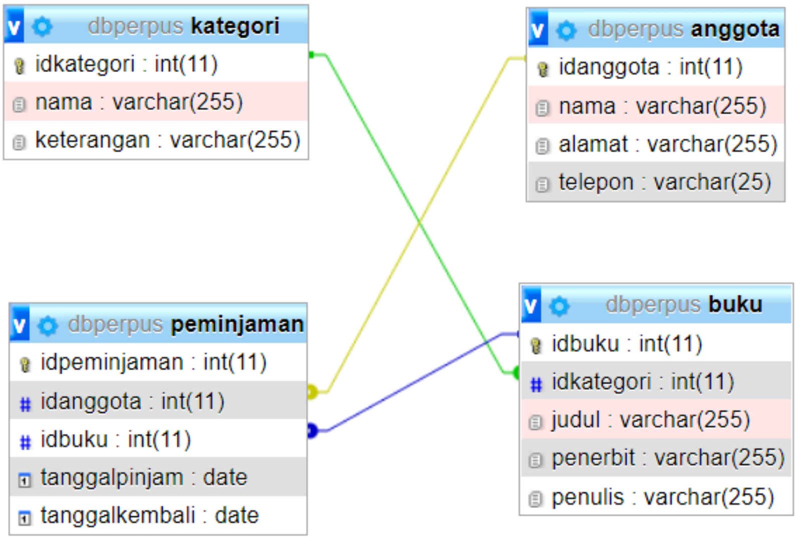
NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

## 3.1 Percobaan 1



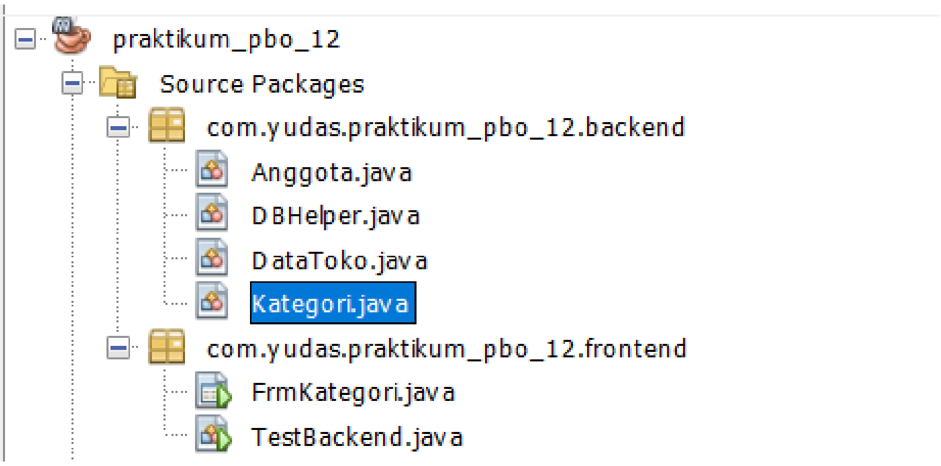
dbperpus kategori
idkategori : int(11)
nama : varchar(255)
keterangan : varchar(255)

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

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

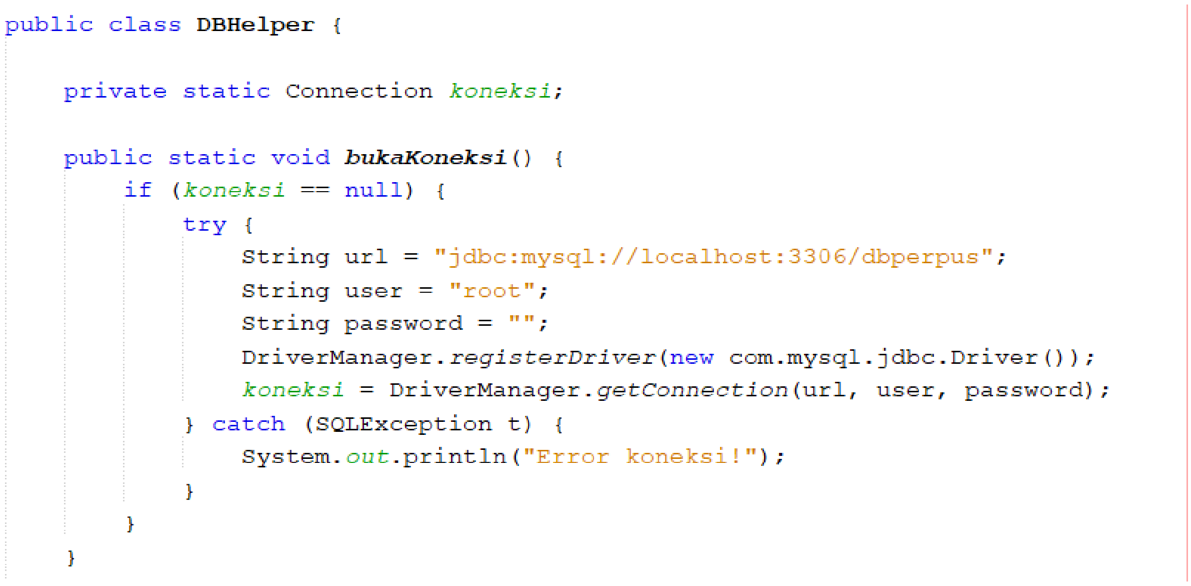
## 3.2 Percobaan 2



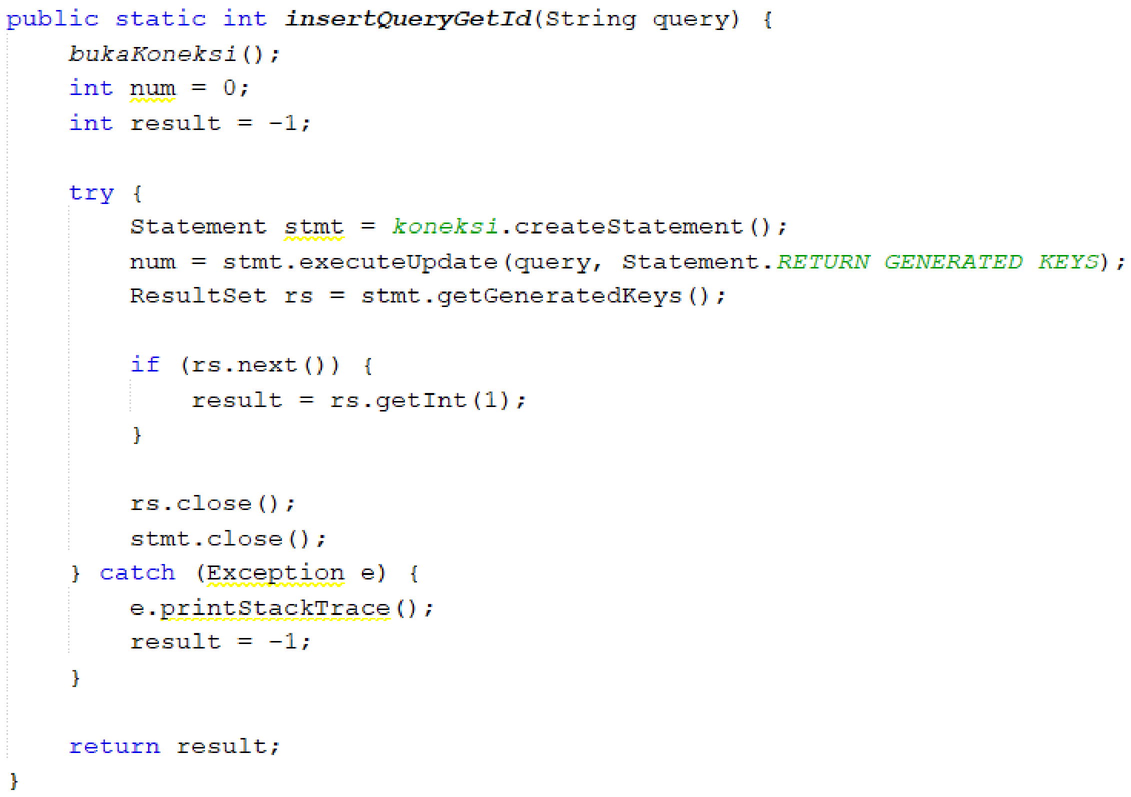
```
praktikum_pbo_12
  Source Packages
    com.yudas.praktikum_pbo_12.backend
      Anggota.java
      DBHelper.java
      DataToko.java
      Kategori.java
    com.yudas.praktikum_pbo_12.frontend
      FrmKategori.java
      TestBackend.java
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

## 3.3 Percobaan 3

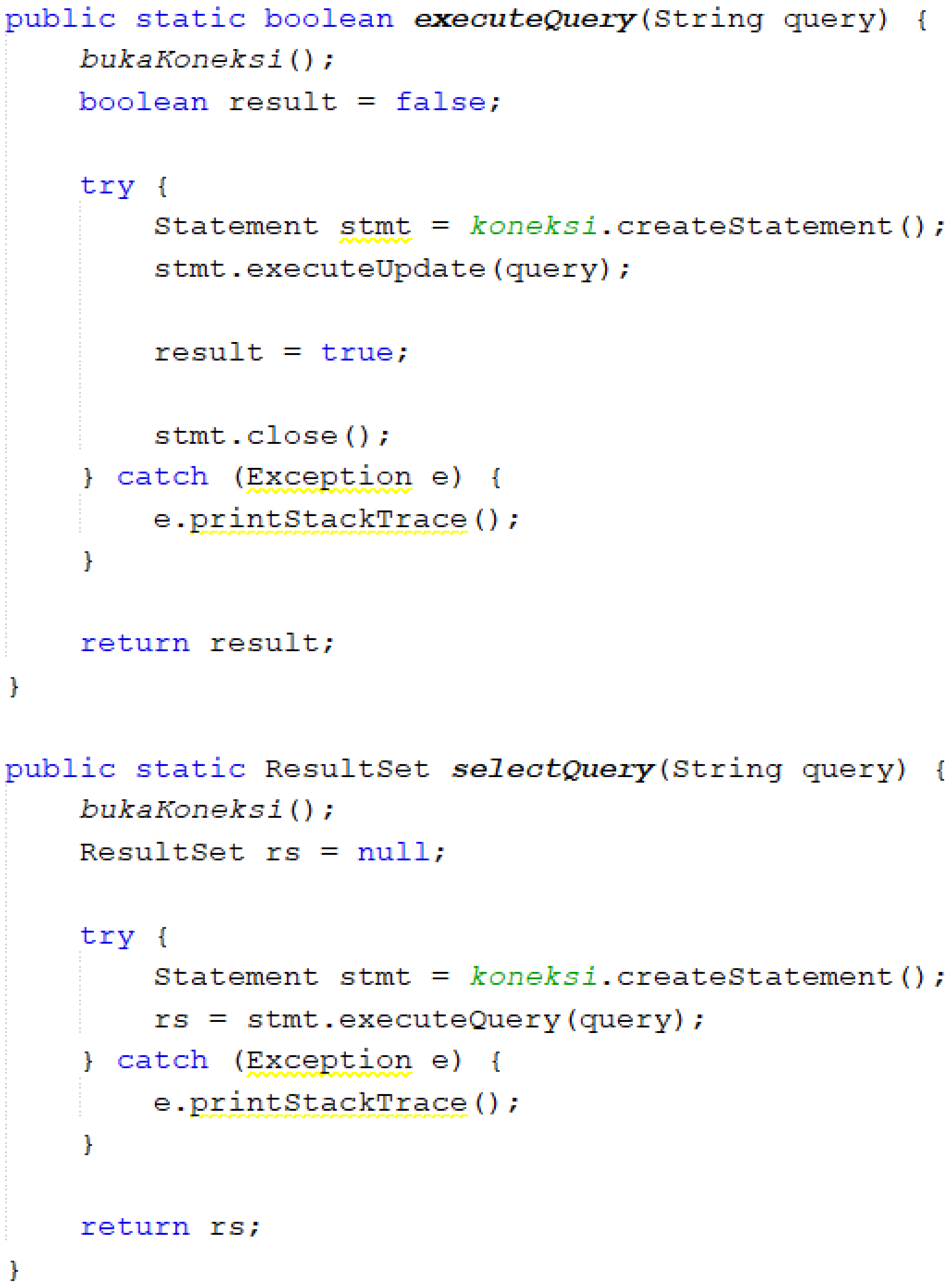


```java
public class DBHelper {

    private static Connection koneksi;

