPenulis(String penulis) {
        this.penulis = penulis;
    }

    public Buku getById(int id){
        Buku buku = new Buku();

        String query = "SELECT "
                +"b.idbuku AS idbuku, "
                +"b.judul AS judul, "
                +"b.penerbit AS penerbit, "
                +"b.penulis AS penulis, "
                +"k.idkategori AS idkategori, "
                +"k.nama AS nama, "
                +"k.keterangan AS keterangan "
                +"FROM buku b "
                +"LEFT JOIN kategori k ON b.idkategori = k.idkategori "
                +"WHERE b.idbuku = '" +id+ "'";
        ResultSet rs = DBHelper.selectQuery(query);

        try {
            while(rs.next()){
                buku = new Buku();
                buku.setIdbuku(rs.getInt("idbuku"));
                buku.getKategori().setIdKategori(rs.getInt("idkategori"));
                buku.getKategori().setNama(rs.getString("nama"));
                buku.getKategori().setKeterangan(rs.getString("keterangan"));
                buku.setJudul(rs.getString("judul"));
                buku.setPenerbit(rs.getString("penerbit"));
                buku.setPenulis(rs.getString("penulis"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return buku;
    }

```



```java
    public ArrayList<Buku> getAll(){
        ArrayList<Buku> ListBuku = new ArrayList();
        String query = "SELECT b.idbuku AS idbuku, b.judul AS judul, b.penerbit AS penerbit, b.penulis AS penulis, k.idkategori AS idkategori, k.nama AS nama, k.keterangan AS keterangan "
                + "FROM buku AS b LEFT JOIN kategori AS k ON b.idkategori = k.idkategori";
        ResultSet rs = DBHelper.selectQuery(query);


        try {
            while(rs.next()){
                Buku buku = new Buku();
                buku.setIdbuku(rs.getInt("idbuku"));
                buku.getKategori().setIdKategori(rs.getInt("idkategori"));
                buku.getKategori().setNama(rs.getString("nama"));
                buku.getKategori().setKeterangan(rs.getString("keterangan"));
                buku.setJudul(rs.getString("judul"));
                buku.setPenerbit(rs.getString("penerbit"));
                buku.setPenulis(rs.getString("penulis"));

                ListBuku.add(buku);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
//
        return ListBuku;
    }

     public ArrayList<Buku> search(String keyword){
        ArrayList<Buku> ListBuku = new ArrayList();


        String query = "SELECT b.idbuku AS idbuku, b.judul AS judul, b.penerbit AS penerbit, b.penulis AS penulis, k.idkategori AS idkategori, k.nama AS nama, k.keterangan AS keterangan "
                + "FROM buku AS b LEFT JOIN kategori AS k ON b.idkategori = k.idkategori "
                + "WHERE b.judul LIKE '%" + keyword + "%'"
                +"OR b.penerbit LIKE '%" + keyword + "%'"
                +"OR b.penulis LIKE '%" + keyword + "%'";
        ResultSet rs = DBHelper.selectQuery(query);

        try {
            while(rs.next()){
                Buku buku = new Buku();
                buku.setIdbuku(rs.getInt("idbuku"));
                buku.getKategori().setIdKategori(rs.getInt("idkategori"));
                buku.getKategori().setNama(rs.getString("nama"));
                buku.getKategori().setKeterangan(rs.getString("keterangan"));
                buku.setJudul(rs.getString("judul"));
                buku.setPenerbit(rs.getString("penerbit"));
                buku.setPenulis(rs.getString("penulis"));
```



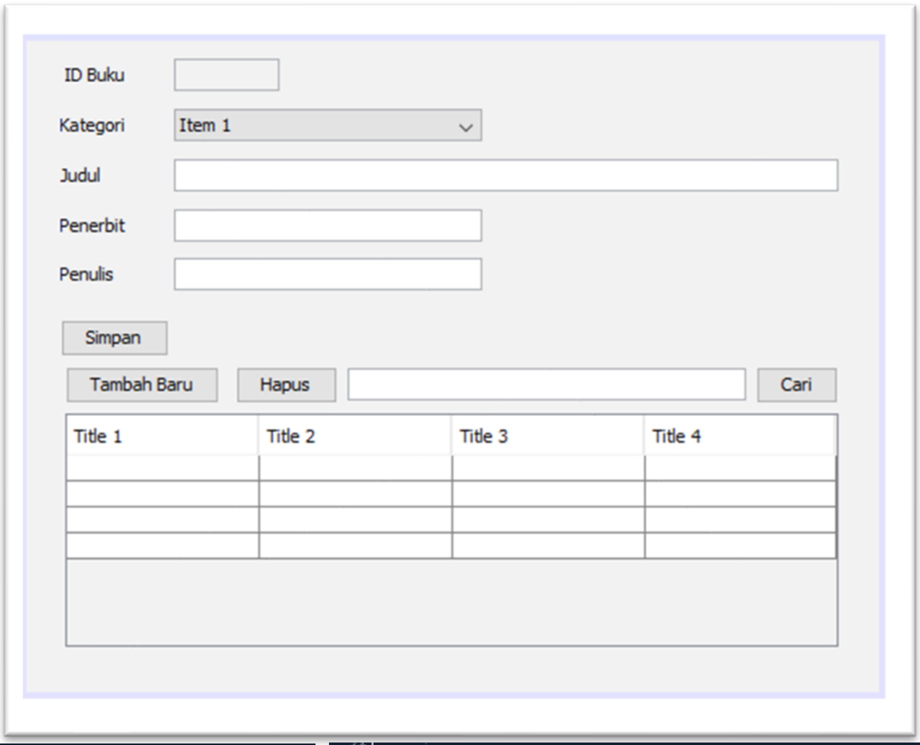
```java
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        return ListBuku;
    }

    public void save(){
        if(getById(idbuku).getIdbuku() == 0){
            String SQL = "INSERT INTO buku (idkategori, judul, penerbit, penulis) VALUES("
                    + "'" + this.getKategori().getIdKategori() + "',"
                    + "'" + this.judul + "',"
                    + "'" + this.penerbit + "',"
                    +"'" + this.penulis + "'"
                    + ")";

