



TEKNIK PENYIMPANAN DATA PADA APLIKASI DESKTOP

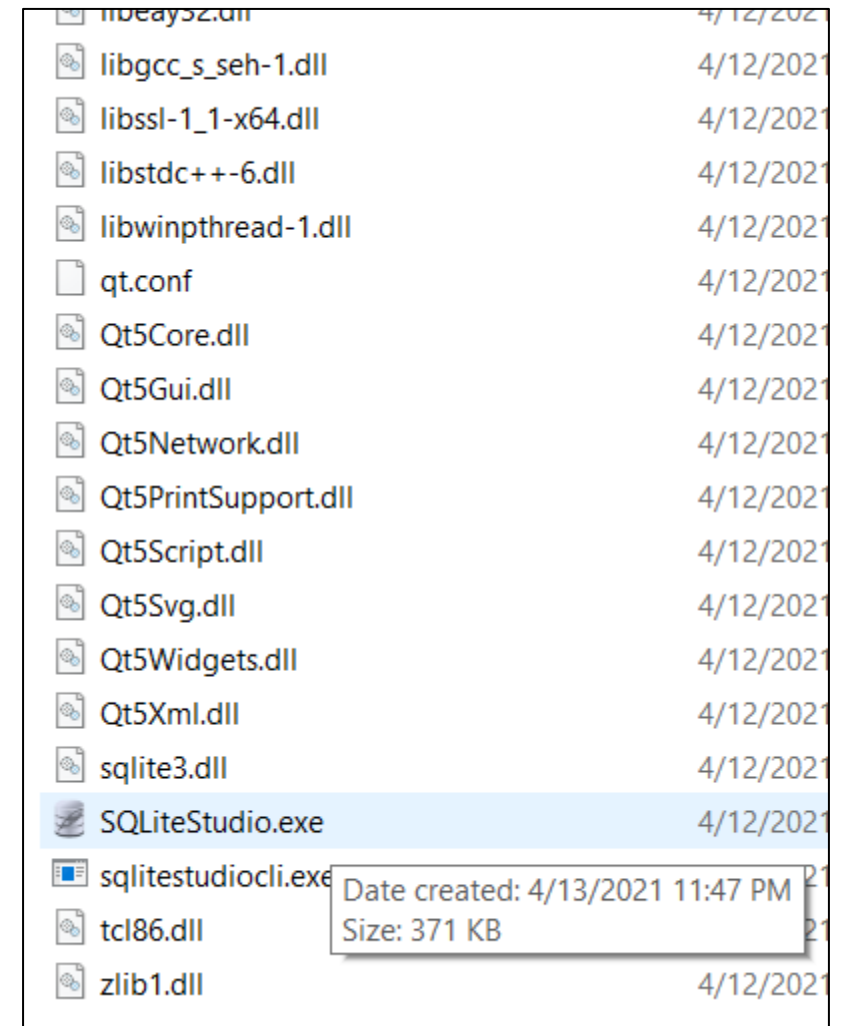


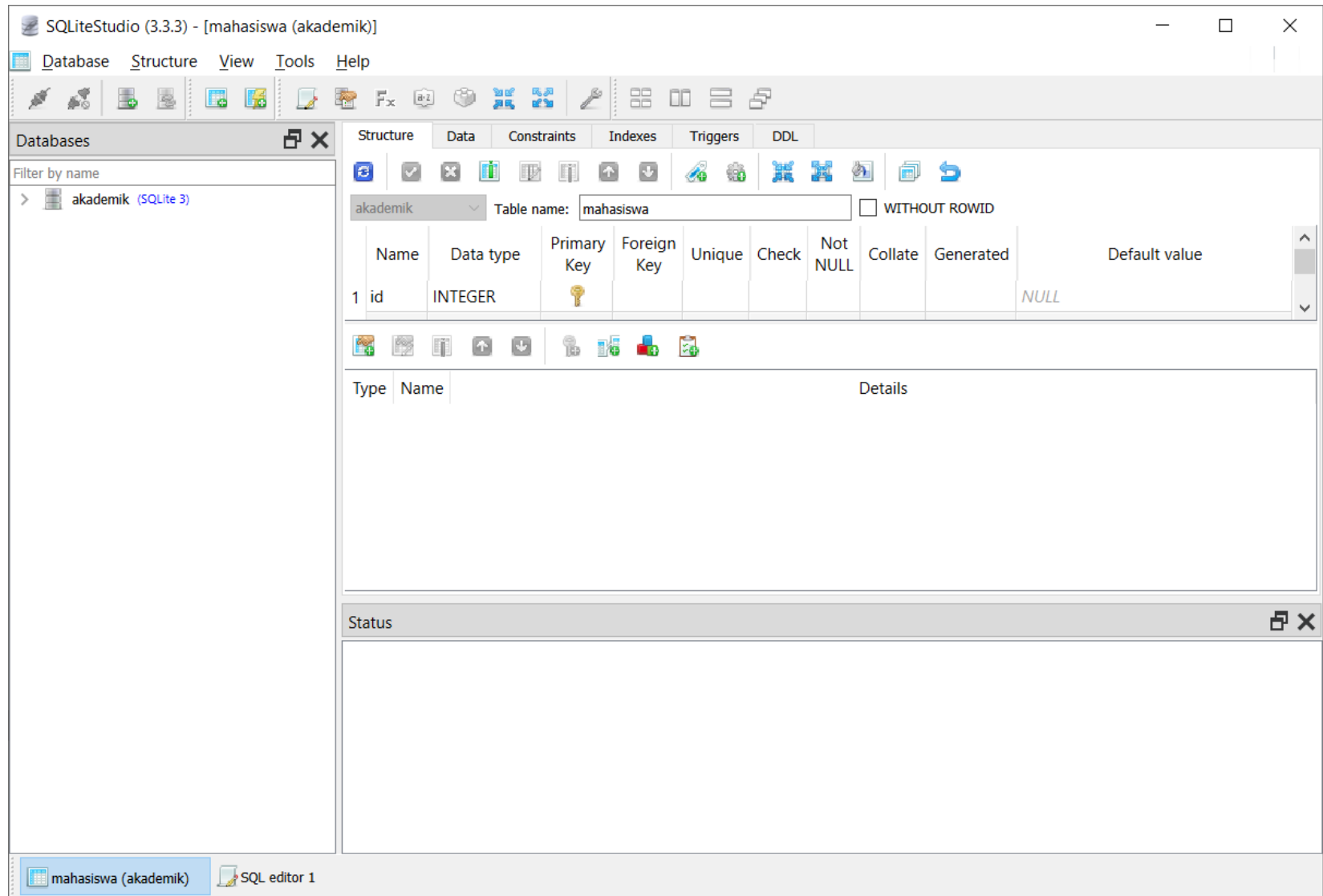
Pengantar

- SQLite adalah perpustakaan perangkat lunak yang menerapkan engine database SQL secara mandiri, tanpa memerlukan server, tanpa perlu melakukan konfigurasi, dan bersifat transaksional.
- SQLite adalah sebuah engine database SQL yang langsung tertanam atau pada aplikasi. Tidak seperti kebanyakan database SQL lainnya, SQLite tidak memiliki server yang terpisah dari aplikasi. SQLite membaca dan menulis langsung ke file disk biasa. Database SQLite memiliki fitur lengkap dengan banyak tabel, indexes, trigger, dan tampilan, serta tersimpan pada satu file tunggal dalam hard-disk. Format file databasenya bersifat cross-platform. Sehingga Anda dapat dengan bebas menyalin database antara sistem 32-bit dan 64-bit atau antara arsitektur yang berbeda platform. Fitur-fitur ini membuat SQLite menjadi pilihan populer sebagai Application File Format.

