



# KOMUNIKASI APLIKASI DENGAN SERVER BASIS DATA DAN MENGOLAH DATA MELALUI APLIKASI DESKTOP



# Pendahuluan

JDBC API (Java Database Connectivity Application Programming Interface ) adalah Java API yang berfungsi melakukan pengaksesan terhadap data tabular dalam basis data (database). JDBC Api memungkinkan kita melakukan 3 hal :

1. Membangun Koneksi
2. Mengirim Pernyataan SQL
3. Memproses Hasil.

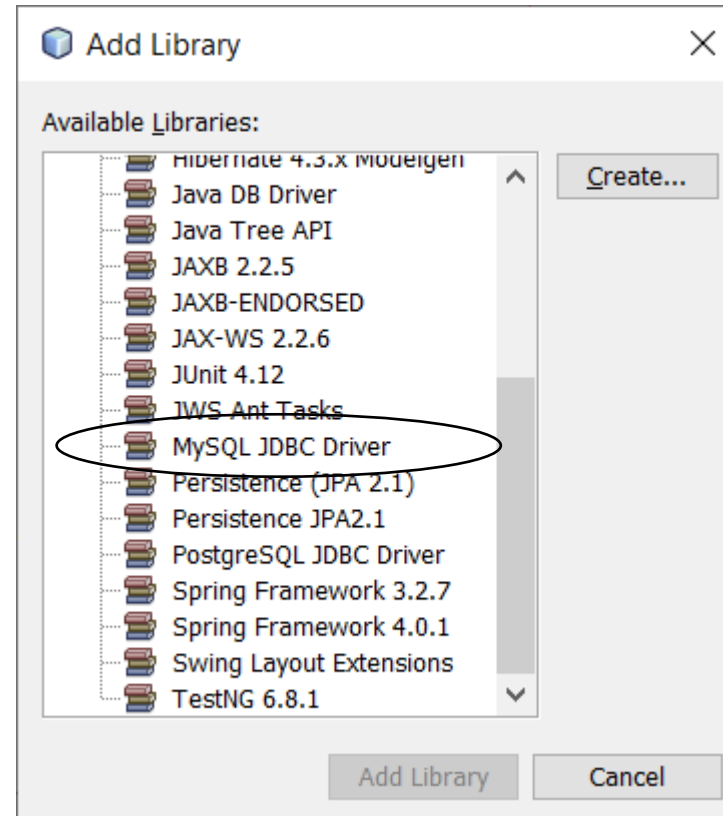
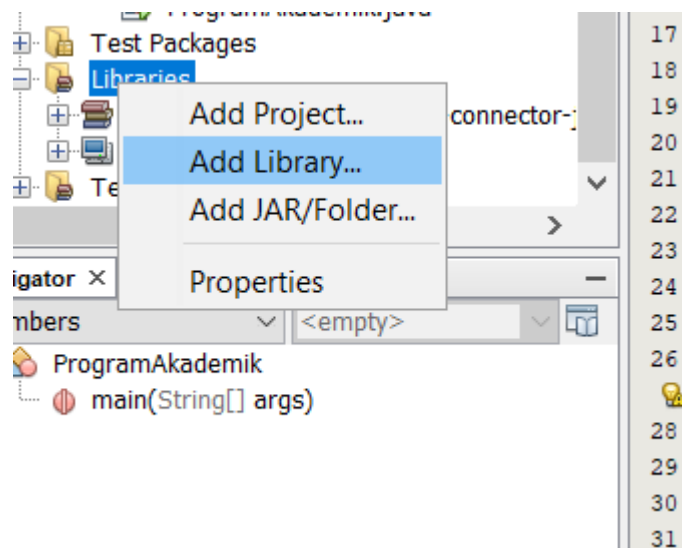
- Buat database baru dengan nama “akademik” dan table “mahasiswa”, dengan struktur seperti dibawah

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
nama	varchar(30)	NO		NULL	
jurusan	varchar(20)	NO		NULL	
email	varchar(20)	NO		NULL	
alamat	text	NO		NULL	

5 rows in set (0.01 sec)

