Koneksi Database

Brigida Arie Minartiningtyas, M.Kom

Membuat Database

```
CREATE DATABASE `perpustakaan` /*!40100 DEFAULT CHARACTER SET latin1 */;
```

Menggunakan Database

```
use perpustakaan;
```

Membuat Tabel

```sql
CREATE TABLE `buku` (
  `kode_buku` varchar(5) NOT NULL,
  `judul_buku` varchar(20) NOT NULL,
  `pengarang` varchar(15) NOT NULL,
  `penerbit` varchar(15) NOT NULL,
  PRIMARY KEY (`kode_buku`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

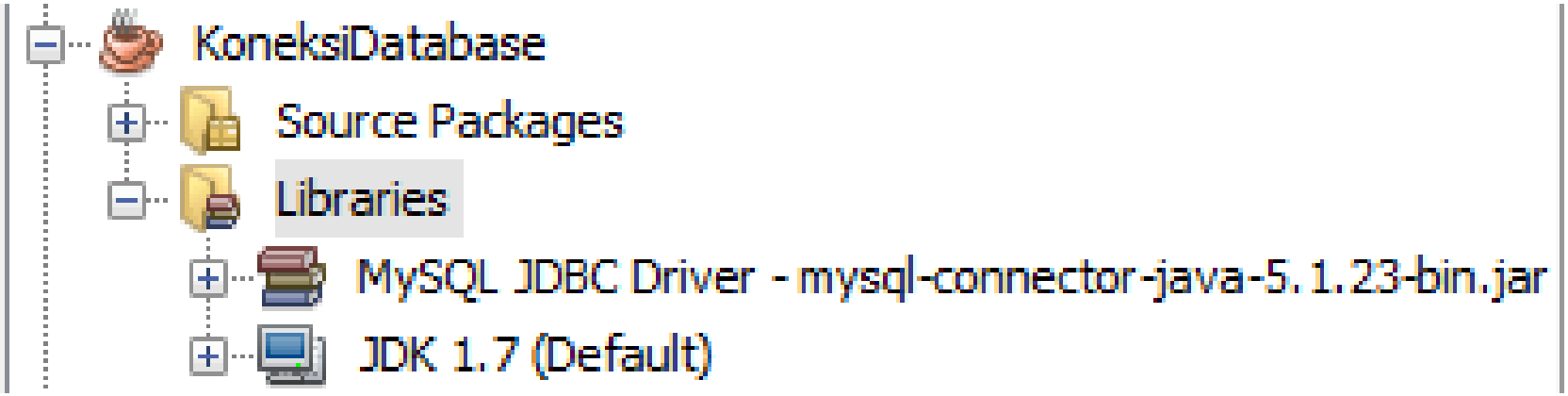
Menambahkan Data ke Tabel