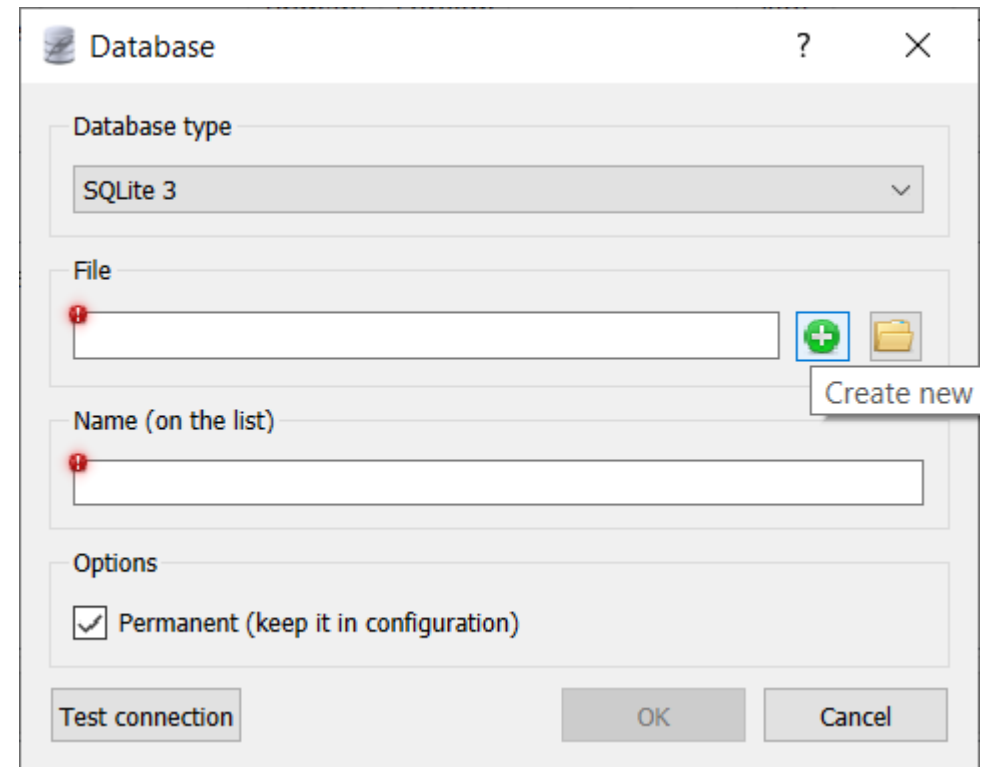
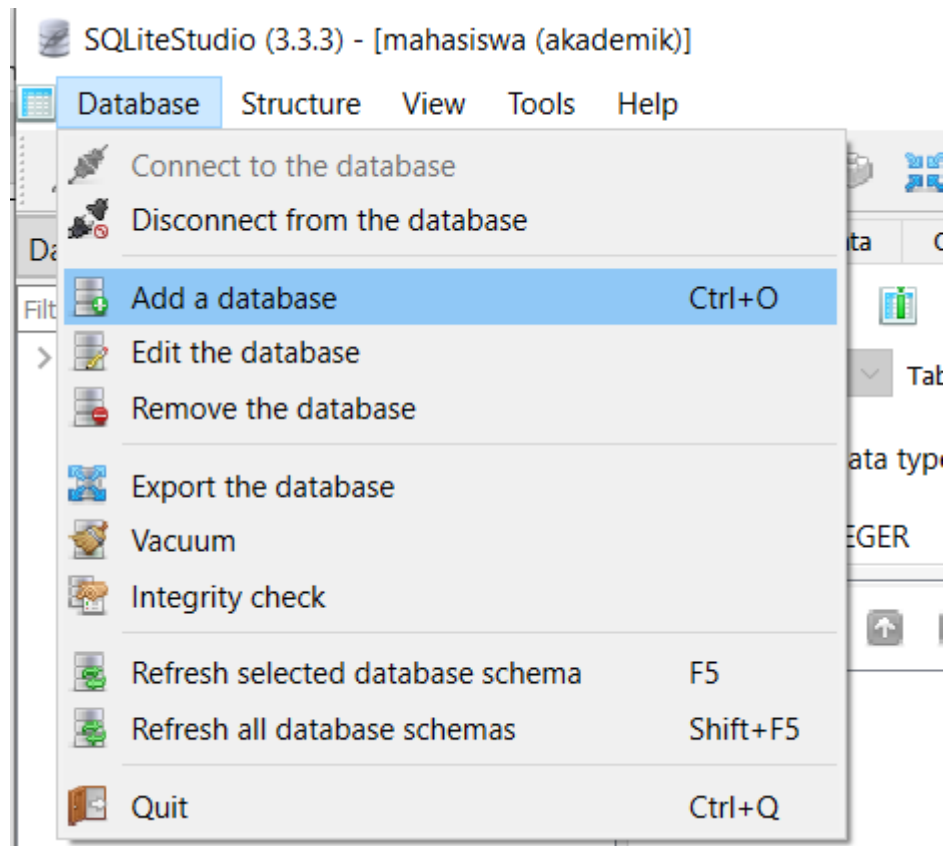
Praktikum