            this.idbuku = DBHelper.insertQueryGetId(SQL);
        } else {
            String SQL = "UPDATE buku SET "
                     +"idkategori = '" + this.getKategori().getIdKategori() + "',"
                     +"judul = '" + this.judul + "',"
                     +"penerbit = '" + this.penerbit + "',"
                     +"penulis = '" + this.penulis + "'"
                     +"WHERE idbuku = '" + this.idbuku + "'";
           DBHelper.executeQuery(SQL);
        }
    }

    public void delete(){
        String SQL = "DELETE FROM buku WHERE idbuku = '" + this.idbuku + "'";
        DBHelper.executeQuery(SQL);
    }
}
```

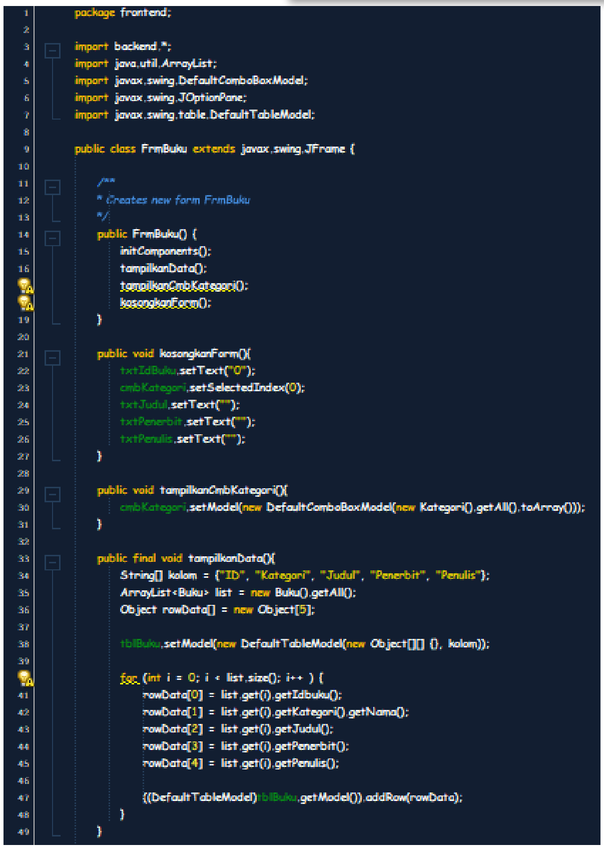


```
run:
Kategori: Novel, Judul: Timun Mas
Kategori: Reference, Judul: Aljabar Linier
Kategori: Novel, Judul: Timun Mas
BUILD SUCCESSFUL (total time: 1 second)
```

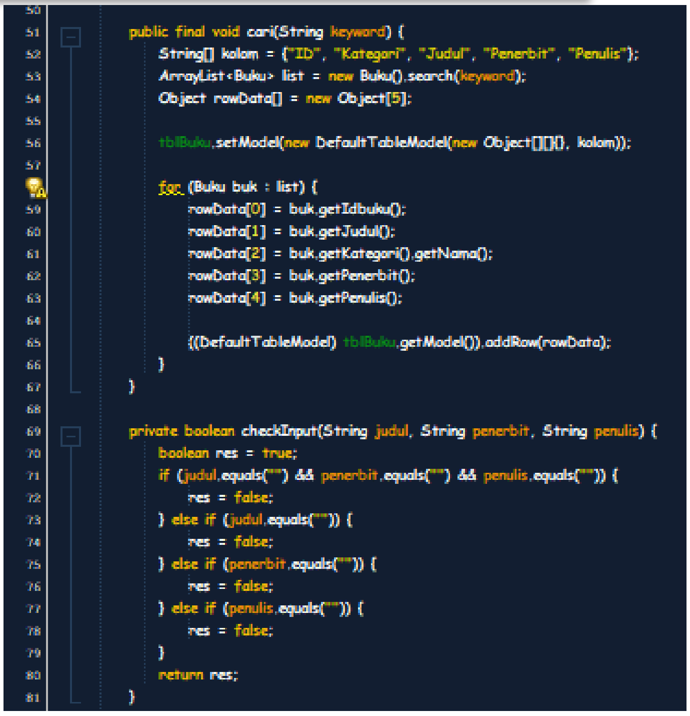
Experiment 8

Creating a GUI for Data Book, which is equipped with a combo box to select the Kategori that is connected with the table Kategori.

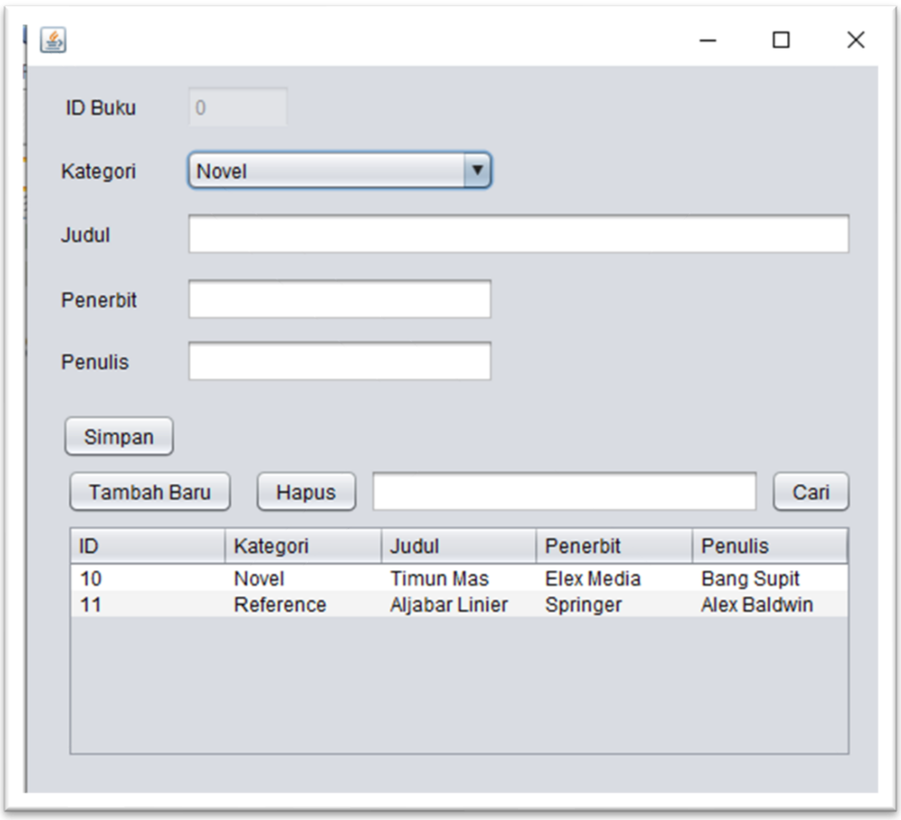