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



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

    try {
        Statement stmt = koneksi.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;
}
```

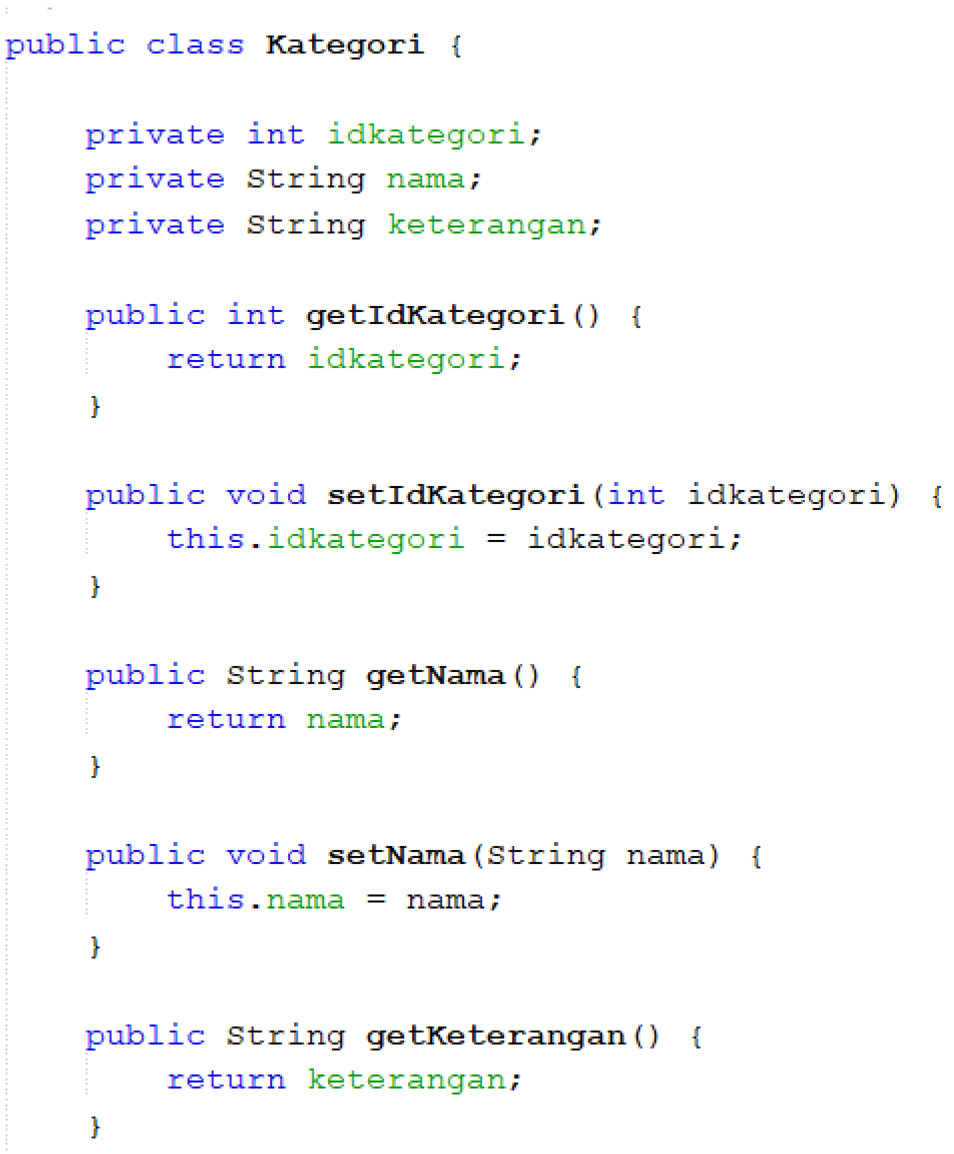
NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO



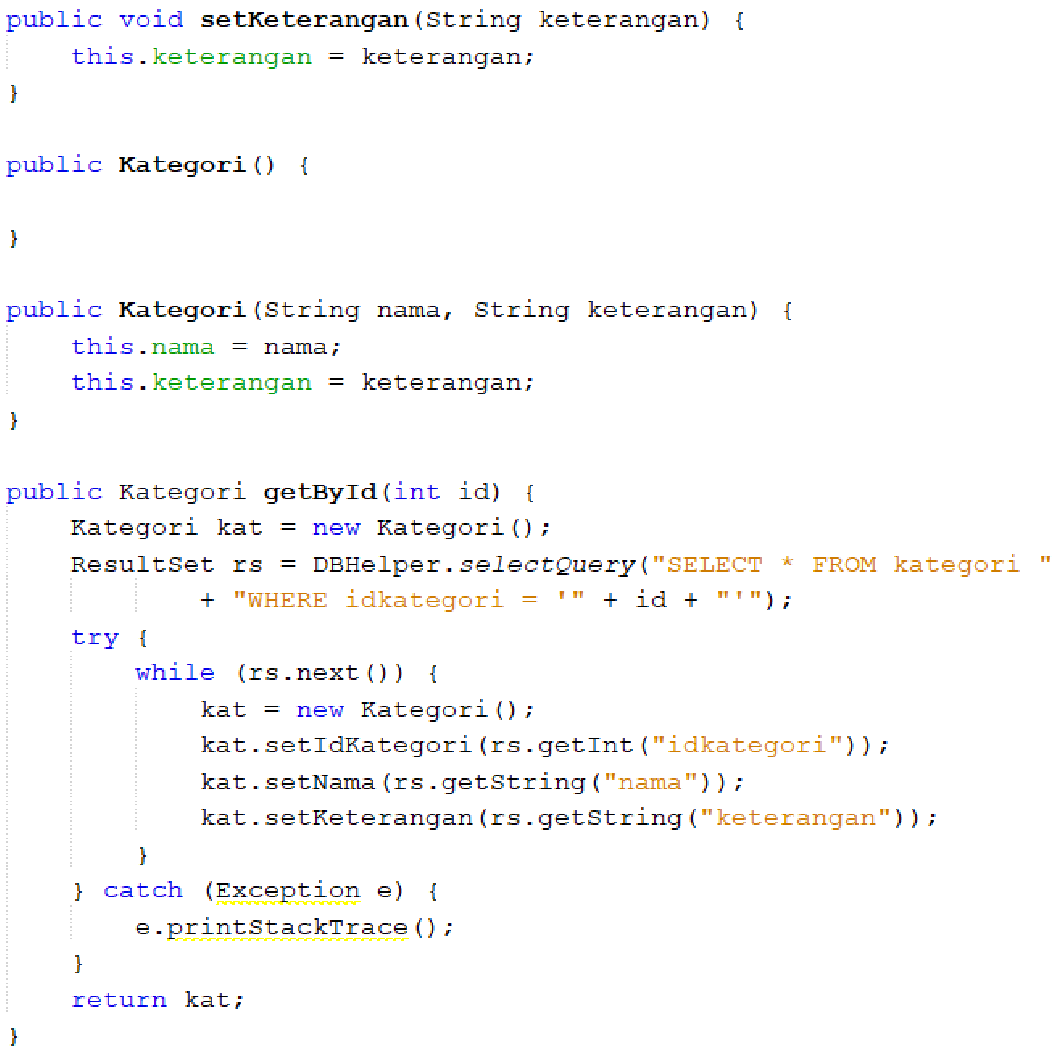
```java
public static boolean executeQuery(String query) {
    bukaKoneksi();
    boolean result = false;

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

    return rs;
}
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

## Percobaan 3.4

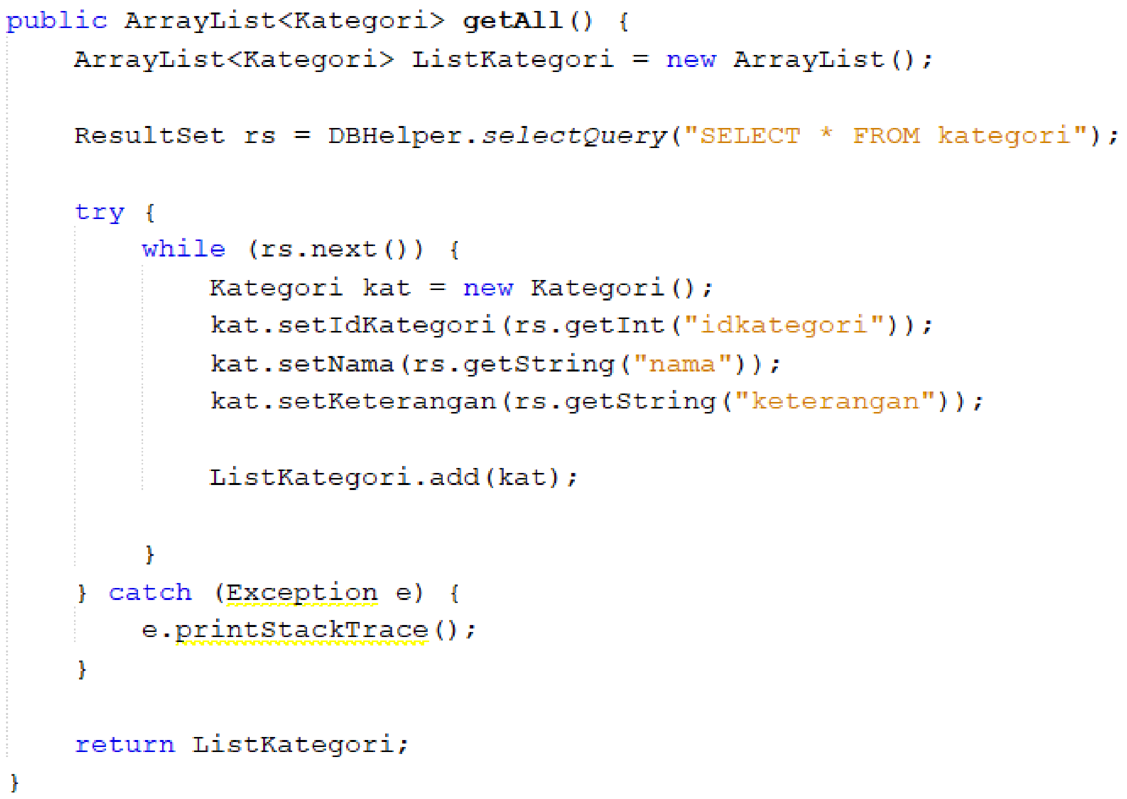


```java
public class Kategori {

    private int idkategori;
    private String nama;
    private String 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;
    }
```

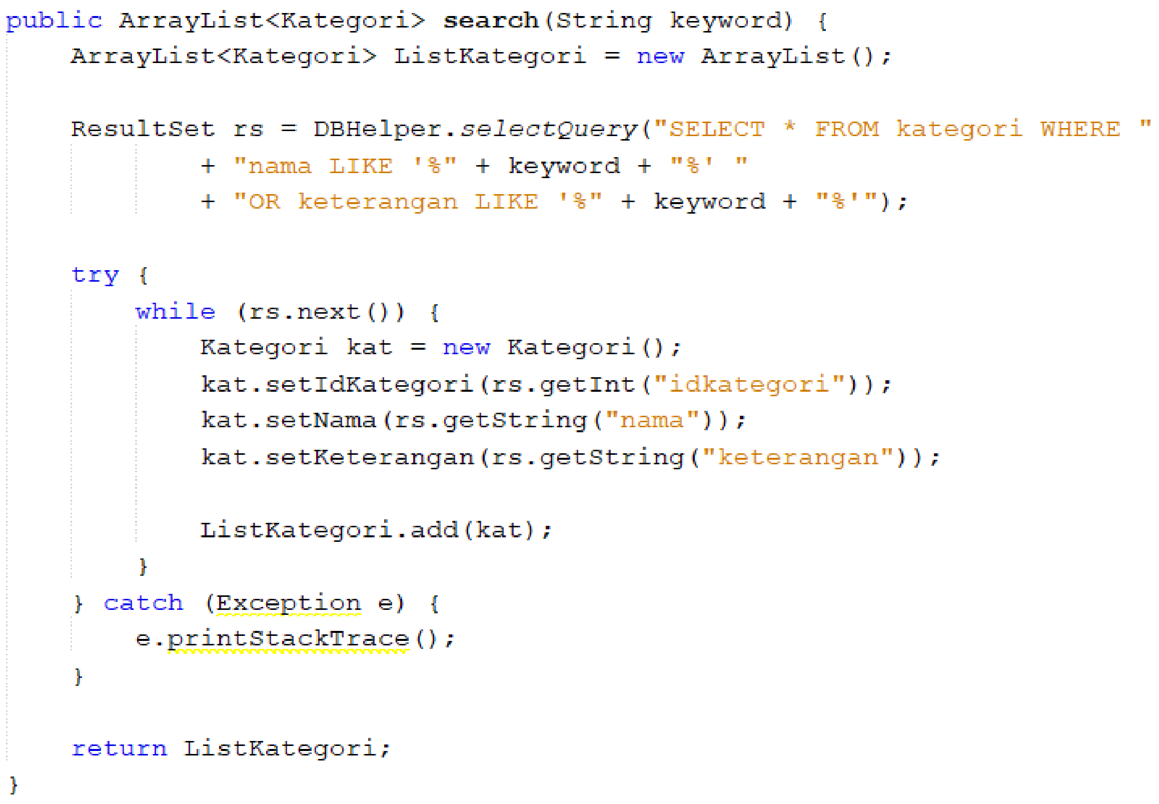


```
public void setKeterangan(String keterangan) {
    this.keterangan = keterangan;
}

public Kategori() {

}

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

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;
}
```