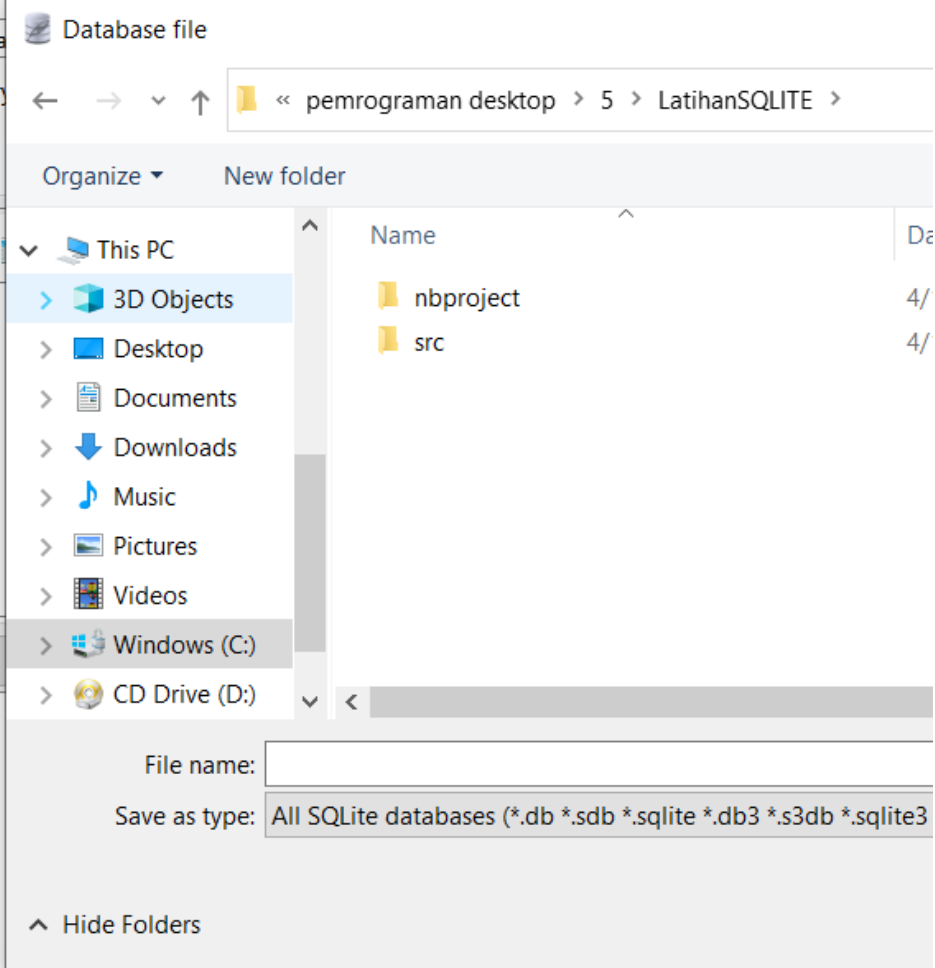
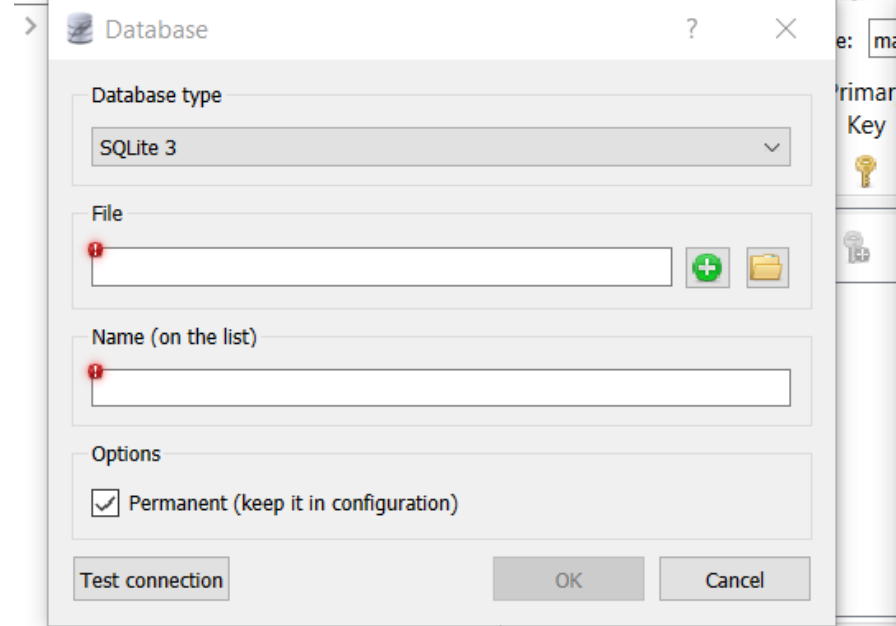
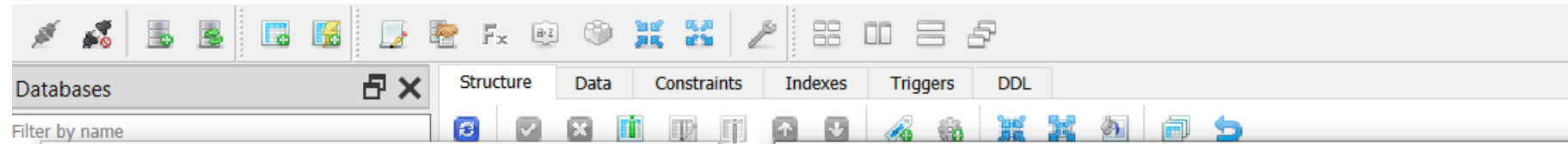
- Download terlebih dahulu sqlite studio dan sqlite jdbc
- Link download
<https://cloud.poltekssn.ac.id/s/Q5dsN9w6eR8sia7>
- Setelah didownload extract sqlite studio dan jalankan sqlitestudio.exe



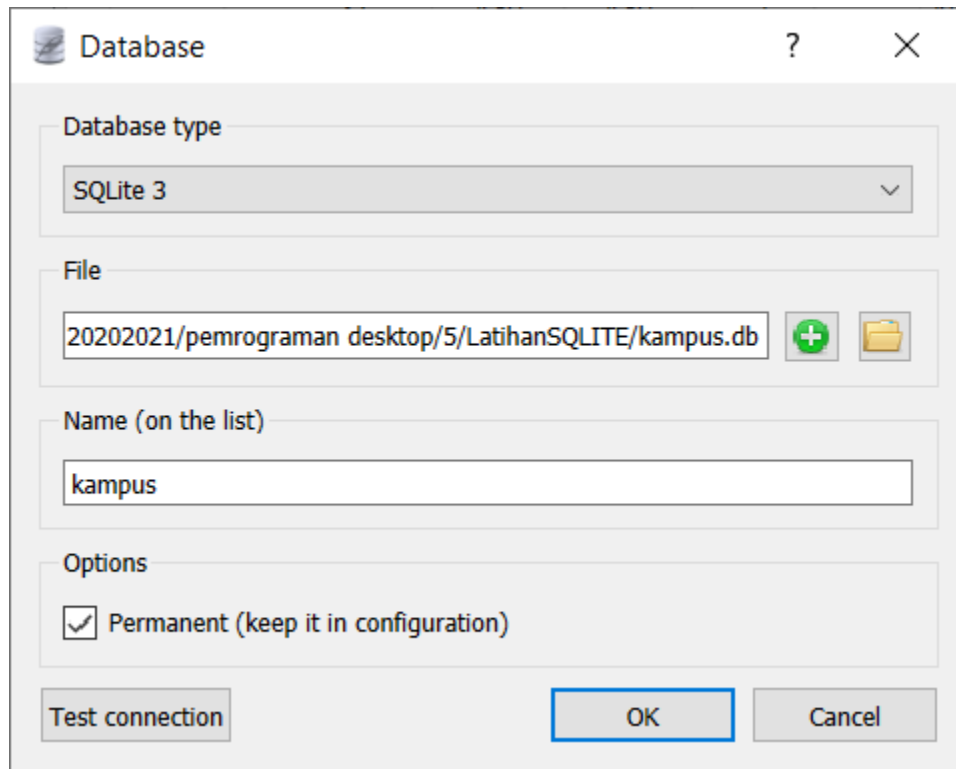


- Buat project baru dengan nama LatihanSQLITE
- Buat database “kampus” dari sqlite dan letakan di project java yang dibuat