```
insert into buku values ('B001','Conan','Gosho','Jepang');
```

Isi Tabel

	kode_buku	judul_buku	pengarang	penerbit
▶	B001	Conan	Gosho	Jepang
*	NULL	NULL	NULL	NULL

Add Library

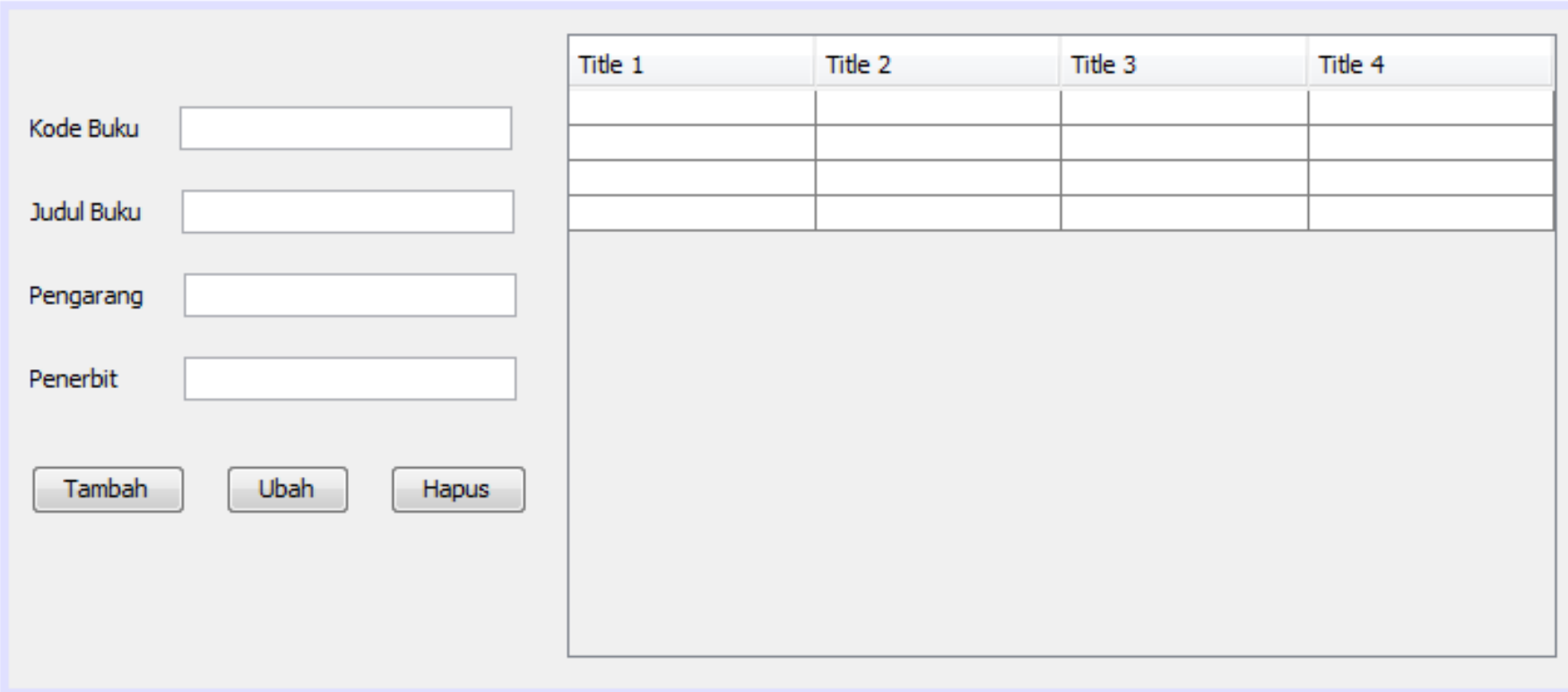

KoneksiDatabase
Source Packages
Libraries
MySQL JDBC Driver - mysql-connector-java-5.1.23-bin.jar
JDK 1.7 (Default)

Koneksi Java dengan MySQL

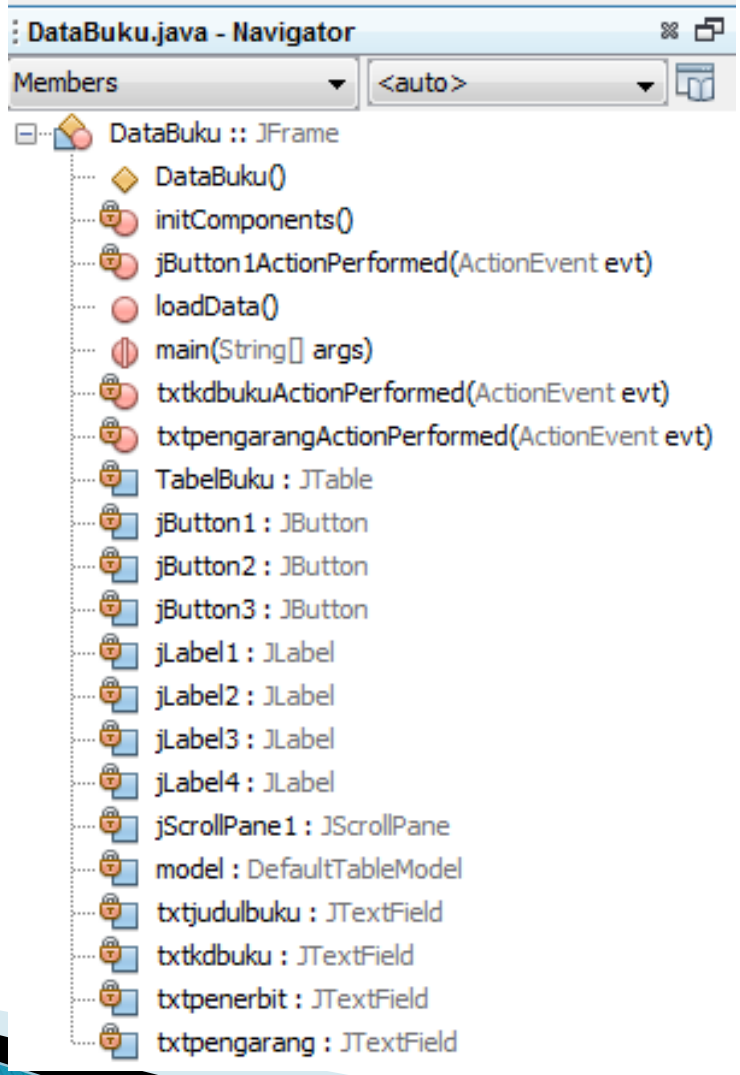
```java
package buku;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class KonekMySQL {
    static Connection koneksi;
    public static Connection getKoneksi() {
     try {
        String url = "jdbc:mysql://localhost/perpustakaan";
        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 Membuat Koneksi");
        }
    return koneksi;
    }
    public static void main(String args[]){
        getKoneksi();
    }
    }
```