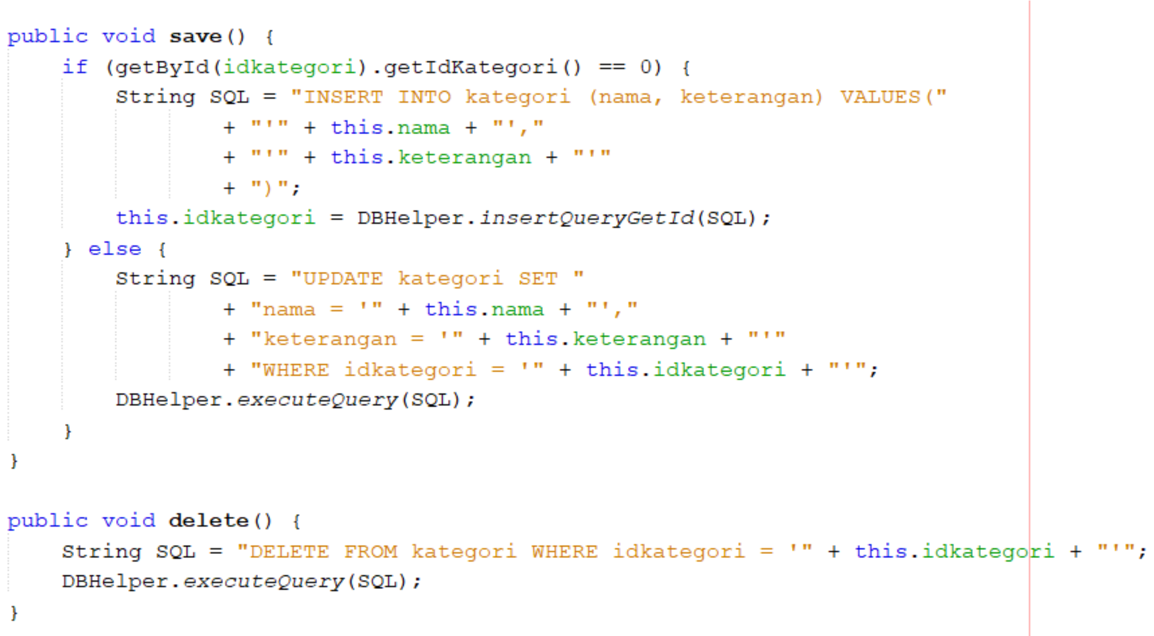
NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO



```java
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;
}
```



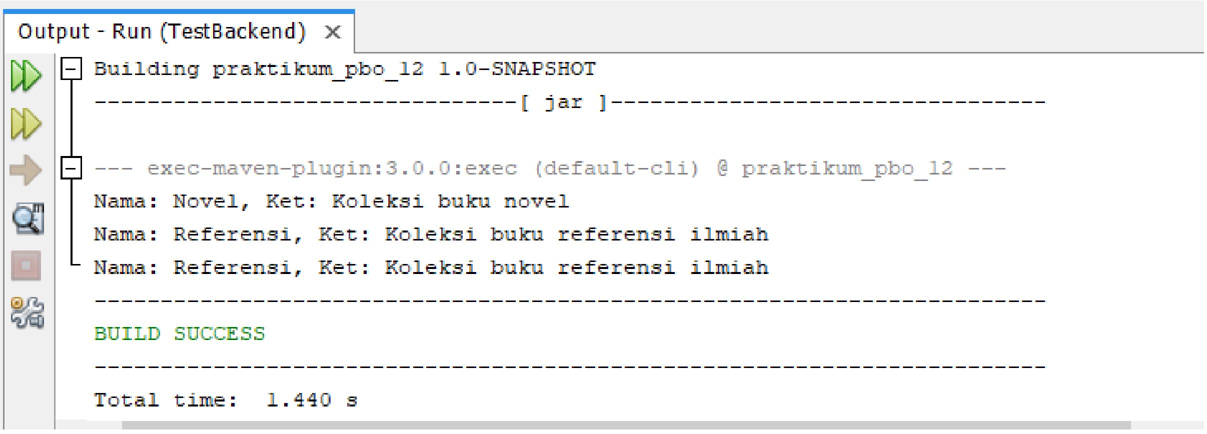
```java
public ArrayList<Kategori> search(String keyword) {
    ArrayList<Kategori> ListKategori = new ArrayList();

    ResultSet rs = DBHelper.selectQuery("SELECT * FROM kategori WHERE "
            + "nama LIKE '%" + keyword + "%' "
            + "OR keterangan LIKE '%" + keyword + "%'");

    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;
}
```



```java
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);
}
```

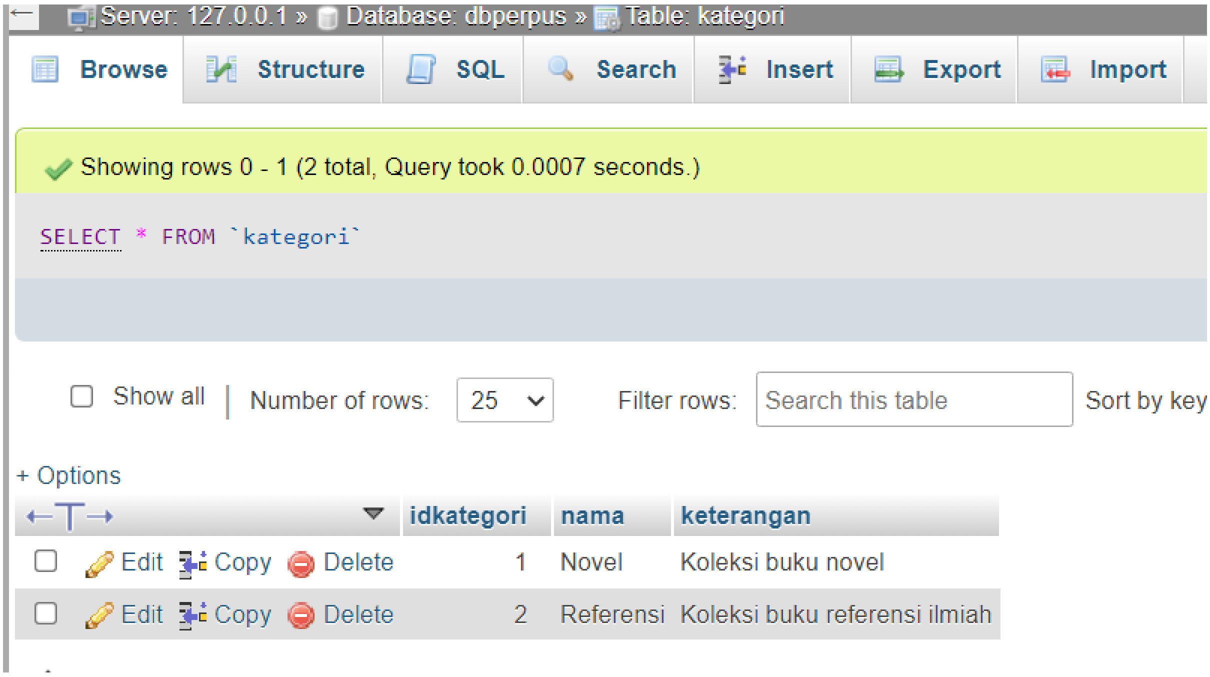
## Percobaan 3.5

Output :



```
Building praktikum_pbo_12 1.0-SNAPSHOT
--------------------------------[ jar ]---------------------------------

--- exec-maven-plugin:3.0.0:exec (default-cli) @ praktikum_pbo_12 ---
Nama: Novel, Ket: Koleksi buku novel
Nama: Referensi, Ket: Koleksi buku referensi ilmiah
Nama: Referensi, Ket: Koleksi buku referensi ilmiah
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  1.440 s
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO



✓ Showing rows 0 - 1 (2 total, Query took 0.0007 seconds.)