```java
package frontend;

import backend.*;
import java.util.ArrayList;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class FrmBuku extends javax.swing.JFrame {

    /**
    * Creates new form FrmBuku
    */
    public FrmBuku() {
        initComponents();
        tampilkanData();
        tampilkanCmbKategori();
        kosongkanForm();
    }

    public void kosongkanForm(){
        txtIdBuku.setText("0");
        cmbKategori.setSelectedIndex(0);
        txtJudul.setText("");
        txtPenerbit.setText("");
        txtPenulis.setText("");
    }

    public void tampilkanCmbKategori(){
        cmbKategori.setModel(new DefaultComboBoxModel(new Kategori().getAll().toArray()));
    }

    public final void tampilkanData(){
        String[] kolom = {"ID", "Kategori", "Judul", "Penerbit", "Penulis"};
        ArrayList<Buku> list = new Buku().getAll();
        Object rowData[] = new Object[5];

        tblBuku.setModel(new DefaultTableModel(new Object[][] {}, kolom));

        for (int i = 0; i < list.size(); i++ ) {
            rowData[0] = list.get(i).getIdbuku();
            rowData[1] = list.get(i).getKategori().getNama();
            rowData[2] = list.get(i).getJudul();
            rowData[3] = list.get(i).getPenerbit();
            rowData[4] = list.get(i).getPenulis();

            ((DefaultTableModel)tblBuku.getModel()).addRow(rowData);
        }
    }
```



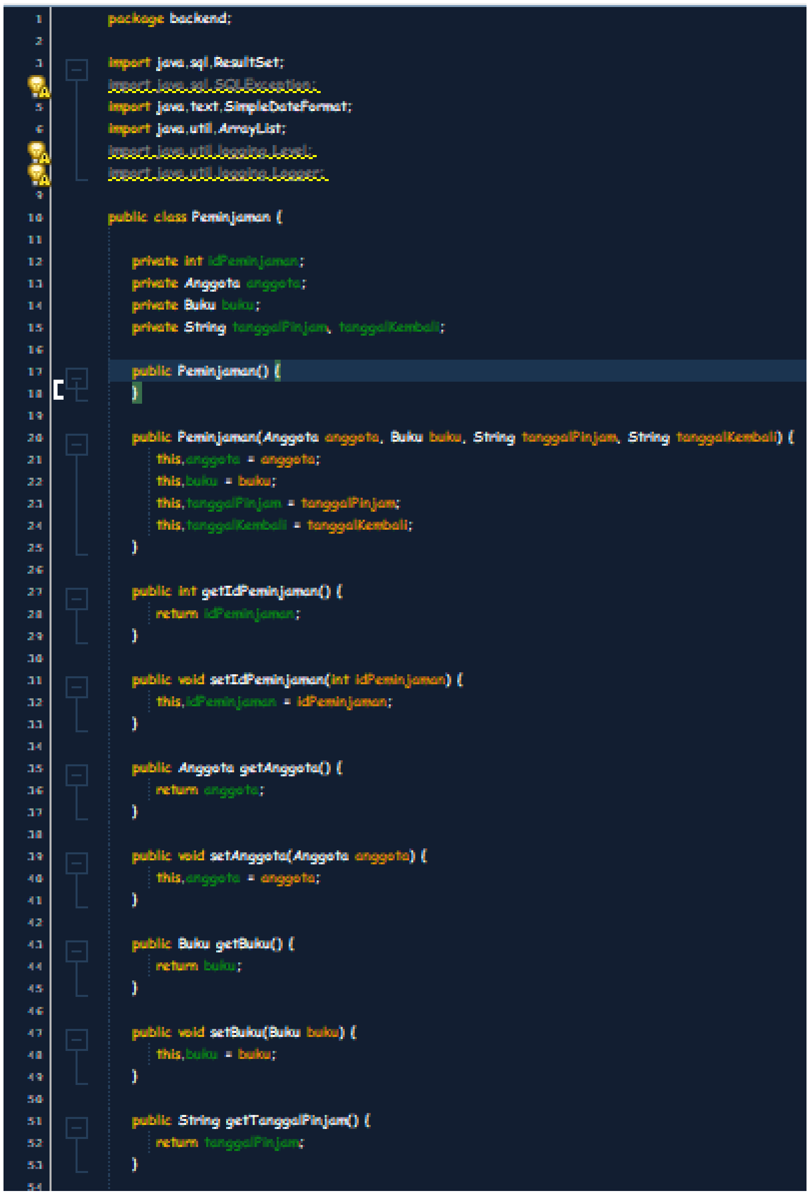
```java

    public final void cari(String keyword) {
        String[] kolom = {"ID", "Kategori", "Judul", "Penerbit", "Penulis"};
        ArrayList<Buku> list = new Buku().search(keyword);
        Object rowData[] = new Object[5];

        tblBuku.setModel(new DefaultTableModel(new Object[][]{}, kolom));

        for (Buku buk : list) {
            rowData[0] = buk.getIdbuku();
            rowData[1] = buk.getJudul();
            rowData[2] = buk.getKategori().getNama();
            rowData[3] = buk.getPenerbit();
            rowData[4] = buk.getPenulis();