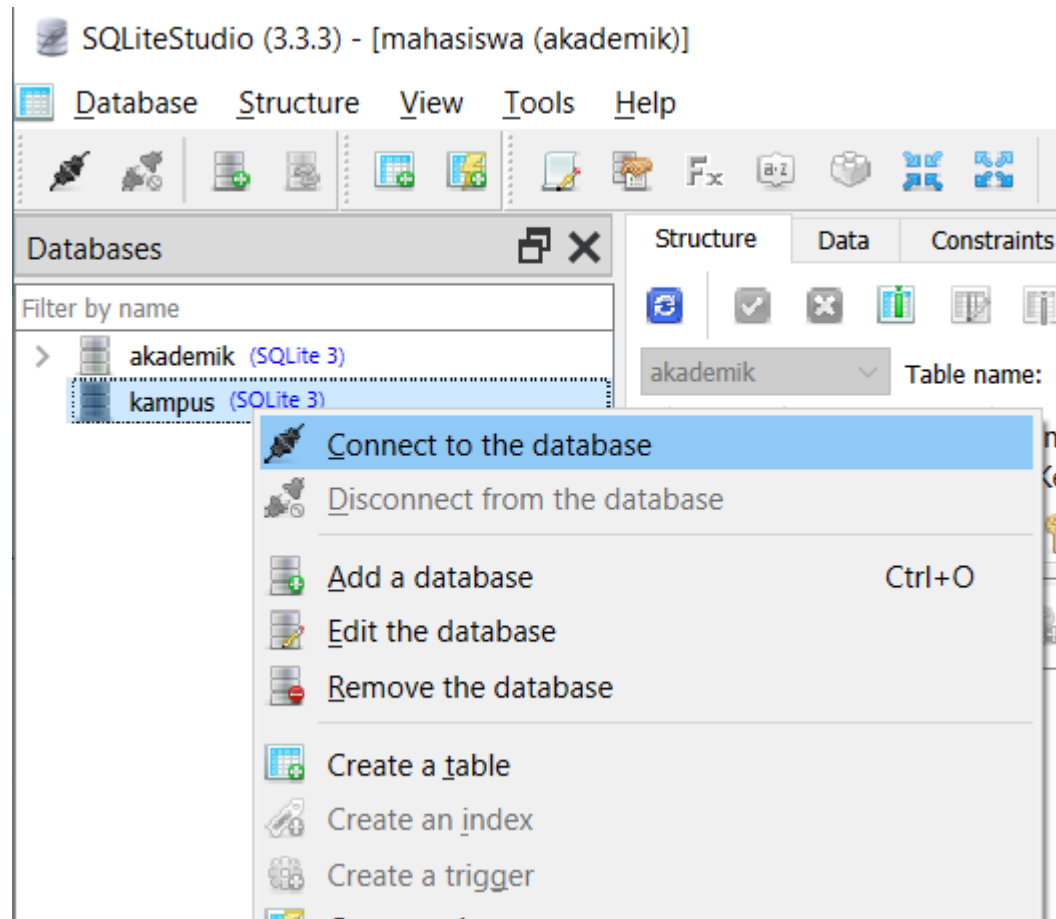




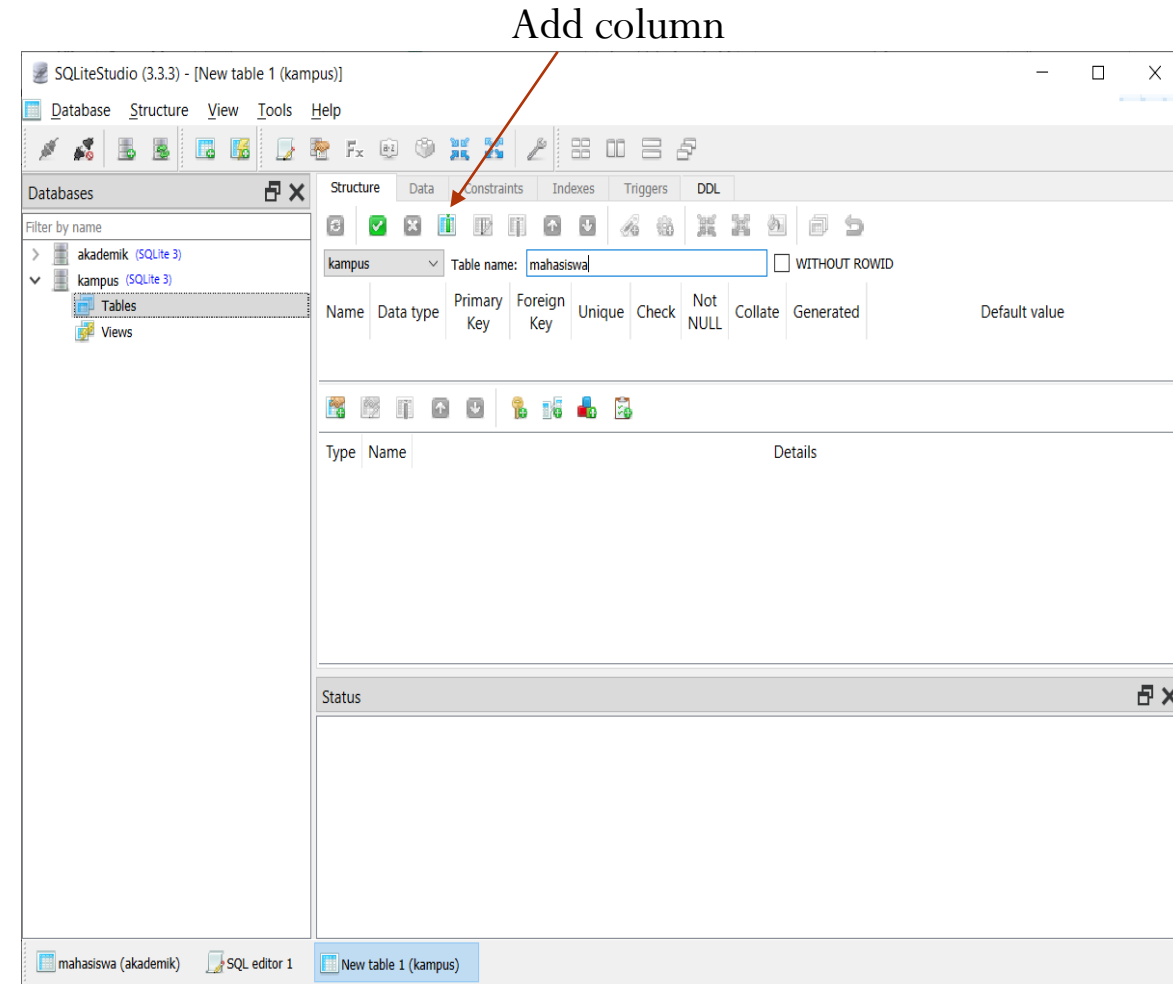
- Setelah muncul seperti dibawah tekan ok



- Klik kanan connect to the database



- Buat table dengan cara klik kanan tables dan create a table
- Isikan nama table misalkan “mahasiswa”, dilanjutkan add column











Column ? X

Name and type

Column name: Data type: Size: ,

Constraints

- ☐  Primary Key
- ☐  Foreign Key
- ☐  Unique
- ☐  Check condition
- ☐  Not NULL
- ☐  Collate
- ☐  Generated value
- ☐  Default