```
SELECT * FROM `kategori`
```

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key

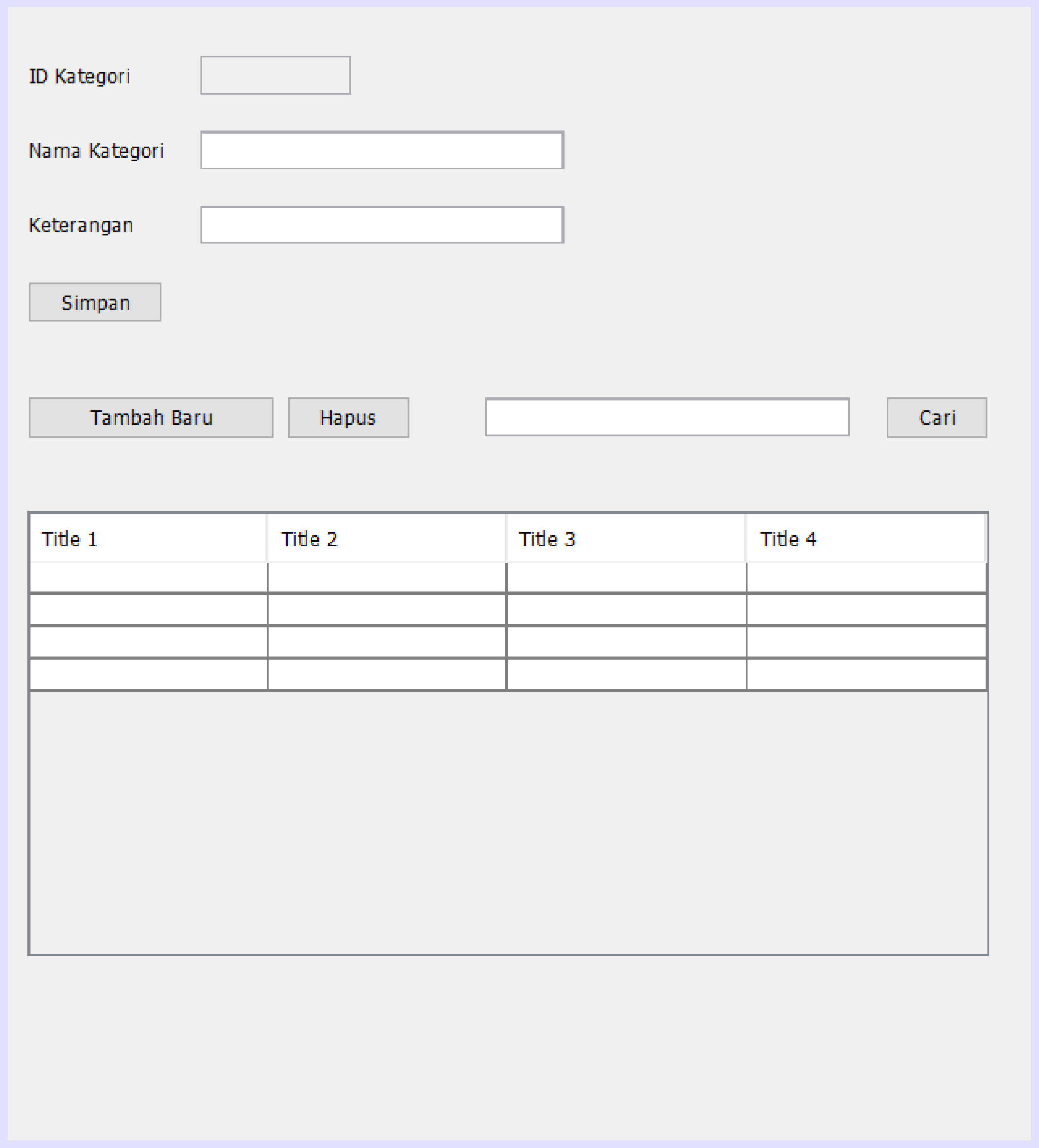
+ Options

| | idkategori | nama | keterangan |
|---|---|---|---|
| □ Edit Copy Delete | 1 | Novel | Koleksi buku novel |
| □ Edit Copy Delete | 2 | Referensi | Koleksi buku referensi ilmiah |

*1 Data akan berhasil masuk ke database*

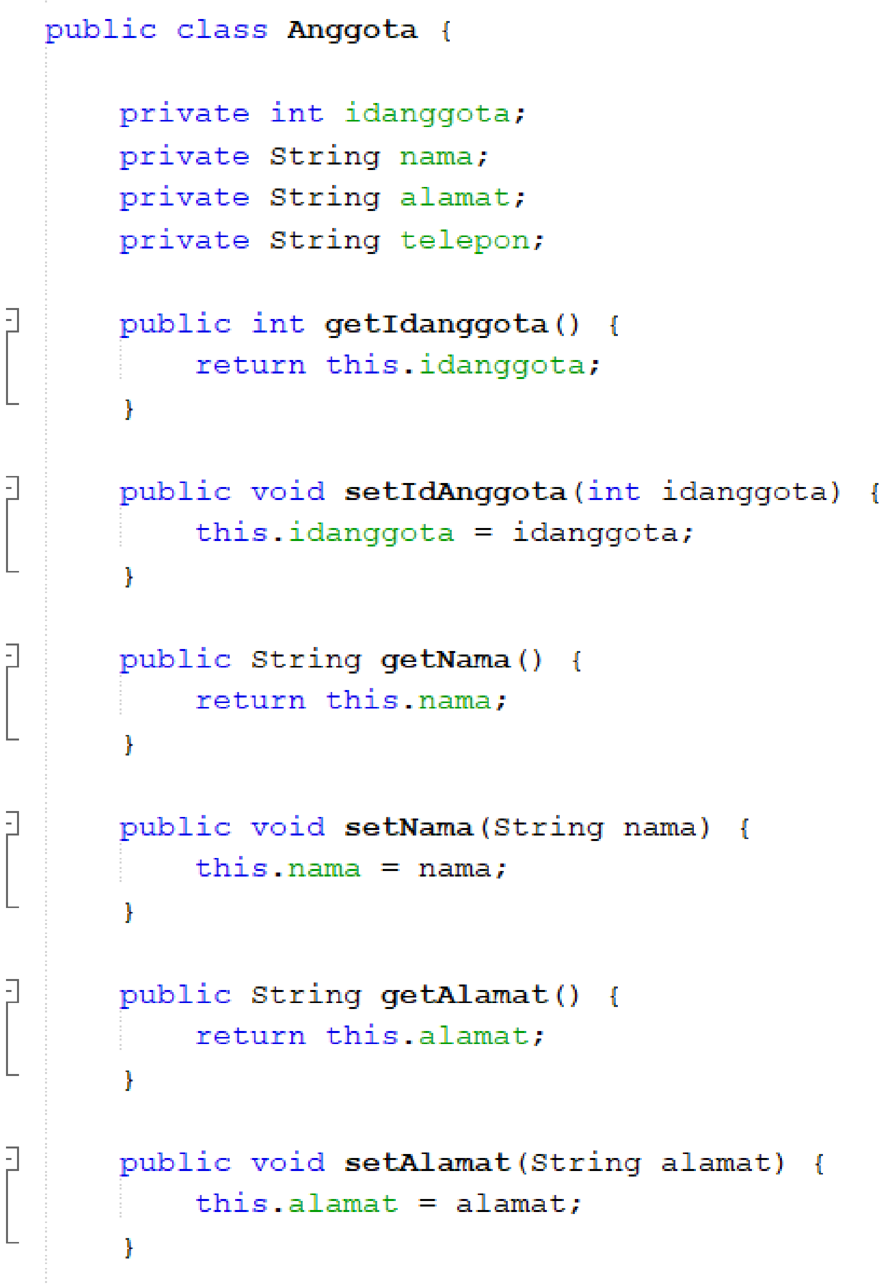
NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

## Percobaan 3.6

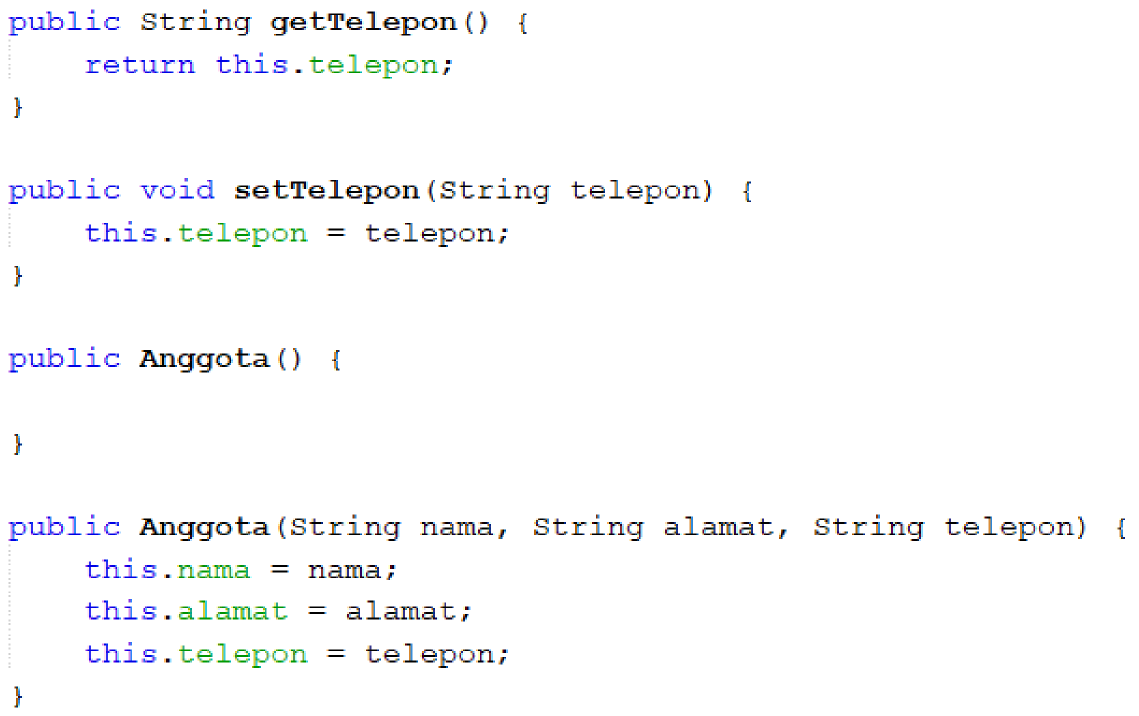


1. Class Anggota :

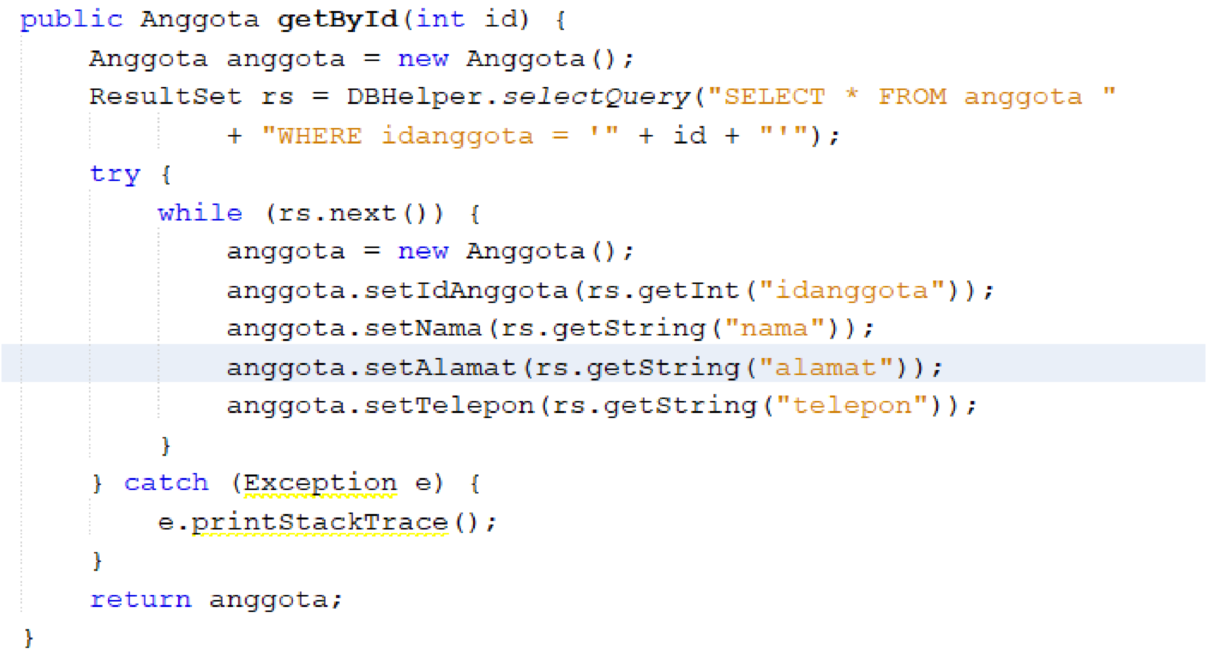
NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO



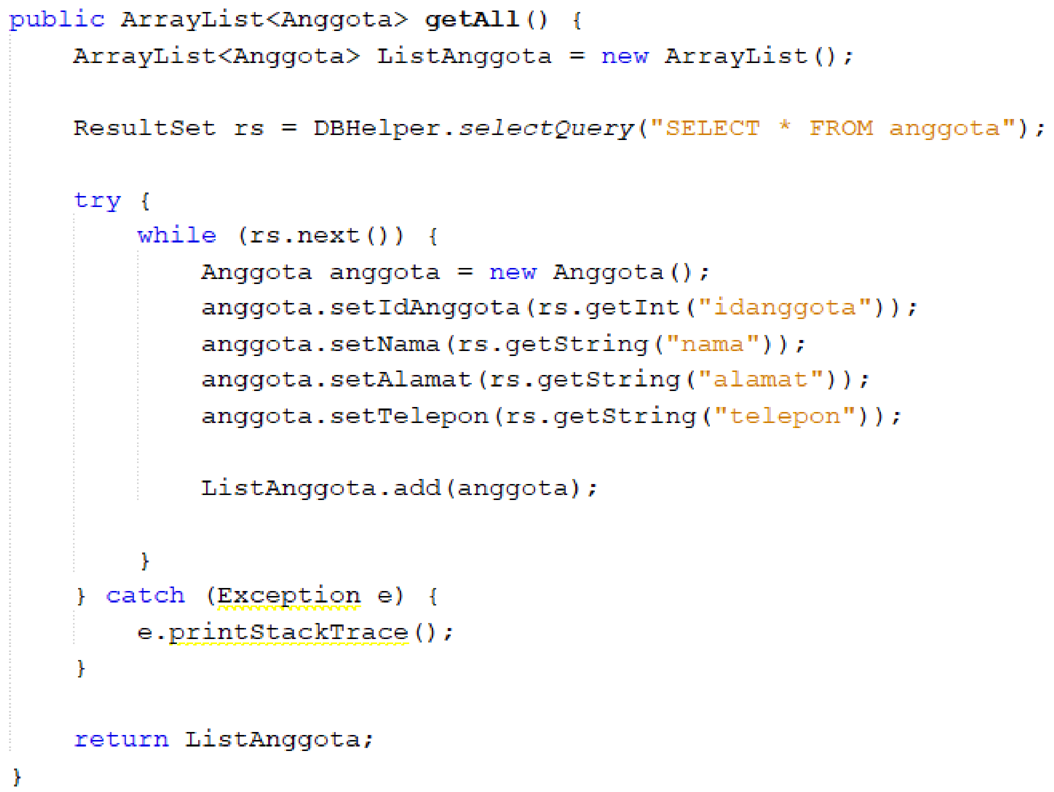
```
public class Anggota {

    private int idanggota;
    private String nama;
    private String alamat;
    private String telepon;