# Praktikum

- Buat project baru dengan nama “ProgramAkademik”
- Tambahkan library MYSQL JDBC Driver dengan cara klik kanan di libraries project yang dibuat



# Praktikum

- Setelah ditambahkan buat class baru dengan nama “Koneksi” dan tambahkan kode seperti berikut

```
public class Koneksi {  
    private static Connection con;  
    public static Connection getKoneksi () {  
        try {  
            String url = "jdbc:mysql://localhost/akademik";  
            String user = "root";  
            String password = "";  
            DriverManager.registerDriver(new com.mysql.jdbc.Driver());  
            con = DriverManager.getConnection(url, user, password);  
            System.out.println("koneksi sukses");  
        } catch (SQLException t) {  
            System.out.println("Error Membuat Koneksi");  
        }  
        return con;  
    }  
}
```

# Praktikum

- Buat form aplikasi dengan tampilan seperti berikut
- Untuk TextField ID buat enable false

The screenshot shows a Java Swing window titled "Input Data Mahasiswa". The window contains a form with the following fields:

- id :
- Nama :
- Jurusan :
- Email :
- Alamat :

Below the form are four buttons:

- Simpan
- rubah
- Hapus
- Keluar

At the bottom of the window is a table with the following columns:

id	nama	jurusan	email	alamat
----	------	---------	-------	--------

- Sama seperti latihan sebelumnya copy paste kode berikut ke menu utama “ProgramAkademik.java”

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

- Hal pertama yang harus dilakukan adalah load data ke table. Buat method loadData dengan kode seperti berikut

```
private void loadData() {  
    try {  
        Connection c = Koneksi.getKoneksi();  
        Statement s = c.createStatement();  
  
        String sql = "SELECT * FROM MAHASISWA";  
        ResultSet r = s.executeQuery(sql);  
        while (r.next()) {  
            // lakukan penelusuran baris  
            String id = r.getString("id");  
            String nama = r.getString("nama");  
            String jurusan = r.getString("jurusan");  
            String email = r.getString("email");  
            String alamat = r.getString("alamat");  
            String tbData[] = {id,nama,jurusan,email,alamat};  
        }  
    }  
}
```

```
DefaultTableModel tblModel =  
(DefaultTableModel)tabelMahasiswa.getModel();  
        tblModel.addRow(tbData);  
    }  
    r.close();  
    s.close();  
} catch (SQLException e) {  
    System.out.println("Terjadi Kesalahan");  
}  
}
```



# Koding simpan data

- Ketikkan metod loadData di konstruktor
- Kemudian klik 2 kali di button simpan tambahkan koding berikut

```
private void jSimpanActionPerformed(java.awt.event.ActionEvent evt) {  
    String nama = textNama.getText();  
    String jurusan = textJurusan.getText();  
    String email = textEmail.getText();  
    String alamat = textAlamat.getText();  
  
    try{  
        Connection c = Koneksi.getKoneksi();  
        String sql = "INSERT INTO MAHASISWA(nama,jurusan,email,alamat) VALUES (?, ?, ?, ?)";  
        PreparedStatement p = c.prepareStatement(sql);  
        p.setString(1, nama);  
        p.setString(2, jurusan);  
        p.setString(3, email);  
        p.setString(4, alamat);  
        p.executeUpdate();  
        p.close();  
    } catch(SQLException e) {  
        System.out.println("Terjadi Kesalahan");  
    }  
    refreshTabel();  
    loadData();  
}
```

# Praktikum

- Jalankan program dan lakukan proses simpan
- Agar jTable langsung terupdate tambahkan method loadData
- Seperti latihan sebelumnya tambahkan method refreshData

```
private void refreshTabel() {  
    DefaultTableModel model=(DefaultTableModel)  
tabelMahasiswa.getModel();  
    while(model.getRowCount()>0){  
        model.setRowCount(0);  
    }  
}
```

Input Data Mahasiswa




Id :

Nama :

Jurusan :

Email :