            ((DefaultTableModel) tblBuku.getModel()).addRow(rowData);
        }
    }

    private boolean checkInput(String judul, String penerbit, String penulis) {
        boolean res = true;
        if (judul.equals("") && penerbit.equals("") && penulis.equals("")) {
            res = false;
        } else if (judul.equals("")) {
            res = false;
        } else if (penerbit.equals("")) {
            res = false;
        } else if (penulis.equals("")) {
            res = false;
        }
        return res;
    }
```



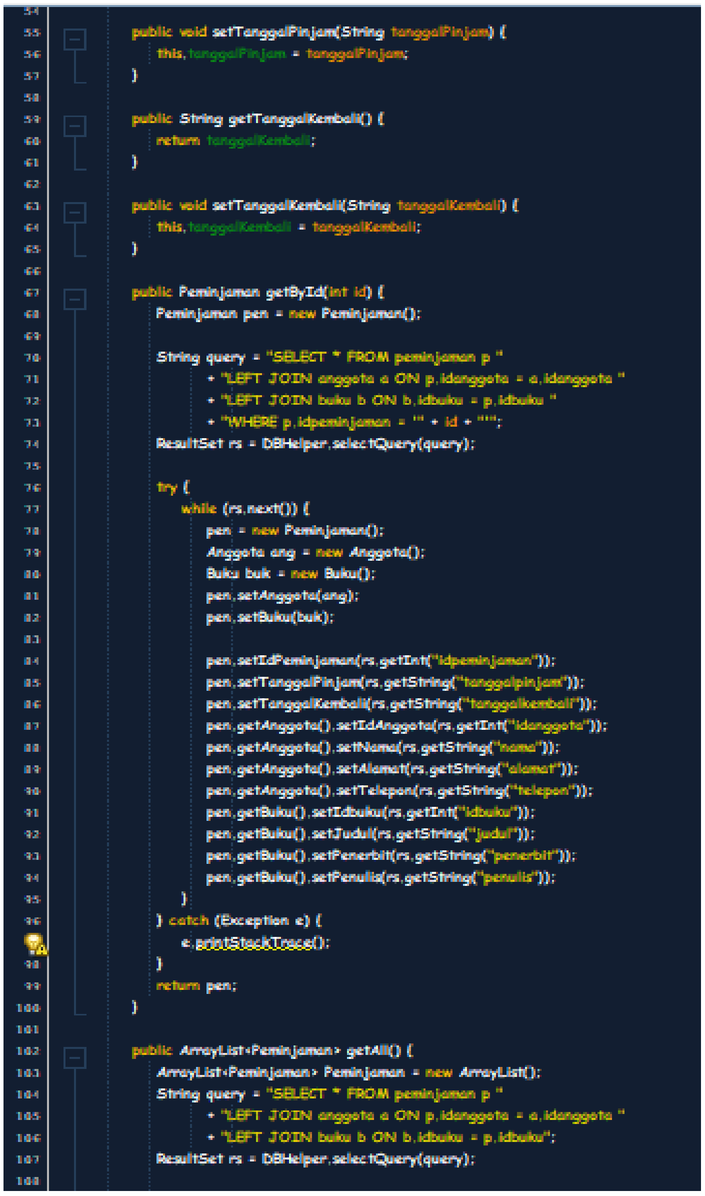
ID	Kategori	Judul	Penerbit	Penulis
10	Novel	Timun Mas	Elex Media	Bang Supit
11	Reference	Aljabar Linier	Springer	Alex Baldwin

Assignment

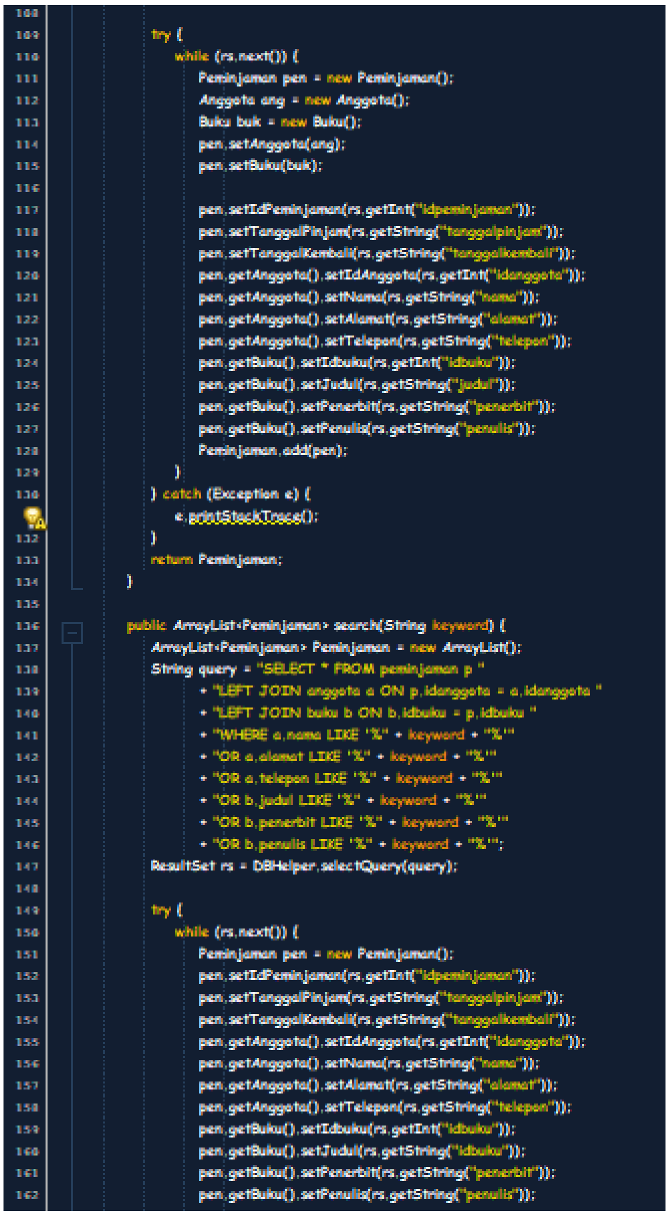
1. **Make Peminjaman class.**



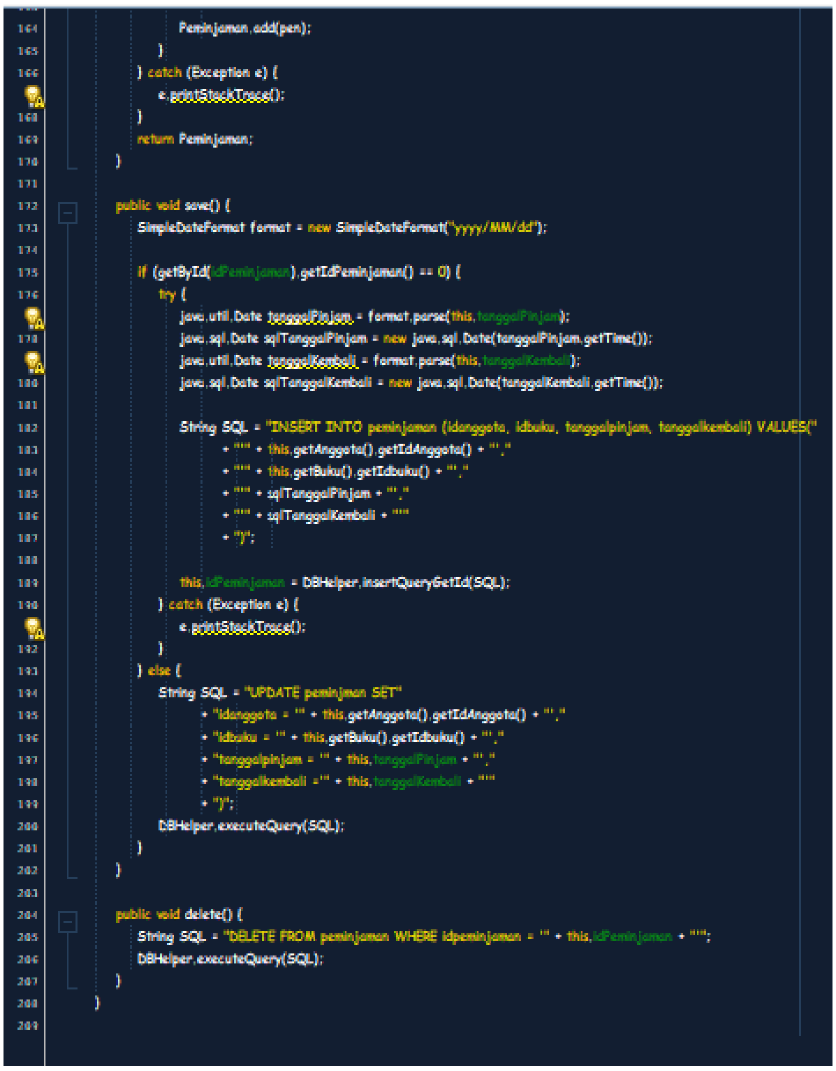
```java
package backend;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Peminjaman {