    public int getIdanggota() {
        return this.idanggota;
    }

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

    public String getNama() {
        return this.nama;
    }

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

    public String getAlamat() {
        return this.alamat;
    }

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



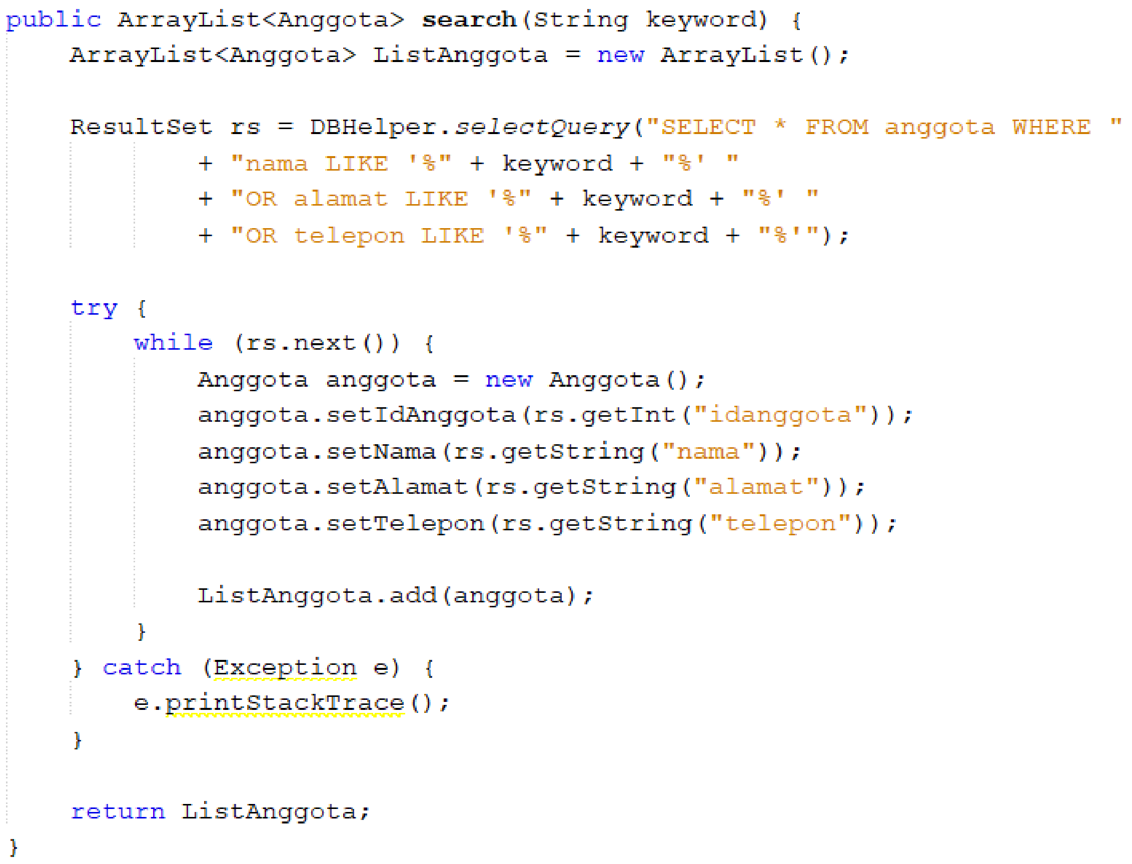
```
public String getTelepon() {
    return this.telepon;
}

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

public Anggota() {

}

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



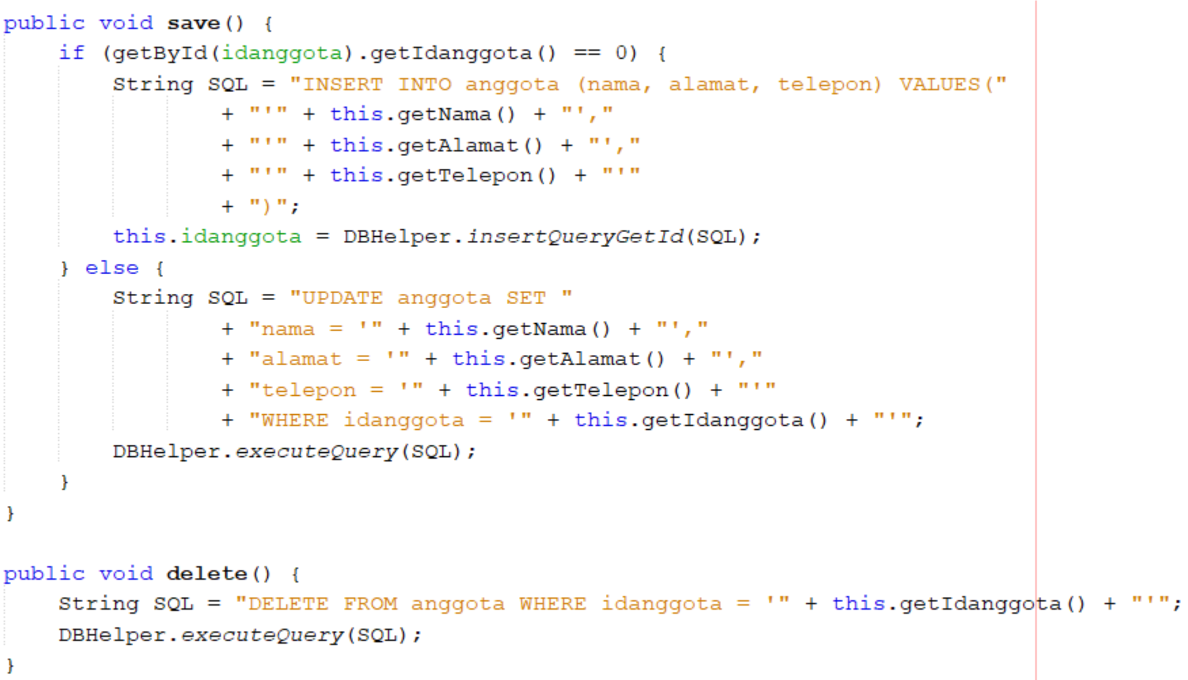
```
public Anggota getById(int id) {
    Anggota anggota = new Anggota();
    ResultSet rs = DBHelper.selectQuery("SELECT * FROM anggota "
            + "WHERE idanggota = '" + id + "'");
    try {
        while (rs.next()) {
            anggota = new Anggota();
            anggota.setIdAnggota(rs.getInt("idanggota"));
            anggota.setNama(rs.getString("nama"));
            anggota.setAlamat(rs.getString("alamat"));
            anggota.setTelepon(rs.getString("telepon"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return anggota;
}
```



```java
public ArrayList<Anggota> getAll() {
    ArrayList<Anggota> ListAnggota = new ArrayList();

    ResultSet rs = DBHelper.selectQuery("SELECT * FROM anggota");

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

            ListAnggota.add(anggota);

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

    return ListAnggota;
}
```

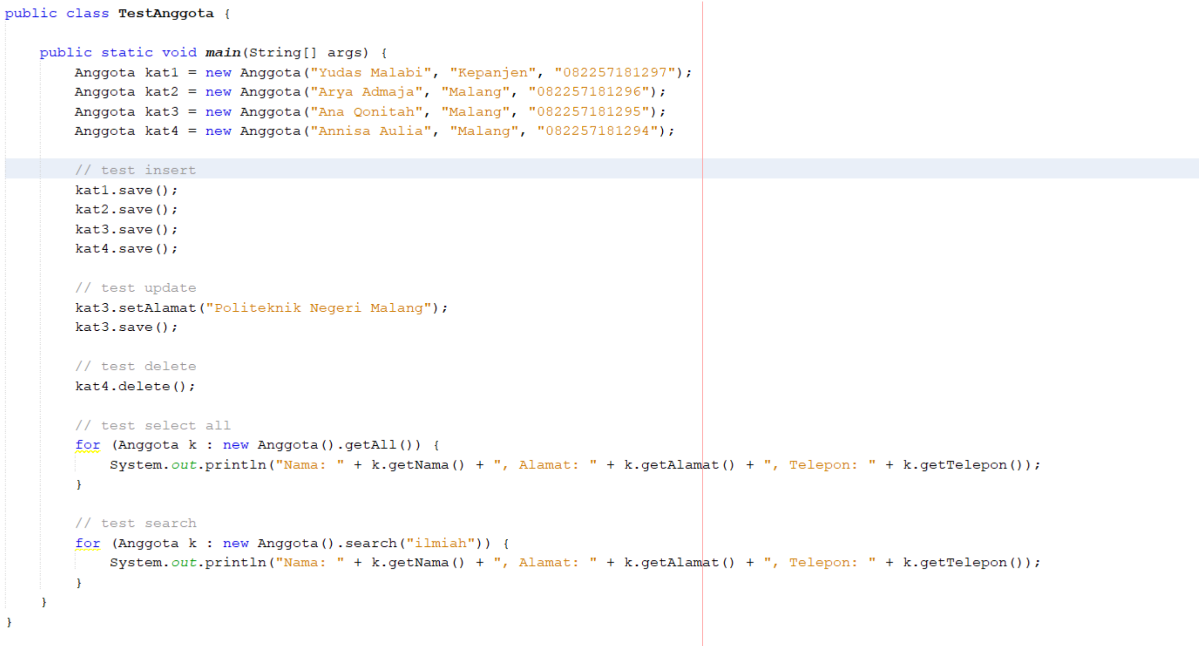


```java
public ArrayList<Anggota> search(String keyword) {
    ArrayList<Anggota> ListAnggota = new ArrayList();

    ResultSet rs = DBHelper.selectQuery("SELECT * FROM anggota WHERE "
            + "nama LIKE '%" + keyword + "%' "
            + "OR alamat LIKE '%" + keyword + "%' "
            + "OR telepon LIKE '%" + keyword + "%'");

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

            ListAnggota.add(anggota);
        }
    } 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.getNama() + "',"
                + "'" + this.getAlamat() + "',"
                + "'" + this.getTelepon() + "'"
                + ")";
        this.idanggota = DBHelper.insertQueryGetId(SQL);
    } else {
        String SQL = "UPDATE anggota SET "
                + "nama = '" + this.getNama() + "',"
                + "alamat = '" + this.getAlamat() + "',"
                + "telepon = '" + this.getTelepon() + "'"
                + "WHERE idanggota = '" + this.getIdanggota() + "'";
        DBHelper.executeQuery(SQL);
    }
}

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

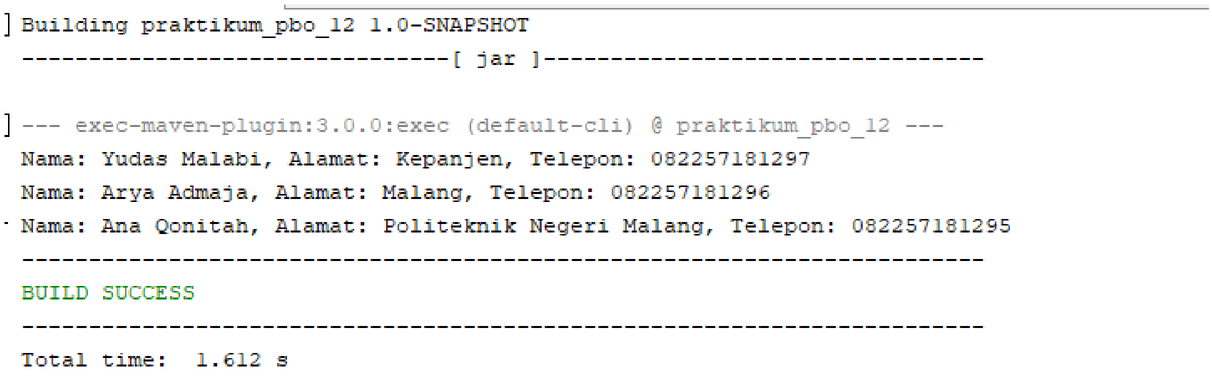
2. Lakukan test pada class TestBackend pada package frontend.