```
put - KoneksiDatabase (run)
run:
BUILD SUCCESSFUL (total time: 3 seconds)
```

Membuat Form



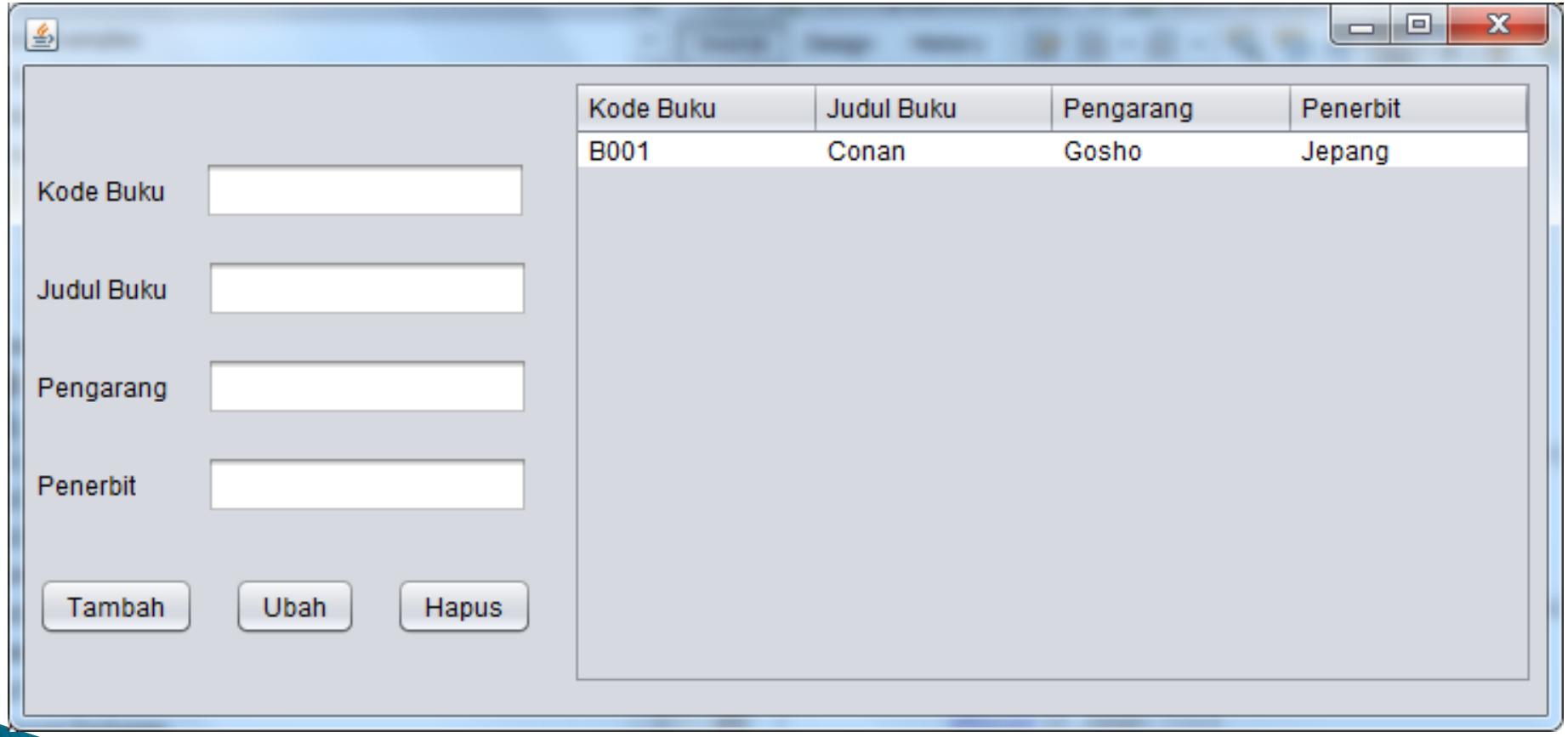
Ganti Nama Komponen



Tambahkan Coding di Source

```java
package buku;

import javax.swing.table.DefaultTableModel;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.sql.PreparedStatement;
import java.sql.Connection;

public class DataBuku extends javax.swing.JFrame {

    private DefaultTableModel model;
    /**
     * Creates new form DataBuku
     */

    public DataBuku() {
        initComponents();

        model = new DefaultTableModel();

        TabelBuku.setModel(model);

        model.addColumn("Kode Buku");
        model.addColumn("Judul Buku");
        model.addColumn("Pengarang");
        model.addColumn("Penerbit");

      loadData();
    }
```

Tambahkan Coding di Source

```
public void loadData(){
// menghapus seluruh data
    model.getDataVector().removeAllElements();
// memberi tahu bahwa data telah kosong
    model.fireTableDataChanged();
    try{
        Connection c = KonekMySQL.getKoneksi(); Statement s = c.createStatement();
        String sql = "SELECT * FROM BUKU";
        ResultSet r = s.executeQuery(sql);
        while(r.next()){
       // lakukan penelusuran baris
            Object[] o = new Object[4];
            o[0] = r.getString("kode_buku");
            o[1] = r.getString("judul_buku");
            o[2] = r.getString("pengarang");
            o[3] = r.getString("penerbit");
            model.addRow(o);
        }
        r.close();
        s.close();
    }catch(SQLException e){ System.out.println("Terjadi Error");
    }
    }
```

Tampilan saat load