    private int idPeminjaman;
    private Anggota anggota;
    private Buku buku;
    private String tanggalPinjam, tanggalKembali;

    public Peminjaman() {
    }

    public Peminjaman(Anggota anggota, Buku buku, String tanggalPinjam, String tanggalKembali) {
        this.anggota = anggota;
        this.buku = buku;
        this.tanggalPinjam = tanggalPinjam;
        this.tanggalKembali = tanggalKembali;
    }

    public int getIdPeminjaman() {
        return idPeminjaman;
    }

    public void setIdPeminjaman(int idPeminjaman) {
        this.idPeminjaman = idPeminjaman;
    }

    public Anggota getAnggota() {
        return anggota;
    }

    public void setAnggota(Anggota anggota) {
        this.anggota = anggota;
    }

    public Buku getBuku() {
        return buku;
    }

    public void setBuku(Buku buku) {
        this.buku = buku;
    }

    public String getTanggalPinjam() {
        return tanggalPinjam;
    }

```



```java

    public void setTanggalPinjam(String tanggalPinjam) {
        this.tanggalPinjam = tanggalPinjam;
    }

    public String getTanggalKembali() {
        return tanggalKembali;
    }

    public void setTanggalKembali(String tanggalKembali) {
        this.tanggalKembali = tanggalKembali;
    }

    public Peminjaman getById(int id) {
        Peminjaman pen = new Peminjaman();

        String query = "SELECT * FROM peminjaman p "
                + "LEFT JOIN anggota a ON p.idanggota = a.idanggota "
                + "LEFT JOIN buku b ON b.idbuku = p.idbuku "
                + "WHERE p.idpeminjaman = '" + id + "'";
        ResultSet rs = DBHelper.selectQuery(query);

        try {
            while (rs.next()) {
                pen = new Peminjaman();
                Anggota ang = new Anggota();
                Buku buk = new Buku();
                pen.setAnggota(ang);
                pen.setBuku(buk);

                pen.setIdPeminjaman(rs.getInt("idpeminjaman"));
                pen.setTanggalPinjam(rs.getString("tanggalpinjam"));
                pen.setTanggalKembali(rs.getString("tanggalkembali"));
                pen.getAnggota().setIdAnggota(rs.getInt("idanggota"));
                pen.getAnggota().setNama(rs.getString("nama"));
                pen.getAnggota().setAlamat(rs.getString("alamat"));
                pen.getAnggota().setTelepon(rs.getString("telepon"));
                pen.getBuku().setIdbuku(rs.getInt("idbuku"));
                pen.getBuku().setJudul(rs.getString("judul"));
                pen.getBuku().setPenerbit(rs.getString("penerbit"));
                pen.getBuku().setPenulis(rs.getString("penulis"));
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return pen;
    }