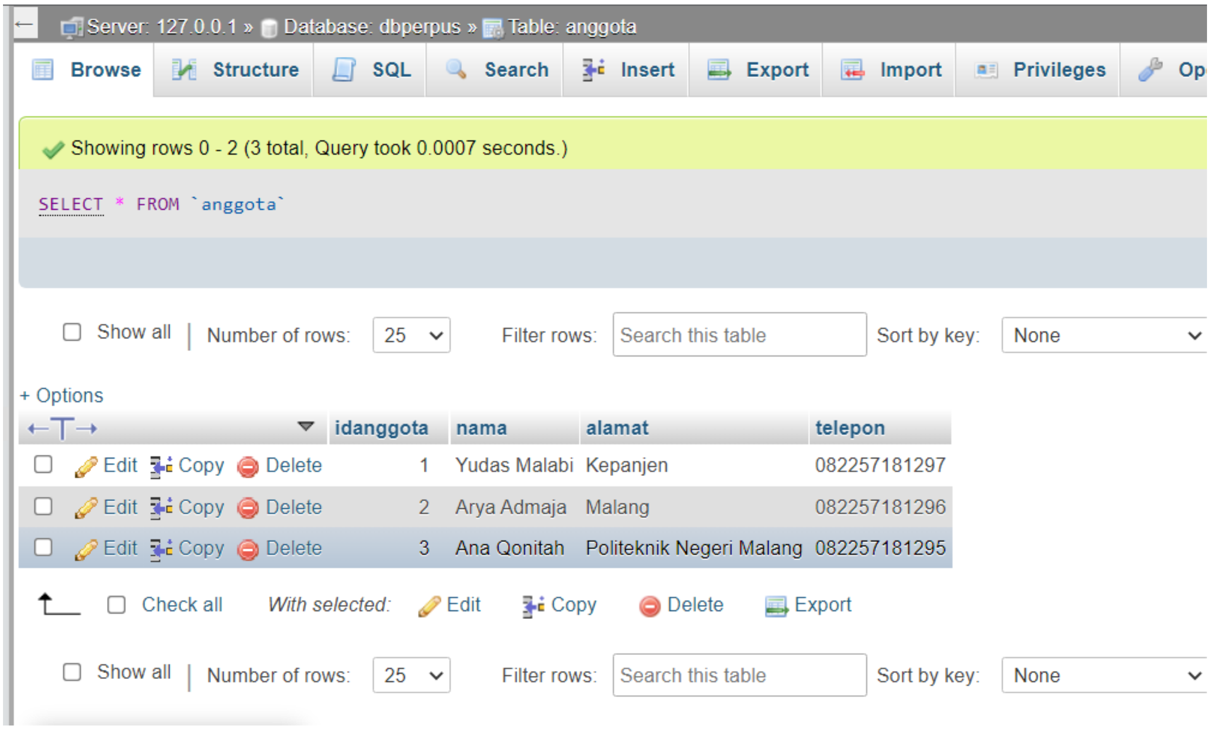
```
public class TestAnggota {

    public static void main(String[] args) {
        Anggota kat1 = new Anggota("Yudas Malabi", "Kepanjen", "082257181297");
        Anggota kat2 = new Anggota("Arya Admaja", "Malang", "082257181296");
        Anggota kat3 = new Anggota("Ana Qonitah", "Malang", "082257181295");
        Anggota kat4 = new Anggota("Annisa Aulia", "Malang", "082257181294");

        // test insert
        kat1.save();
        kat2.save();
        kat3.save();
        kat4.save();

        // test update
        kat3.setAlamat("Politeknik Negeri Malang");
        kat3.save();

        // test delete
        kat4.delete();

        // test select all
        for (Anggota k : new Anggota().getAll()) {
            System.out.println("Nama: " + k.getNama() + ", Alamat: " + k.getAlamat() + ", Telepon: " + k.getTelepon());
        }

        // test search
        for (Anggota k : new Anggota().search("ilmiah")) {
            System.out.println("Nama: " + k.getNama() + ", Alamat: " + k.getAlamat() + ", Telepon: " + k.getTelepon());
        }
    }
}
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

output :



```
] Building praktikum_pbo_12 1.0-SNAPSHOT
  --------------------------------[ jar ]---------------------------------

] --- exec-maven-plugin:3.0.0:exec (default-cli) @ praktikum_pbo_12 ---
  Nama: Yudas Malabi, Alamat: Kepanjen, Telepon: 082257181297
  Nama: Arya Admaja, Alamat: Malang, Telepon: 082257181296
  Nama: Ana Qonitah, Alamat: Politeknik Negeri Malang, Telepon: 082257181295
  ------------------------------------------------------------------------
  BUILD SUCCESS
  ------------------------------------------------------------------------
  Total time:  1.612 s
```



Showing rows 0 - 2 (3 total, Query took 0.0007 seconds.)

```
SELECT * FROM `anggota`
```

□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

| | idanggota | nama | alamat | telepon |
|---|---|---|---|---|
| □ Edit Copy Delete | 1 | Yudas Malabi | Kepanjen | 082257181297 |
| □ Edit Copy Delete | 2 | Arya Admaja | Malang | 082257181296 |
| □ Edit Copy Delete | 3 | Ana Qonitah | Politeknik Negeri Malang | 082257181295 |

□ Check all *With selected:* Edit Copy Delete Export

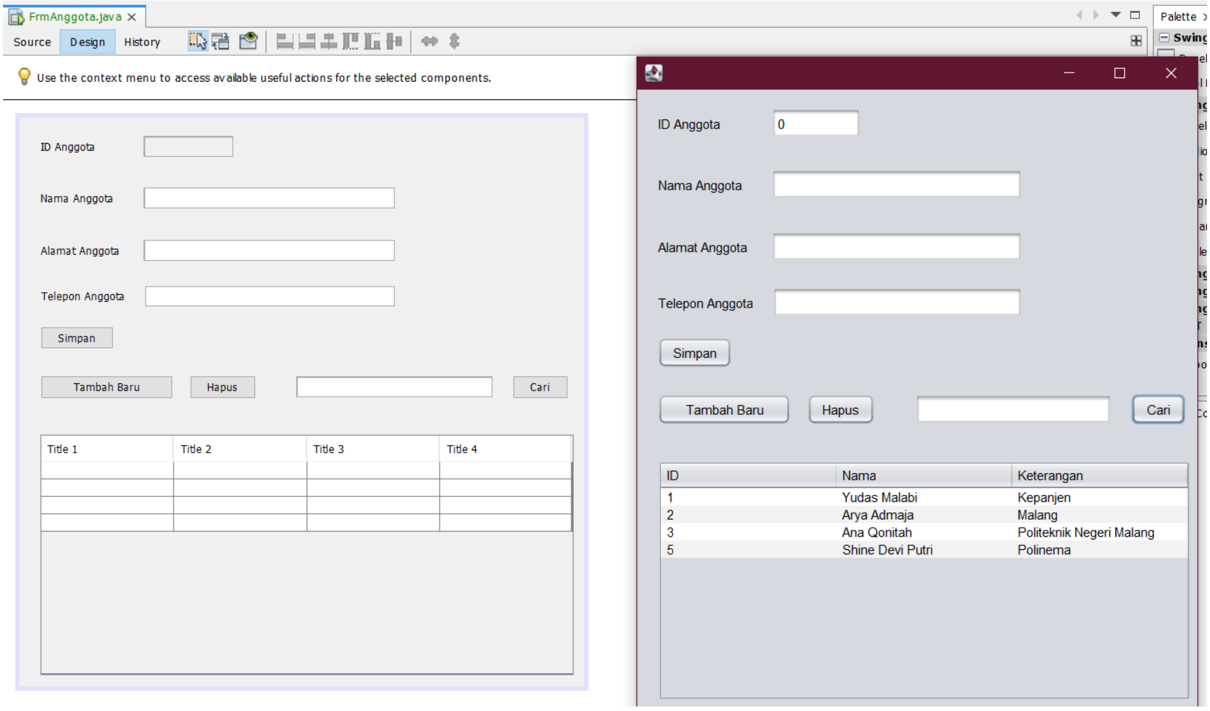
□ Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

*2 Data Anggota masuk ke database MySQL*

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

## Percobaan 3.7

FormAnggota :

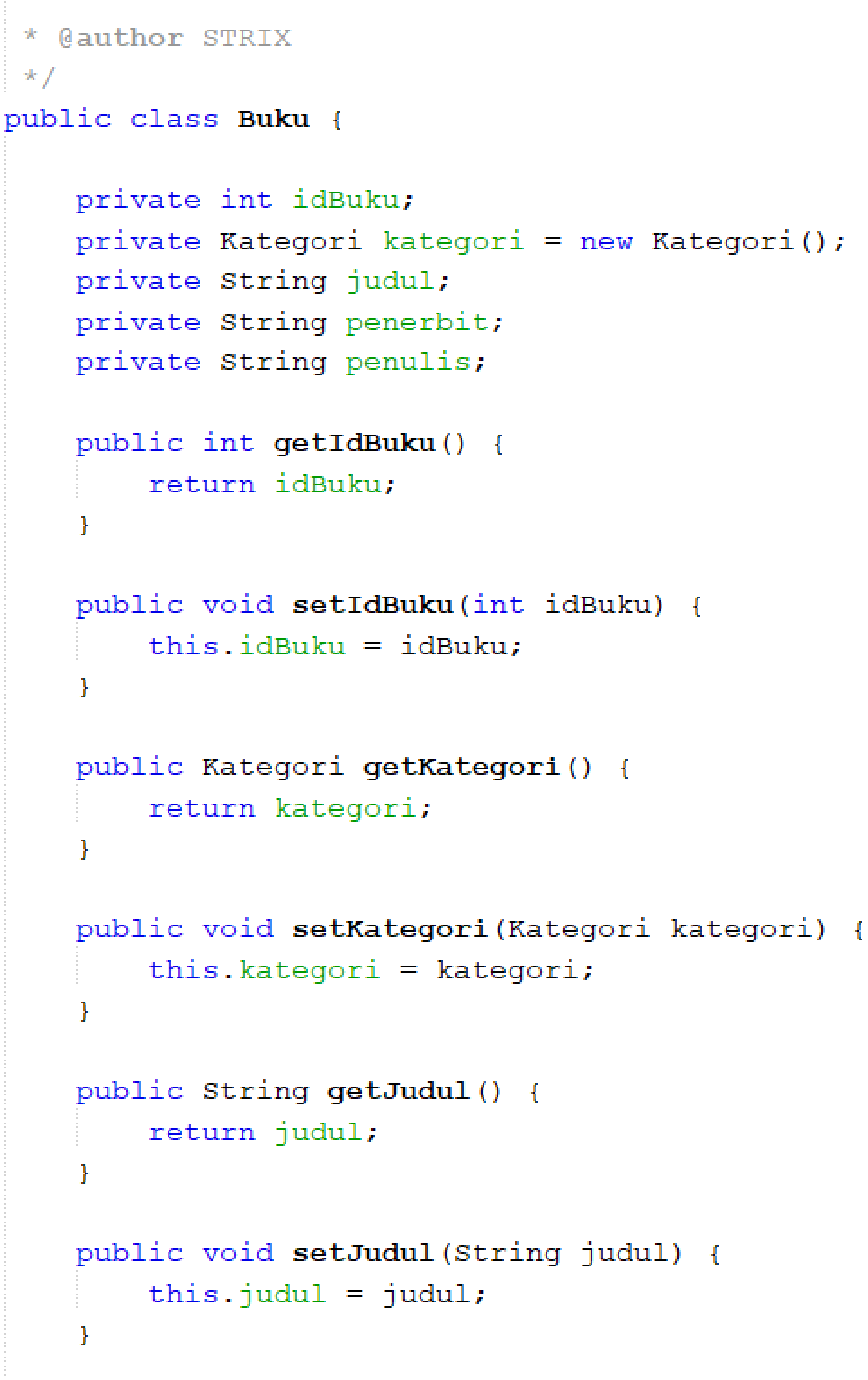


ID Anggota: 0

| ID | Nama | Keterangan |
|---|---|---|
| 1 | Yudas Malabi | Kepanjen |
| 2 | Arya Admaja | Malang |
| 3 | Ana Qonitah | Politeknik Negeri Malang |
| 5 | Shine Devi Putri | Polinema |

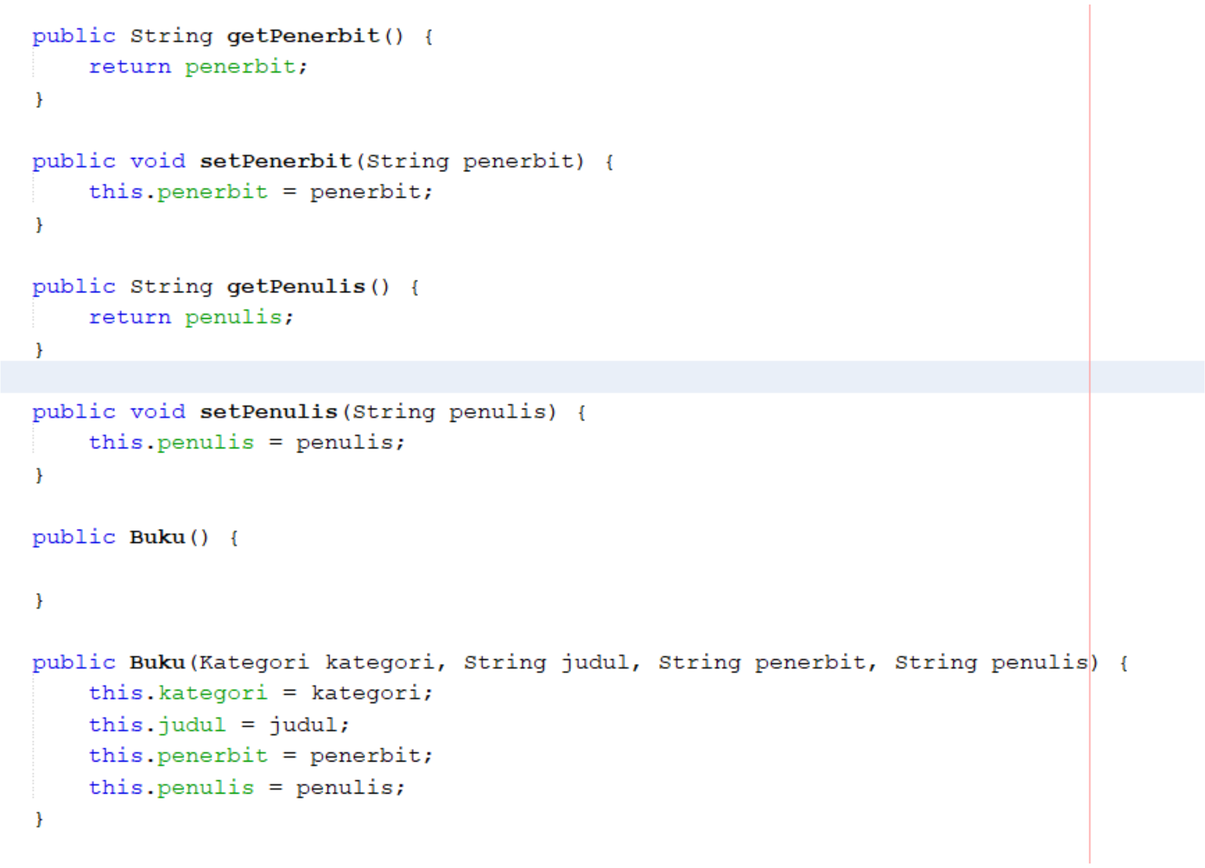
## Percobaan 3.8

Class Buku :

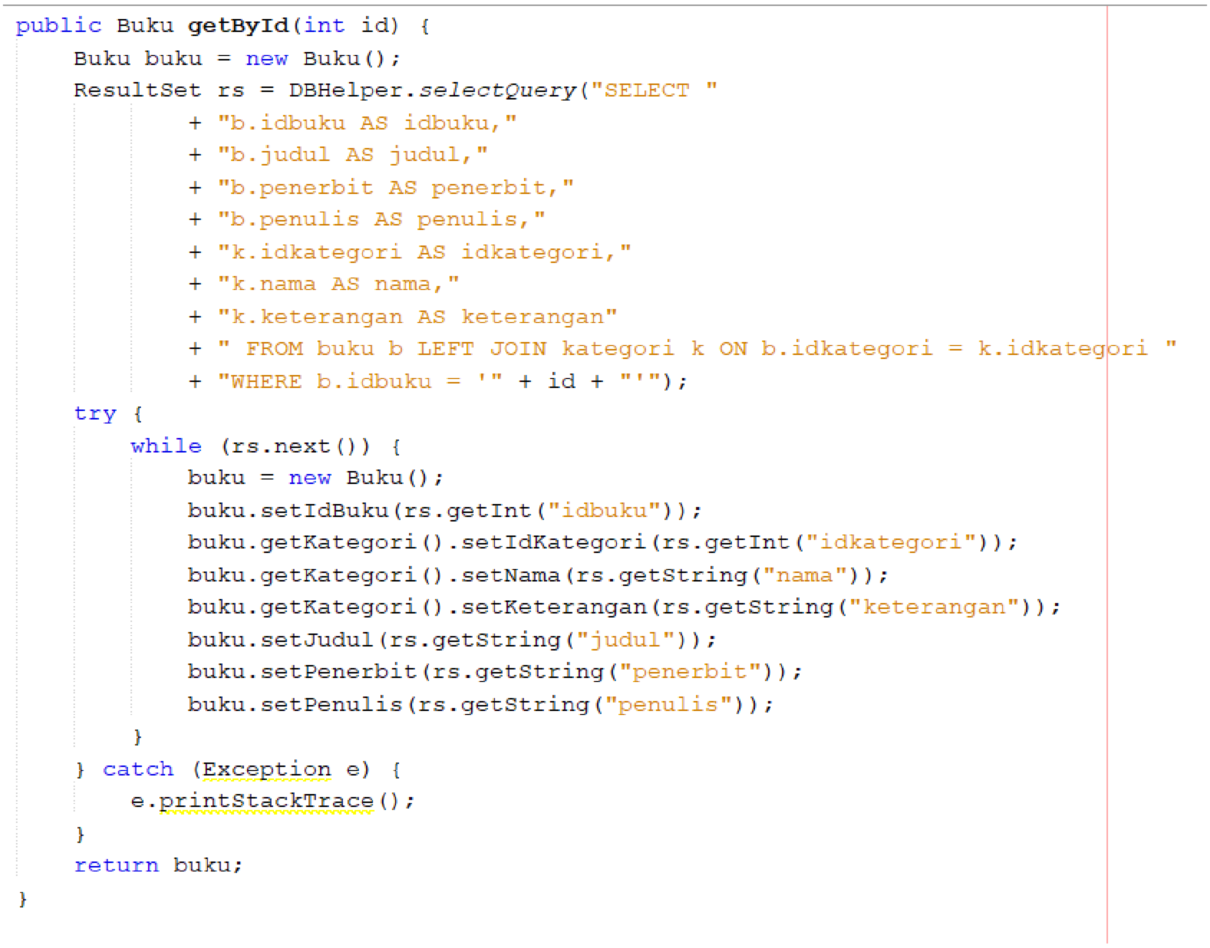