☐ Advanced mode

Column ? X

Name and type


Column name: Data type: Size: ,

Constraints

- ☒  Primary Key
- ☐ 
- ☐ 
- ☐ 
- ☐ 
- ☐ 
- ☐ 
- ☐ 

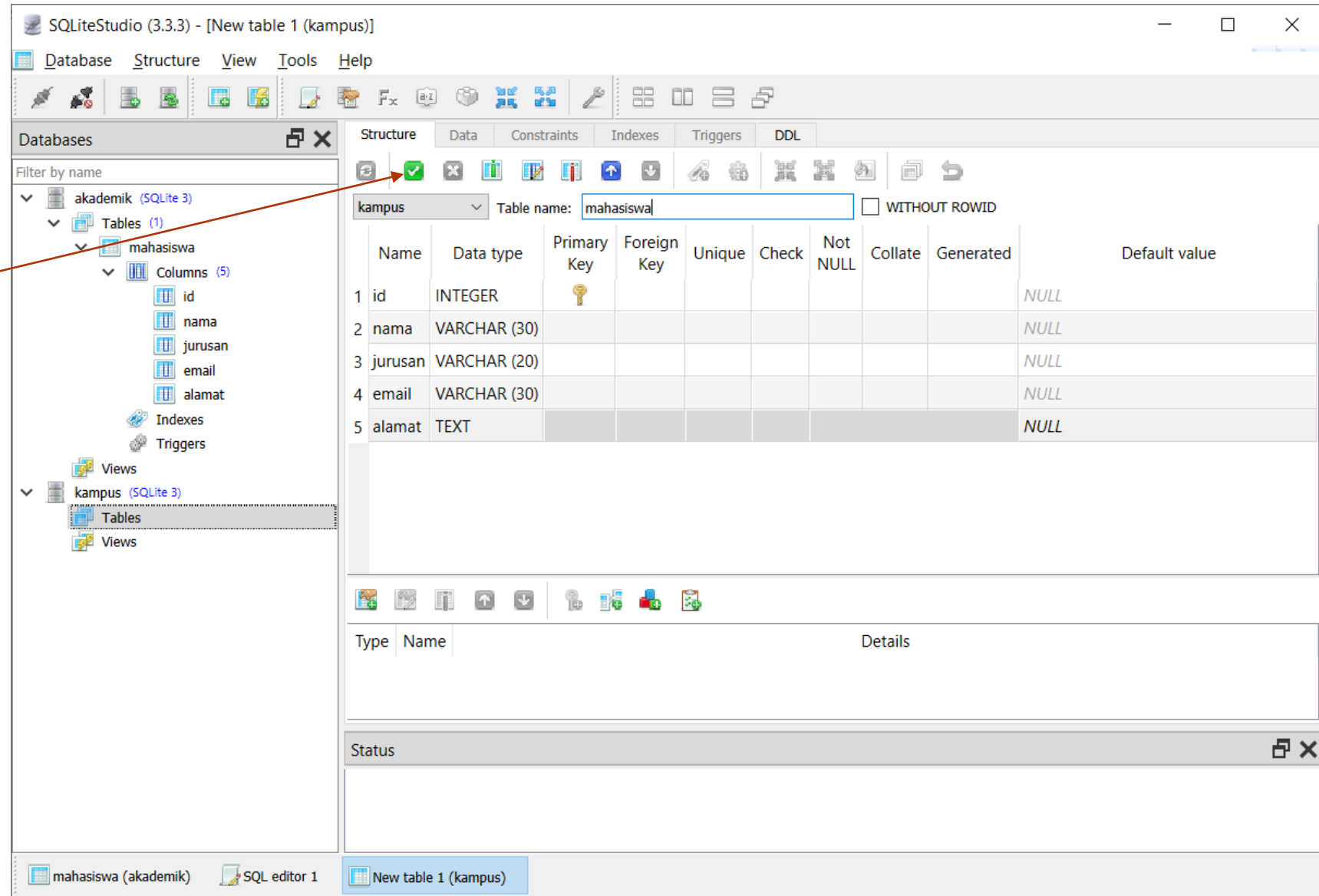
☐ Advanced mode

Edit constraint ? X

 **Primary key**

- ☒ Autoincrement
- ☐ Sort order:
- ☐ Named constraint:
- ☐ On conflict:

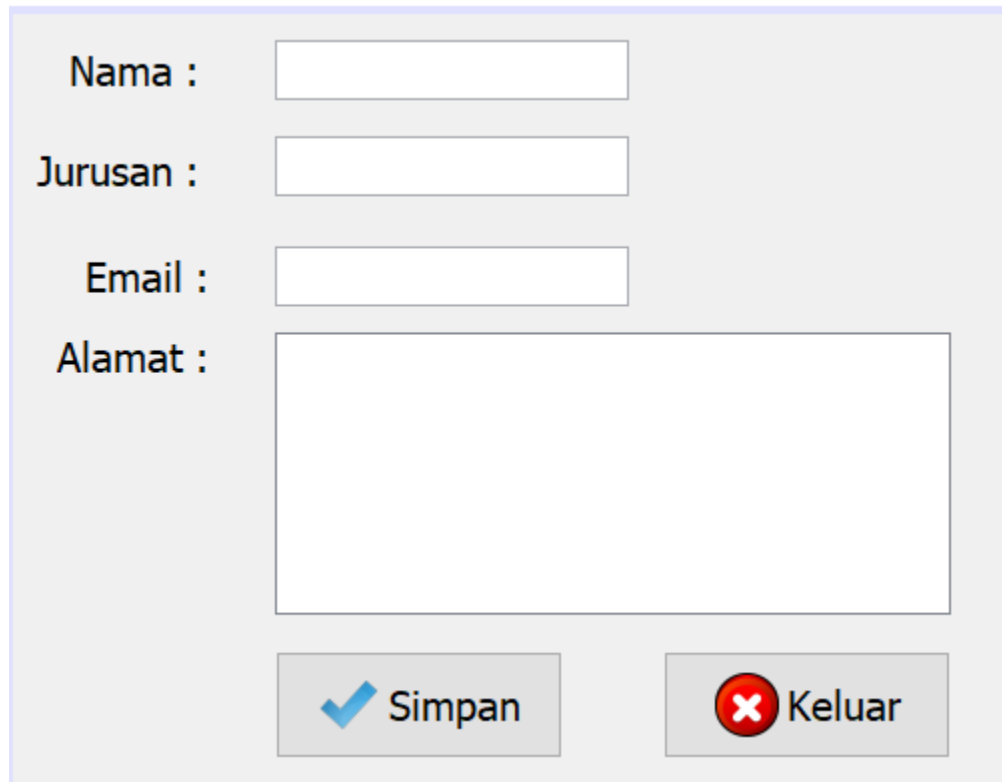
- Setelah lengkap seperti gambar disamping tekan tanda centang
- Setelah diklik database dan tabel sudah berhasil dibuat



- Kembali ke netbeans dan tambahkan java class dengan nama “Connect” dan modifikasi sesuai dengan koding berikut

```
public class Connect {  
    private static Connection con;  
    private static Statement st;  
  
    public static Statement connection(){  
        try {  
            Class.forName("org.sqlite.JDBC");  
            String url = "jdbc:sqlite:kampus.db";  
            con = DriverManager.getConnection(url);  
            st = con.createStatement();  
            System.out.println("koneksi sukses");  
        } catch (Exception ex){  
            System.out.println("koneksi gagal");  
        }  
        return st;  
    }  
}
```

- Tambahkan JFrame Form dengan nama “InputData”





Nama :