    public ArrayList<Peminjaman> getAll() {
        ArrayList<Peminjaman> Peminjaman = new ArrayList();
        String query = "SELECT * FROM peminjaman p "
                + "LEFT JOIN anggota a ON p.idanggota = a.idanggota "
                + "LEFT JOIN buku b ON b.idbuku = p.idbuku";
        ResultSet rs = DBHelper.selectQuery(query);

```

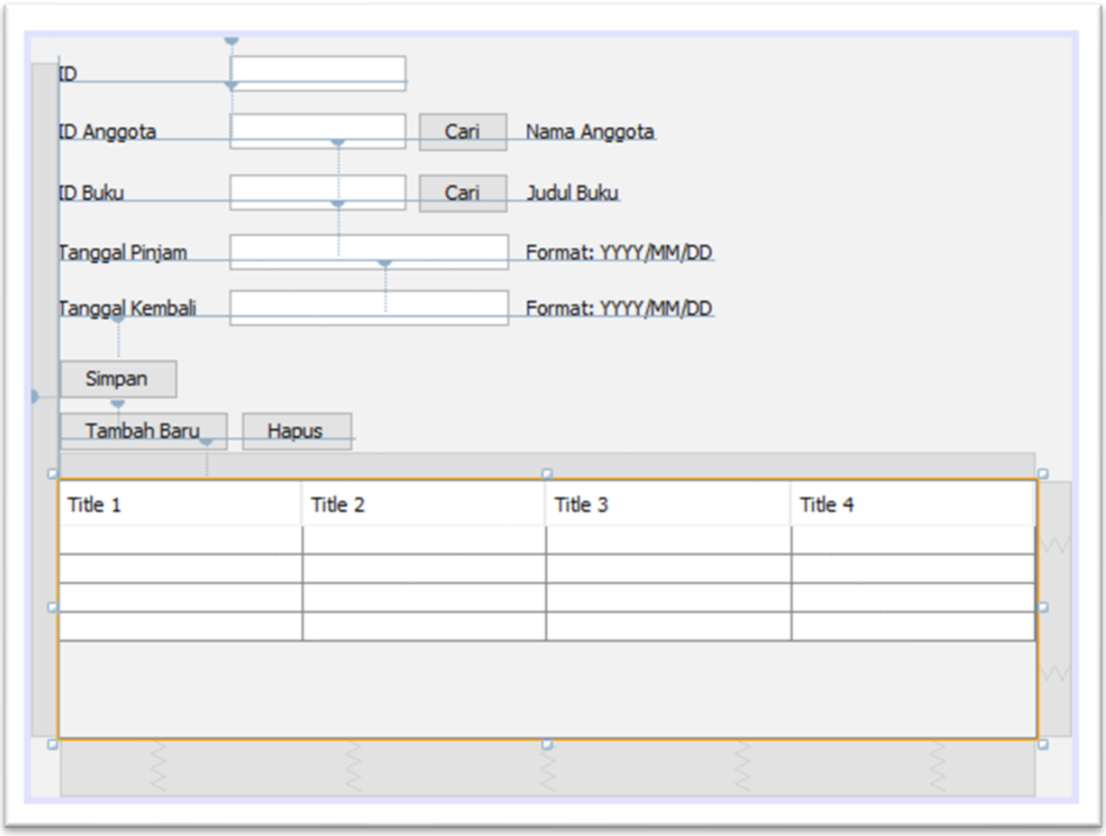


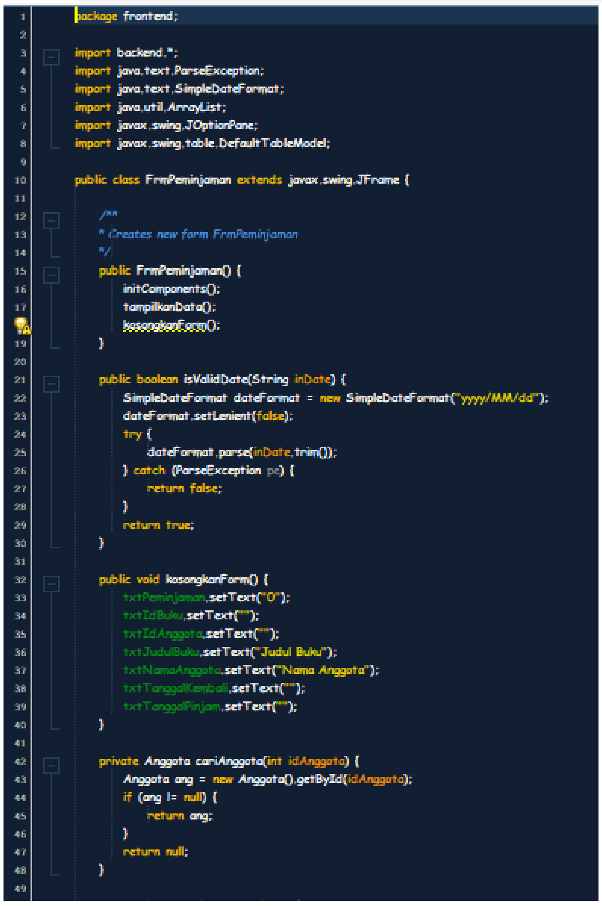
```java

        try {
            while (rs.next()) {
                Peminjaman pen = new Peminjaman();
                Anggota ang = new Anggota();
                Buku buk = new Buku();
                pen.setAnggota(ang);
                pen.setBuku(buk);

                pen.setIdPeminjaman(rs.getInt("idpeminjaman"));
                pen.setTanggalPinjam(rs.getString("tanggalpinjam"));
                pen.setTanggalKembali(rs.getString("tanggalkembali"));
                pen.getAnggota().setIdAnggota(rs.getInt("idanggota"));
                pen.getAnggota().setNama(rs.getString("nama"));
                pen.getAnggota().setAlamat(rs.getString("alamat"));
                pen.getAnggota().setTelepon(rs.getString("telepon"));
                pen.getBuku().setIdbuku(rs.getInt("idbuku"));
                pen.getBuku().setJudul(rs.getString("judul"));
                pen.getBuku().setPenerbit(rs.getString("penerbit"));
                pen.getBuku().setPenulis(rs.getString("penulis"));
                Peminjaman.add(pen);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return Peminjaman;
    }