```
 * @author STRIX
 */
public class Buku {

    private int idBuku;
    private Kategori kategori = new Kategori();
    private String judul;
    private String penerbit;
    private String 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;
    }
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

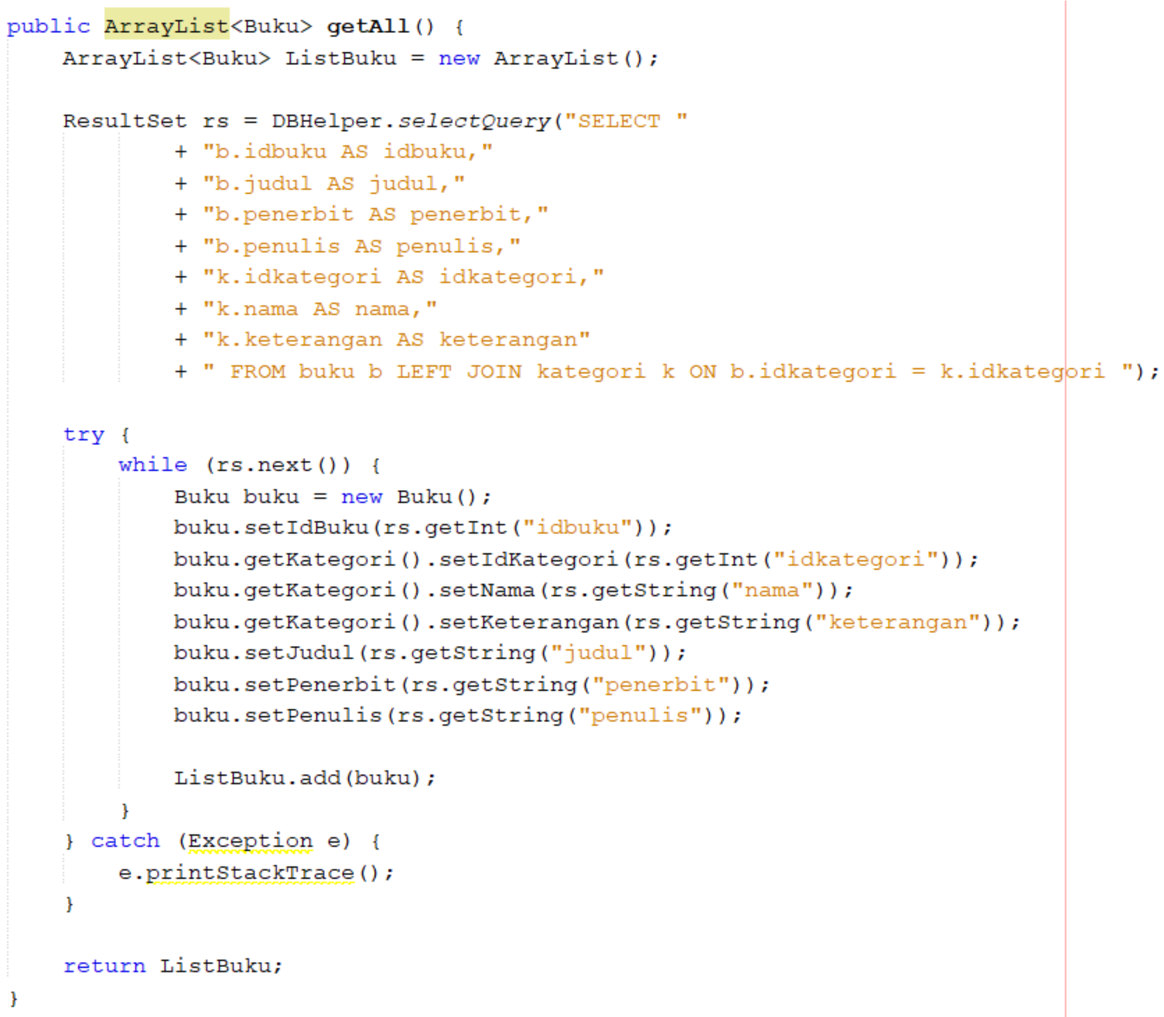


```
public String getPenerbit() {
    return penerbit;
}

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

public String getPenulis() {
    return penulis;
}

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



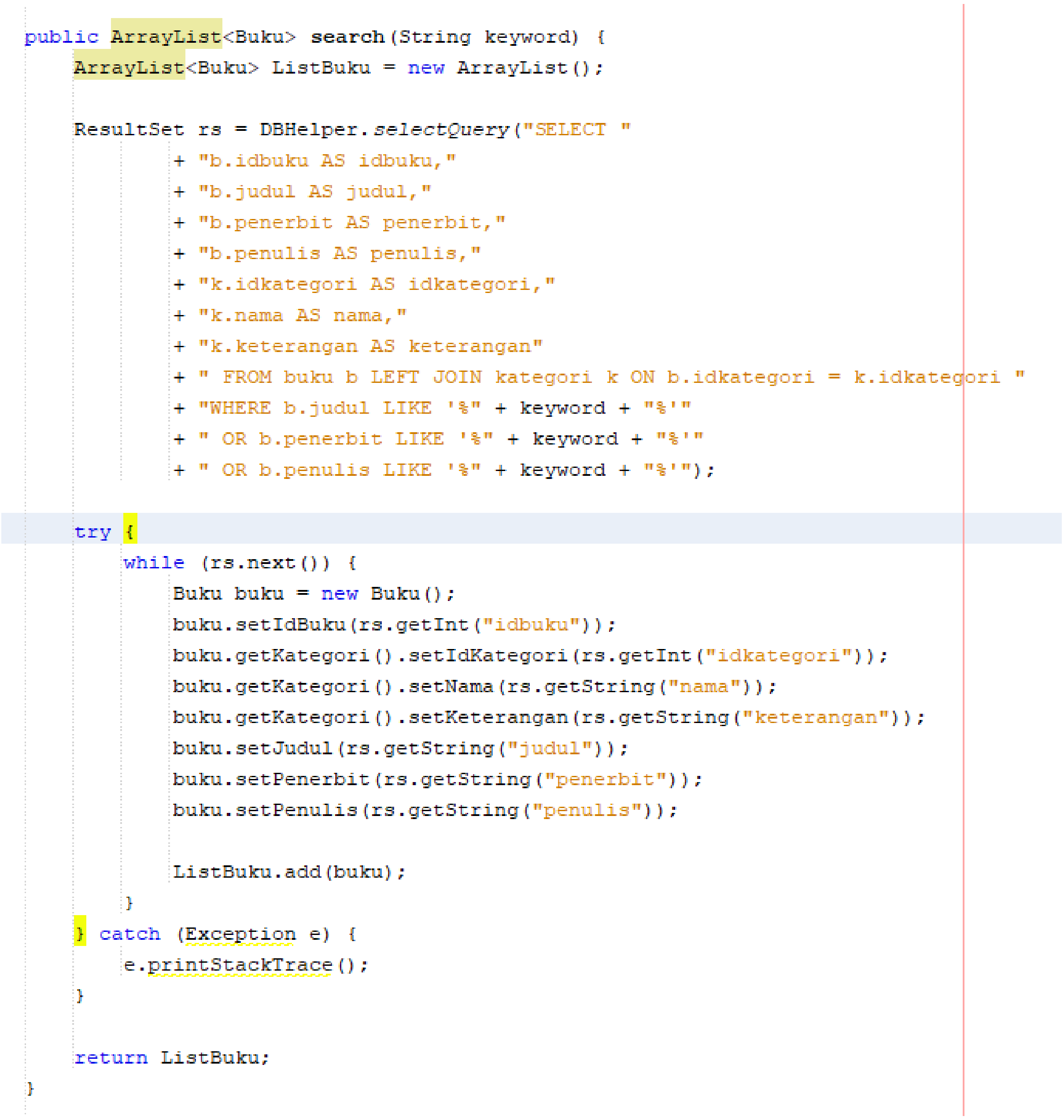
```
public Buku getById(int id) {
    Buku buku = new Buku();
    ResultSet rs = DBHelper.selectQuery("SELECT "
            + "b.idbuku AS idbuku,"
            + "b.judul AS judul,"
            + "b.penerbit AS penerbit,"
            + "b.penulis AS penulis,"
            + "k.idkategori AS idkategori,"
            + "k.nama AS nama,"
            + "k.keterangan AS keterangan"
            + " FROM buku b LEFT JOIN kategori k ON b.idkategori = k.idkategori "
            + "WHERE b.idbuku = '" + id + "'");
    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;
}
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO



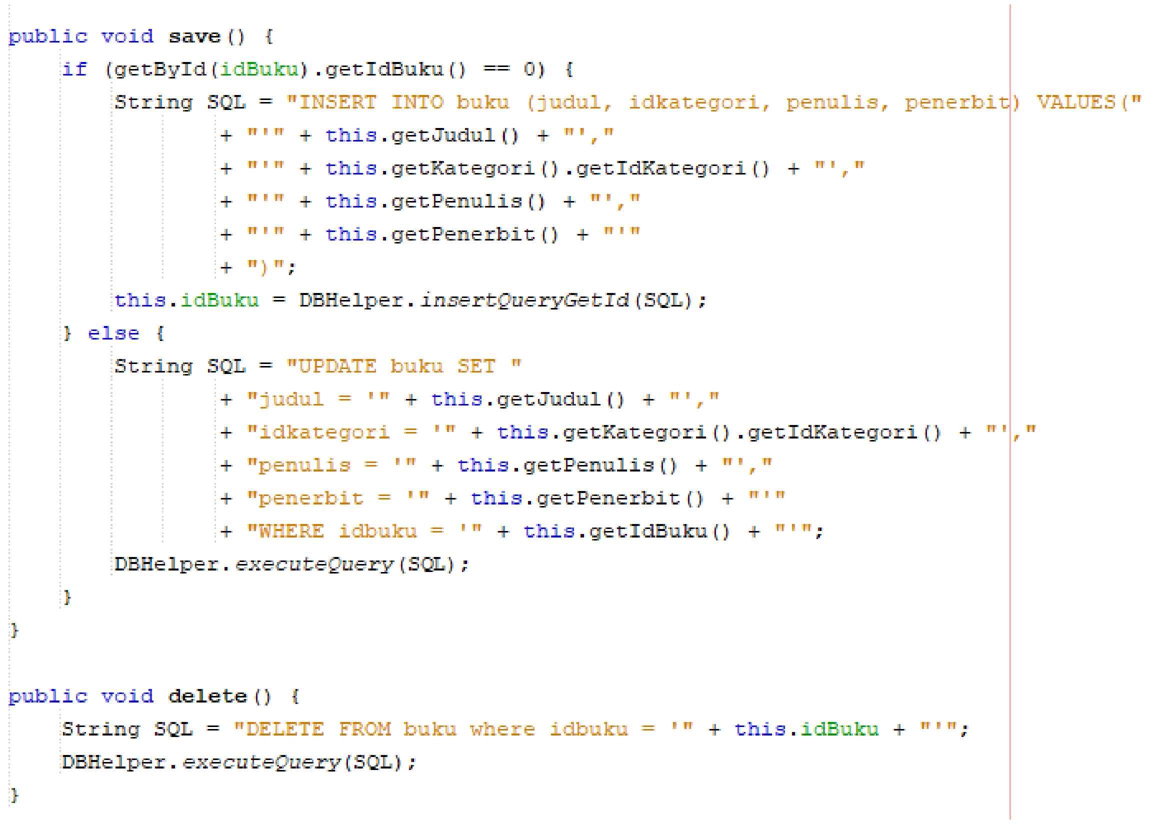
```java
public ArrayList<Buku> getAll() {
    ArrayList<Buku> ListBuku = new ArrayList();

    ResultSet rs = DBHelper.selectQuery("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 ");

    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;
}
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO



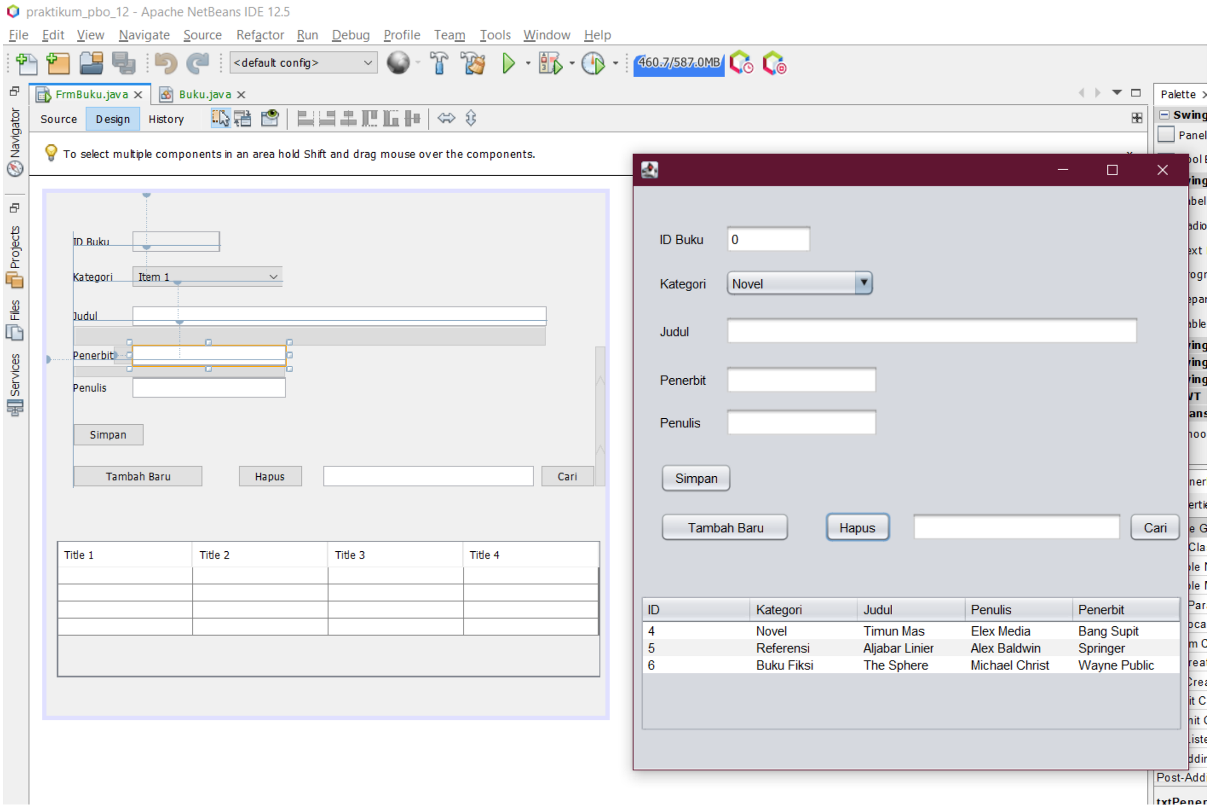
```
public ArrayList<Buku> search(String keyword) {
    ArrayList<Buku> ListBuku = new ArrayList();

    ResultSet rs = DBHelper.selectQuery("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.judul LIKE '%" + keyword + "%'"
            + " OR b.penerbit LIKE '%" + keyword + "%'"
            + " OR b.penulis LIKE '%" + keyword + "%'");

    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;
}
```

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO



```
public void save() {
    if (getById(idBuku).getIdBuku() == 0) {
        String SQL = "INSERT INTO buku (judul, idkategori, penulis, penerbit) VALUES("
                + "'" + this.getJudul() + "',"
                + "'" + this.getKategori().getIdKategori() + "',"
                + "'" + this.getPenulis() + "',"
                + "'" + this.getPenerbit() + "'"
                + ")";
        this.idBuku = DBHelper.insertQueryGetId(SQL);
    } else {
        String SQL = "UPDATE buku SET "
                + "judul = '" + this.getJudul() + "',"
                + "idkategori = '" + this.getKategori().getIdKategori() + "',"
                + "penulis = '" + this.getPenulis() + "',"
                + "penerbit = '" + this.getPenerbit() + "'"
                + "WHERE idbuku = '" + this.getIdBuku() + "'";
        DBHelper.executeQuery(SQL);
    }
}

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

FormBuku :



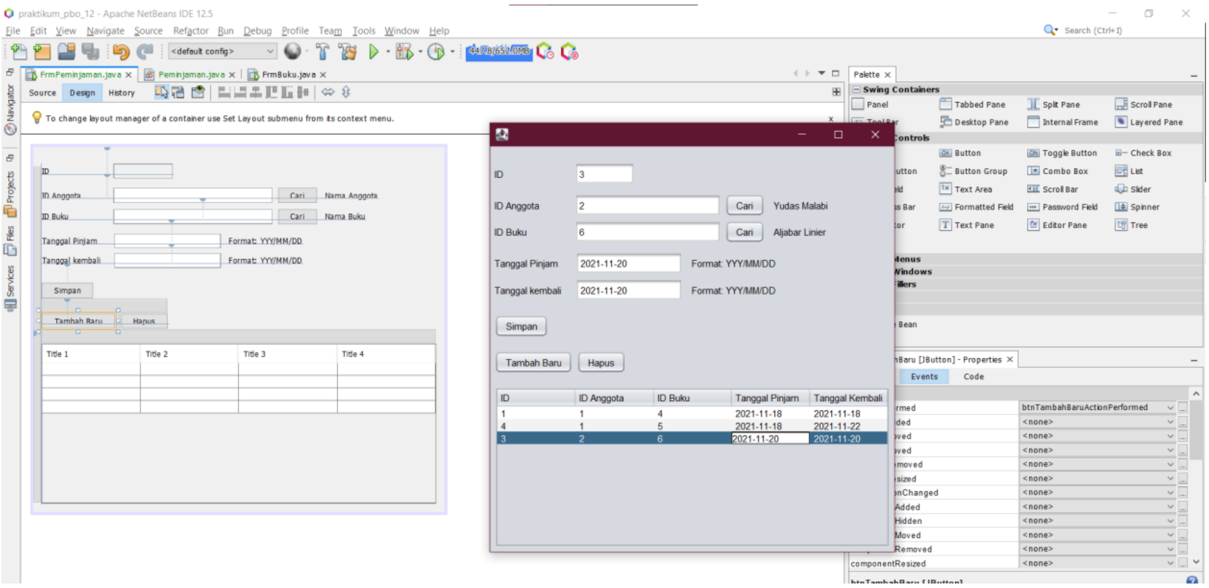
To select multiple components in an area hold Shift and drag mouse over the components.

| | |
|---|---|
| ID Buku | 0 |
| Kategori | Novel |
| Judul | |
| Penerbit | |
| Penulis | |

| ID | Kategori | Judul | Penulis | Penerbit |
|---|---|---|---|---|
| 4 | Novel | Timun Mas | Elex Media | Bang Supit |
| 5 | Referensi | Aljabar Linier | Alex Baldwin | Springer |
| 6 | Buku Fiksi | The Sphere | Michael Christ | Wayne Public |

NAMA : YUDAS MALABI
NIM : 2041720054
KELAS : TI 2C
MATERI : PRAKTIKUM PBO

## TUGAS



ID: 3

ID Anggota: 2 — Cari — Yudas Malabi

ID Buku: 6 — Cari — Aljabar Linier

Tanggal Pinjam: 2021-11-20 — Format: YYY/MM/DD

Tanggal kembali: 2021-11-20 — Format: YYY/MM/DD

| ID | ID Anggota | ID Buku | Tanggal Pinjam | Tanggal Kembali |
|---|---|---|---|---|
| 1 | 1 | 4 | 2021-11-18 | 2021-11-18 |
| 4 | 1 | 5 | 2021-11-18 | 2021-11-22 |
| 3 | 2 | 6 | 2021-11-20 | 2021-11-20 |

praktikum_pbo_12 - Apache NetBeans IDE 12.5
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
FrmPeminjaman.java
Peminjaman.java
FrmBuku.java
Source
Design
History
Simpan
Tambah Baru
Hapus
Palette
Swing Containers
Events
Code