Jurusan :

Email :

Alamat :

 Simpan  Keluar

- Buka source di class InputData dan cut coding berikut yang ada di method main

```
/* Create and display the form */  
java.awt.EventQueue.invokeLater(new Runnable() {  
    public void run() {  
        new InputData().setVisible(true);  
    }  
});
```

- setelah dicut, paste di class utama (LatihanSQLITE.java)
- Setelah di cut, hapus method main yang ada di class InputData

```
package latihansqlite;  
  
/**  
 *  
 * @author ikmse  
 */  
public class LatihanSQLITE {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        // TODO code application logic here  
        java.awt.EventQueue.invokeLater(new Runnable() {  
            public void run() {  
                new InputData().setVisible(true);  
            }  
        });  
    }  
}
```

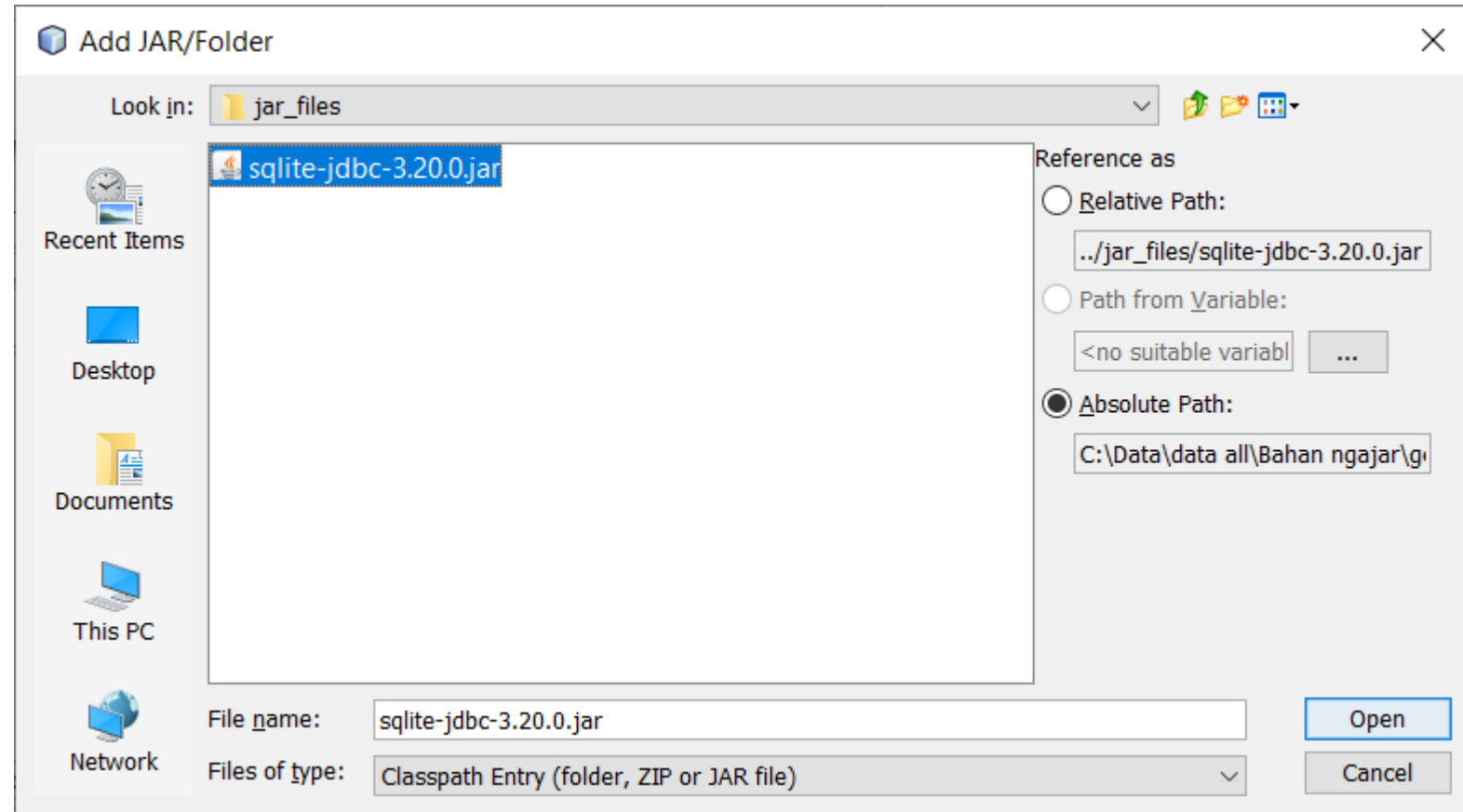
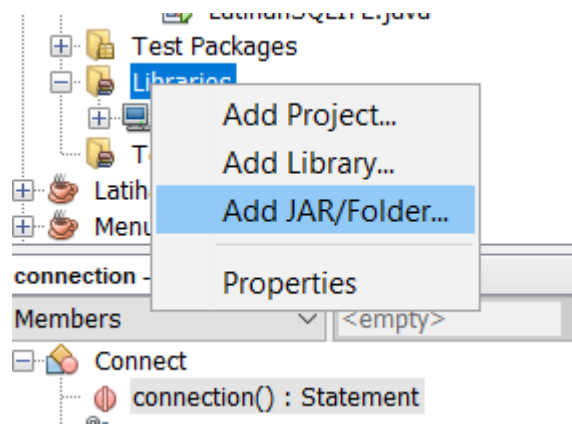
- Kembali ke class InputData dan tambahkan koding berikut (**yang ditambahkan yang dibold**)

```
private static Statement st;  
public InputData() {  
    initComponents();  
    st = Connect.connection();  
}
```