    public ArrayList<Peminjaman> search(String keyword) {
        ArrayList<Peminjaman> Peminjaman = new ArrayList();
        String query = "SELECT * FROM peminjaman p "
                + "LEFT JOIN anggota a ON p.idanggota = a.idanggota "
                + "LEFT JOIN buku b ON b.idbuku = p.idbuku "
                + "WHERE a.nama LIKE '%" + keyword + "%'"
                + "OR a.alamat LIKE '%" + keyword + "%'"
                + "OR a.telepon LIKE '%" + keyword + "%'"
                + "OR b.judul LIKE '%" + keyword + "%'"
                + "OR b.penerbit LIKE '%" + keyword + "%'"
                + "OR b.penulis LIKE '%" + keyword + "%'";
        ResultSet rs = DBHelper.selectQuery(query);

        try {
            while (rs.next()) {
                Peminjaman pen = new Peminjaman();
                pen.setIdPeminjaman(rs.getInt("idpeminjaman"));
                pen.setTanggalPinjam(rs.getString("tanggalpinjam"));
                pen.setTanggalKembali(rs.getString("tanggalkembali"));
                pen.getAnggota().setIdAnggota(rs.getInt("idanggota"));
                pen.getAnggota().setNama(rs.getString("nama"));
                pen.getAnggota().setAlamat(rs.getString("alamat"));
                pen.getAnggota().setTelepon(rs.getString("telepon"));
                pen.getBuku().setIdbuku(rs.getInt("idbuku"));
                pen.getBuku().setJudul(rs.getString("idbuku"));
                pen.getBuku().setPenerbit(rs.getString("penerbit"));
                pen.getBuku().setPenulis(rs.getString("penulis"));
```



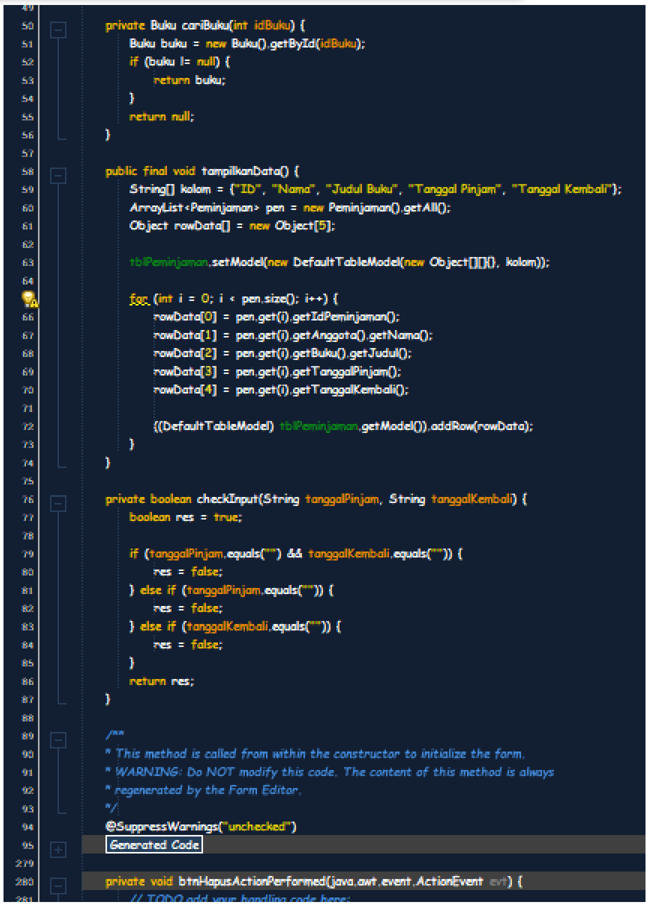
```java
                Peminjaman.add(pen);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return Peminjaman;
    }

    public void save() {
        SimpleDateFormat format = new SimpleDateFormat("yyyy/MM/dd");

        if (getById(idPeminjaman).getIdPeminjaman() == 0) {
            try {
                java.util.Date tanggalPinjam = format.parse(this.tanggalPinjam);
                java.sql.Date sqlTanggalPinjam = new java.sql.Date(tanggalPinjam.getTime());
                java.util.Date tanggalKembali = format.parse(this.tanggalKembali);
                java.sql.Date sqlTanggalKembali = new java.sql.Date(tanggalKembali.getTime());

                String SQL = "INSERT INTO peminjaman (idanggota, idbuku, tanggalpinjam, tanggalkembali) VALUES("
                        + "'" + this.getAnggota().getIdAnggota() + "',"
                        + "'" + this.getBuku().getIdbuku() + "',"
                        + "'" + sqlTanggalPinjam + "',"
                        + "'" + sqlTanggalKembali + "'"
                        + ")";

                this.idPeminjaman = DBHelper.insertQueryGetId(SQL);
            } catch (Exception e) {
                e.printStackTrace();
            }
        } else {
            String SQL = "UPDATE peminjman SET"
                    + "idanggota = '" + this.getAnggota().getIdAnggota() + "',"
                    + "idbuku = '" + this.getBuku().getIdbuku() + "',"
                    + "tanggalpinjam = '" + this.tanggalPinjam + "',"
                    + "tanggalkembali ='" + this.tanggalKembali + "'"
                    + ")";
            DBHelper.executeQuery(SQL);
        }
    }