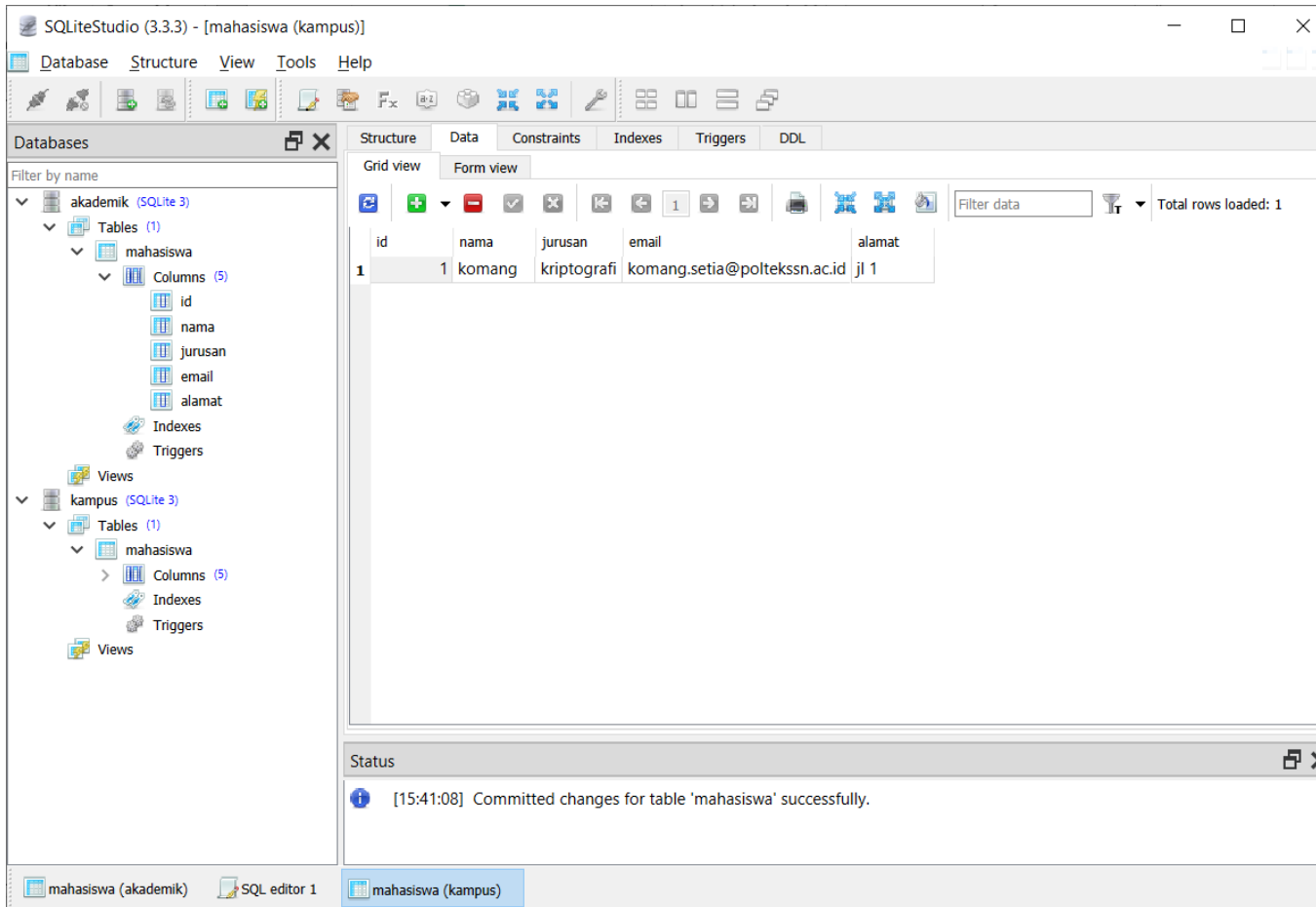
- Klik dua kali di button simpan dan tambahkan koding berikut

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        // TODO add your handling code here:  
        String query = "insert into  
mahasiswa(nama,jurusan,email,alamat) values('" +  
textNama.getText() + "','"  
        + textJurusan.getText() + "','" + textEmail.getText()  
+ "','" + textAlamat.getText() + "')";  
        st.executeUpdate(query);  
        JOptionPane.showMessageDialog(this, "data berhasil  
disimpan");  
    } catch (SQLException ex) {  
  
        Logger.getLogger(InputData.class.getName()).log(Level.SEVERE,  
null, ex);  
    }  
}
```

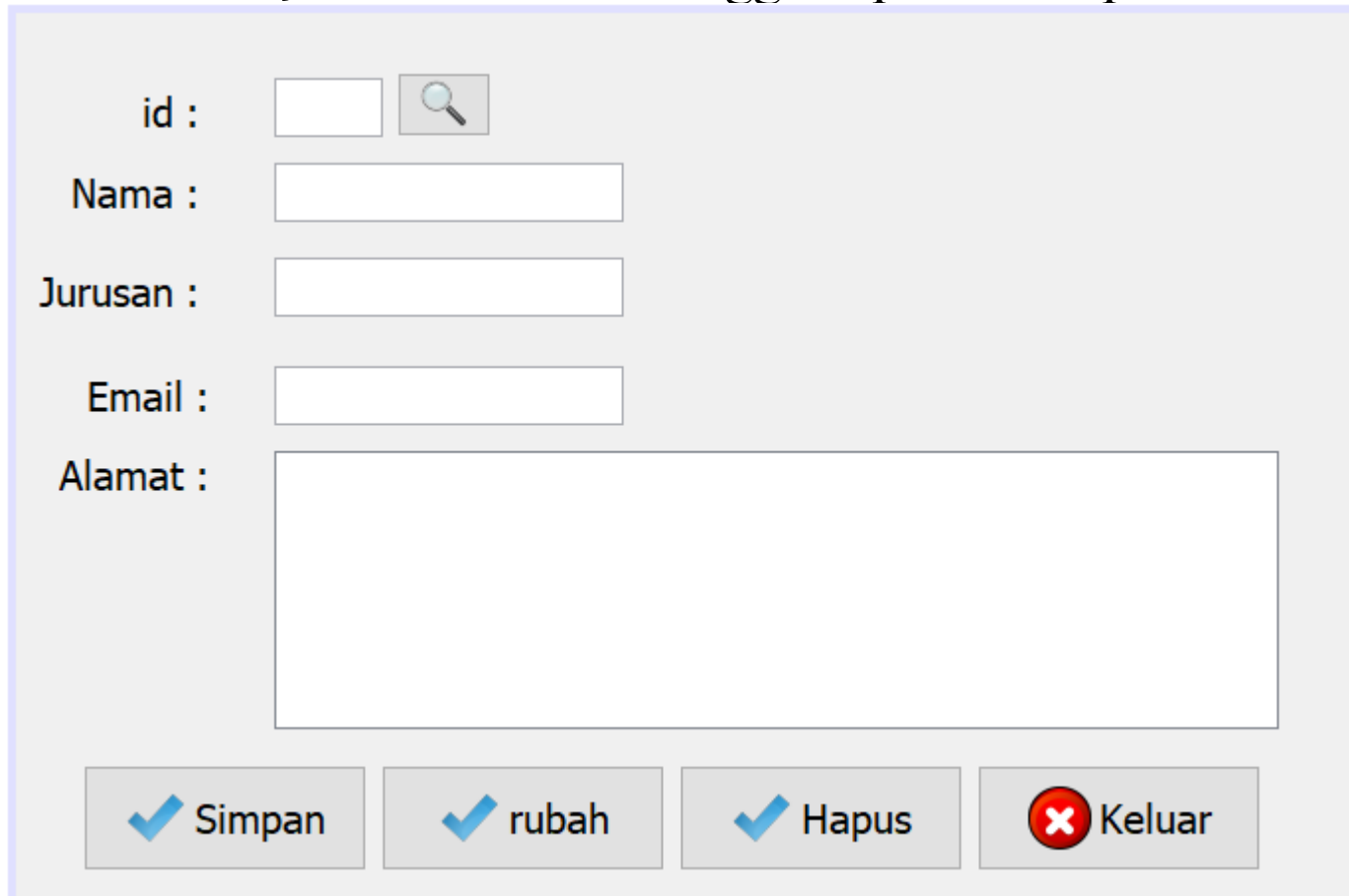

- Terakhir tambahkan library .jar yang sudah didownload sebelumnya




- Jalankan program, jika tidak ada error maka data bisa disimpan ke database



- Setelah berhasil simpan data sekarang kita coba update dan delete data
- Modifikasi jframe form sehingga seperti tampilan berikut



The image shows a Java Swing window with a light gray background. It contains several text input fields and a search button. The labels for the fields are 'id :', 'Nama :', 'Jurusan :', 'Email :', and 'Alamat :'. The 'id' field is a small text box with a magnifying glass icon to its right. The 'Nama', 'Jurusan', and 'Email' fields are standard text boxes. The 'Alamat' field is a larger text area. At the bottom of the window, there are four buttons: 'Simpan' (with a blue checkmark icon), 'rubah' (with a blue checkmark icon), 'Hapus' (with a blue checkmark icon), and 'Keluar' (with a red circle and a white 'X' icon).





id : 

Nama :

Jurusan :

Email :

Alamat :

 Simpan  rubah  Hapus  Keluar

- Masuk ke source tambahkan **ResultSet** seperti koding berikut :