    public void delete() {
        String SQL = "DELETE FROM peminjaman WHERE idpeminjaman = '" + this.idPeminjaman + "'";
        DBHelper.executeQuery(SQL);
    }
}

```

2. **Make FrmPeminjaman**

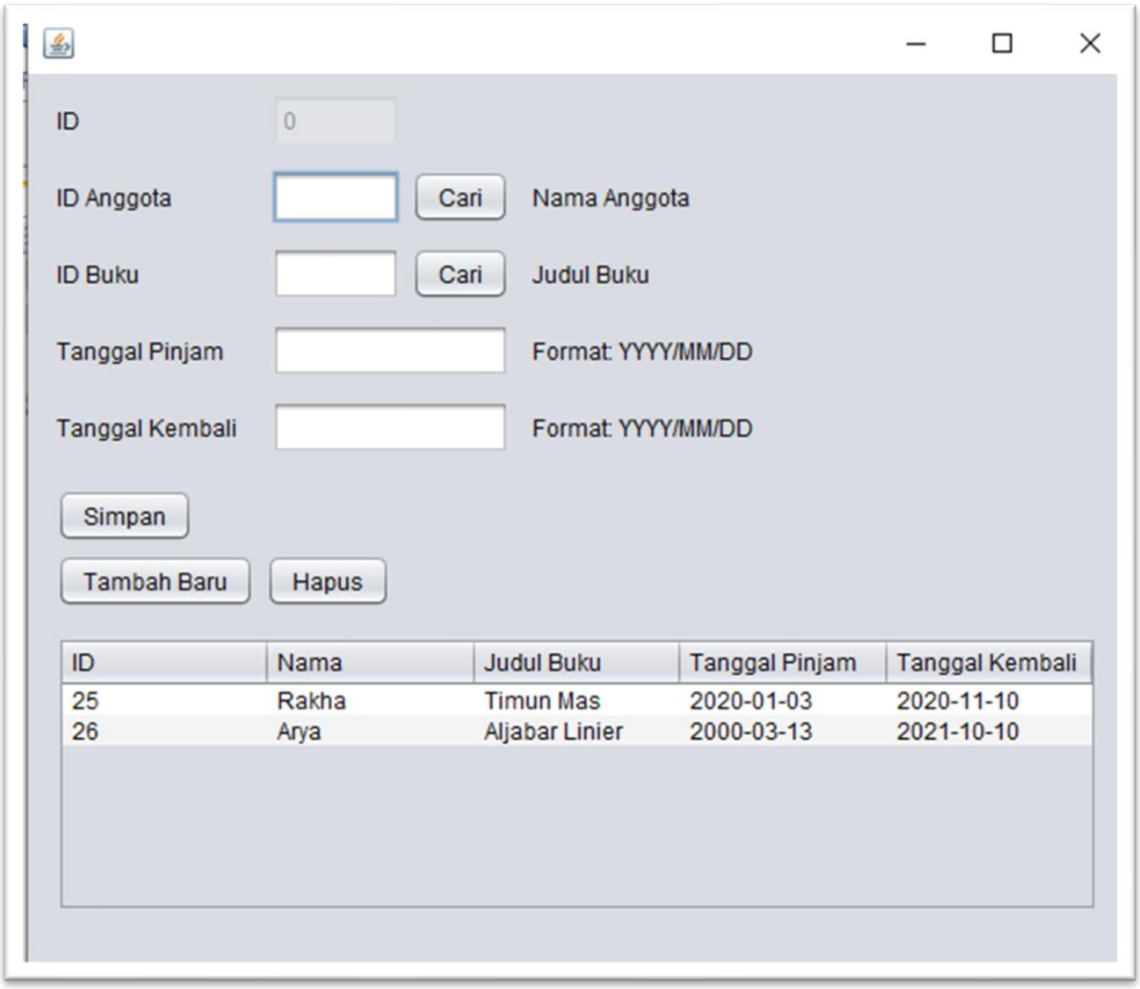




```java
package frontend;

import backend.*;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.ArrayList;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class FrmPeminjaman extends javax.swing.JFrame {

    /**
    * Creates new form FrmPeminjaman
    */
    public FrmPeminjaman() {
        initComponents();
        tampilkanData();
        kosongkanForm();
    }

    public boolean isValidDate(String inDate) {
        SimpleDateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd");
        dateFormat.setLenient(false);
        try {
            dateFormat.parse(inDate.trim());
        } catch (ParseException pe) {
            return false;
        }
        return true;
    }

    public void kosongkanForm() {
        txtPeminjaman.setText("0");
        txtIdBuku.setText("");
        txtIdAnggota.setText("");
        txtJudulBuku.setText("Judul Buku");
        txtNamaAnggota.setText("Nama Anggota");
        txtTanggalKembali.setText("");
        txtTanggalPinjam.setText("");
    }

    private Anggota cariAnggota(int idAnggota) {
        Anggota ang = new Anggota().getById(idAnggota);
        if (ang != null) {
            return ang;
        }
        return null;
    }

```



```java
    private Buku cariBuku(int idBuku) {
        Buku buku = new Buku().getById(idBuku);
        if (buku != null) {
            return buku;
        }
        return null;
    }

    public final void tampilkanData() {
        String[] kolom = {"ID", "Nama", "Judul Buku", "Tanggal Pinjam", "Tanggal Kembali"};
        ArrayList<Peminjaman> pen = new Peminjaman().getAll();
        Object rowData[] = new Object[5];

        tblPeminjaman.setModel(new DefaultTableModel(new Object[][]{}, kolom));

        for (int i = 0; i < pen.size(); i++) {
            rowData[0] = pen.get(i).getIdPeminjaman();
            rowData[1] = pen.get(i).getAnggota().getNama();
            rowData[2] = pen.get(i).getBuku().getJudul();
            rowData[3] = pen.get(i).getTanggalPinjam();
            rowData[4] = pen.get(i).getTanggalKembali();

            ((DefaultTableModel) tblPeminjaman.getModel()).addRow(rowData);
        }
    }

    private boolean checkInput(String tanggalPinjam, String tanggalKembali) {
        boolean res = true;

        if (tanggalPinjam.equals("") && tanggalKembali.equals("")) {
            res = false;
        } else if (tanggalPinjam.equals("")) {
            res = false;
        } else if (tanggalKembali.equals("")) {
            res = false;
        }
        return res;
    }

    /**
    * This method is called from within the constructor to initialize the form.
    * WARNING: Do NOT modify this code. The content of this method is always
    * regenerated by the Form Editor.
    */
    @SuppressWarnings("unchecked")
    Generated Code

    private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
```



ID	Nama	Judul Buku	Tanggal Pinjam	Tanggal Kembali
25	Rakha	Timun Mas	2020-01-03	2020-11-10
26	Arya	Aljabar Linier	2000-03-13	2021-10-10