```
private static ResultSet rs = null;
private static Statement st;
public InputData() {
    initComponents();
    st = Connect.connection();
}
```

- Klik 2 kali di button search dan ketikkan program disamping

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

    try {
        String query = "select * from mahasiswa where id = '"
+ textkode.getText() + "'";
        rs = st.executeQuery(query);

        while (rs.next()){
            String nama = rs.getString("nama");
            String jurusan = rs.getString("jurusan");
            String email = rs.getString("email");
            String alamat = rs.getString("alamat");
            textNama.setText(nama);
            textJurusan.setText(jurusan);
            textEmail.setText(email);
            textAlamat.setText(alamat);
        }
    } catch (Exception e){

    }
}
```

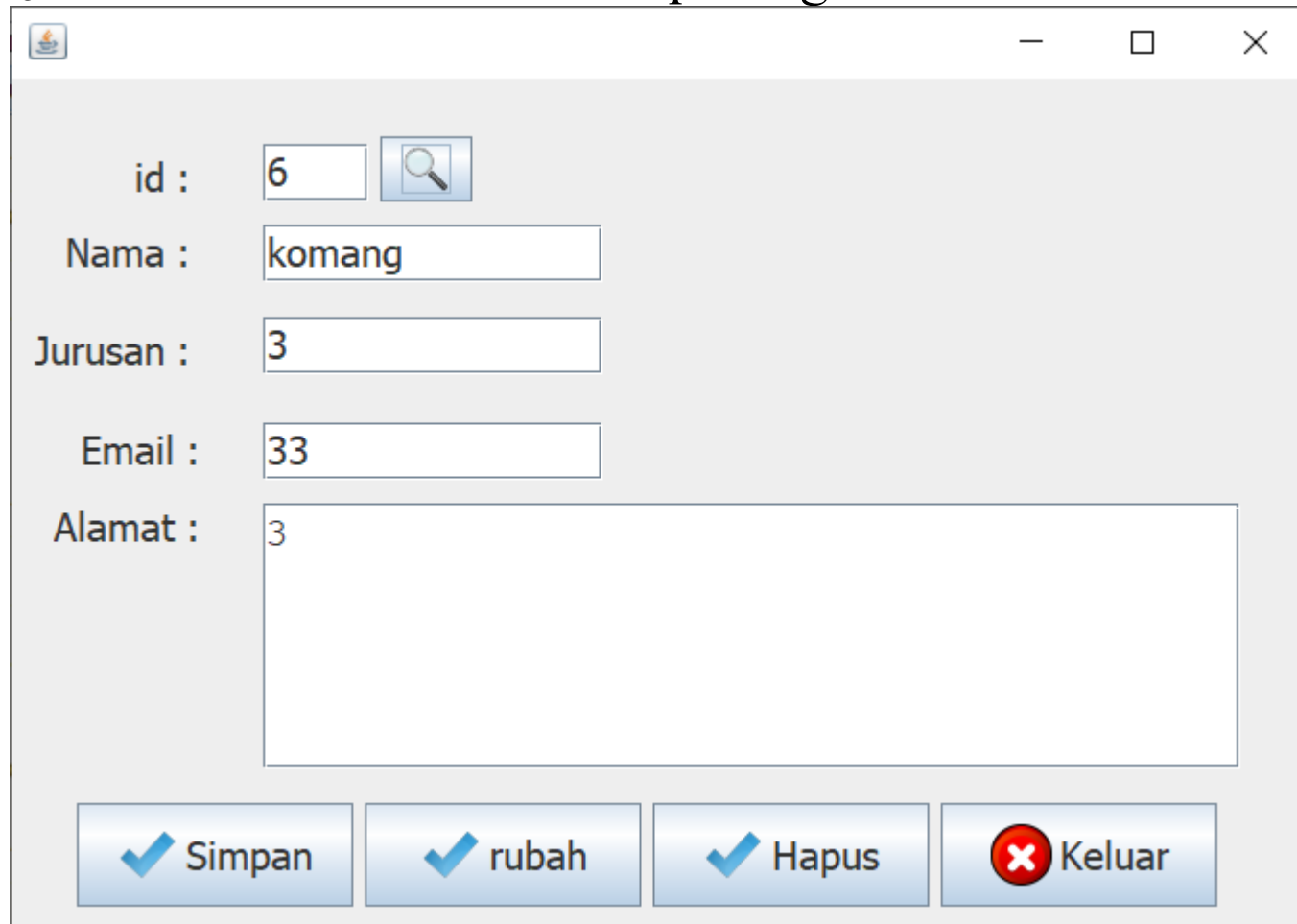
- Untuk sintak update

```
try {  
    // TODO add your handling code here:  
    String query = "update mahasiswa set nama = '"+  
textNama.getText() + "',"  
        + "jurusan = '" + textJurusan.getText() + "',email = '" +  
textEmail.getText() + "',alamat = '"+ textAlamat.getText() + "' "  
        + "where id = '" + textkode.getText() + "'";  
    st.executeUpdate(query);  
    JOptionPane.showMessageDialog(this, "data berhasil dirubah");  
} catch (SQLException ex) {  
  
    Logger.getLogger(InputData.class.getName()).log(Level.SEVERE,  
null, ex);  
}
```

- Sintak delete

```
try {  
    // TODO add your handling code here:  
    JOptionPane.showMessageDialog(this, "delete from mahasiswa  
where id = '" + textkode.getText() + "'");  
    String query = "delete from mahasiswa where id = '" +  
textkode.getText() + "'";  
    st.executeUpdate(query);  
    JOptionPane.showMessageDialog(this, "data berhasil dihapus");  
} catch (SQLException ex) {  
  
    Logger.getLogger(InputData.class.getName()).log(Level.SEVERE,  
null, ex);  
}
```

- Jika tidak ada error akan seperti gambar berikut



The screenshot shows a web application window with a light gray background. At the top, there is a search bar with the label 'id :', a text input field containing the number '6', and a magnifying glass icon. Below the search bar, there are four more input fields: 'Nama : komang', 'Jurusan : 3', 'Email : 33', and 'Alamat : 3'. At the bottom of the window, there are four buttons: 'Simpan' (with a blue checkmark icon), 'rubah' (with a blue checkmark icon), 'Hapus' (with a blue checkmark icon), and 'Keluar' (with a red 'X' icon).

- Ketika diketikkan angka 6 dan tekan button search, maka akan muncul hasil seperti yang terdapat di database
- Setelah itu untuk merubah data tinggal rubah text yang terdapat diaplikasi dan klik button rubah
- Untuk menghapus tinggal klik tombol hapus

Tugas

1. Deskripsikan hasil praktikum yang dilakukan ke dalam laporan!