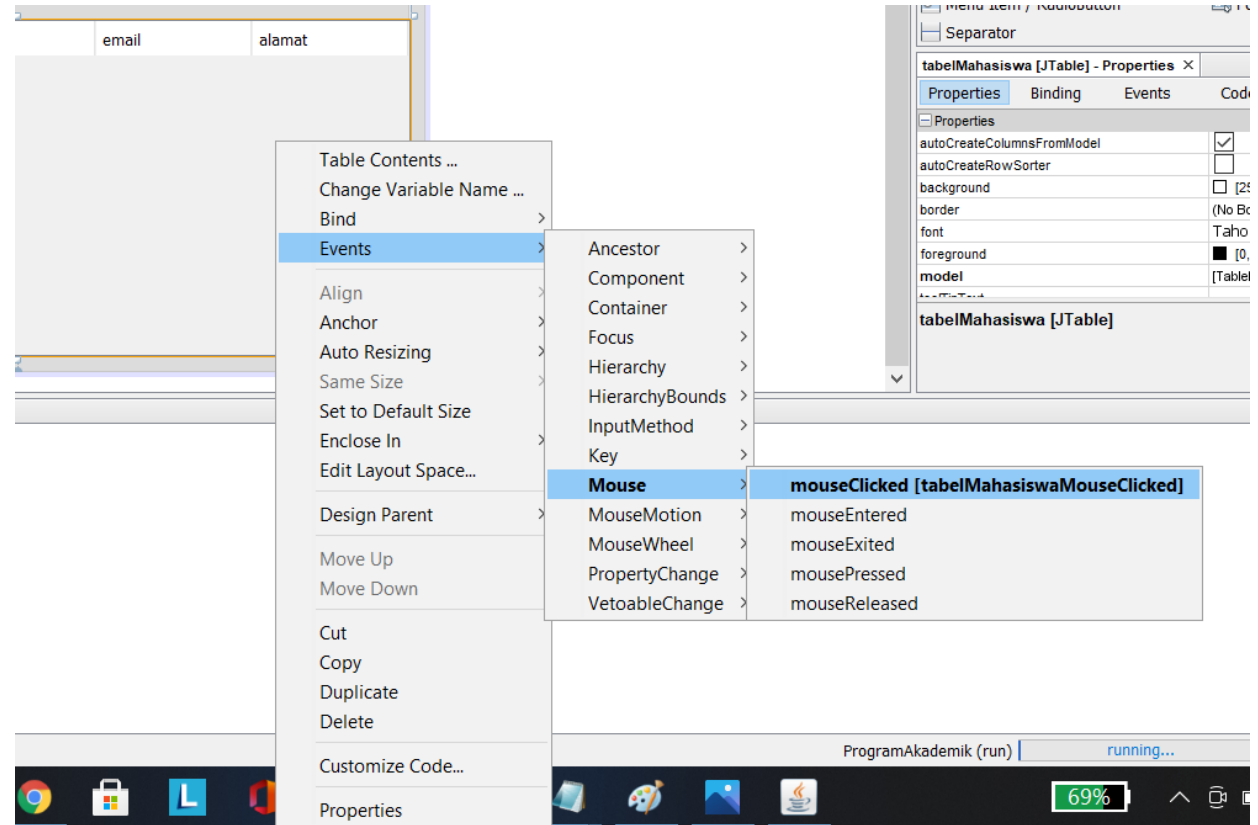
Alamat :

 Simpan  rubah  Hapus  Keluar

id	nama	jurusan	email	alamat
1	komang	Kriptografi	komang@poltekssn....	bogor

# Praktikum

- Untuk melakukan proses update, tambahkan koding berikut agar ketika diklik record di table, langsung muncul di text
- Klik kanan di jTablel dan pilih mouseClicked seperti gambar disamping



- Tambahkan kode berikut

```
private void tabelMahasiswaMouseClicked(java.awt.event.MouseEvent evt) {  
    // TODO add your handling code here:  
    int index = tabelMahasiswa.getSelectedRow();  
    TableModel model = tabelMahasiswa.getModel();  
  
    textkode.setText(model.getValueAt(index, 0).toString());  
    textNama.setText(model.getValueAt(index, 1).toString());  
    textJurusan.setText(model.getValueAt(index, 2).toString());  
    textEmail.setText(model.getValueAt(index, 3).toString());  
    textAlamat.setText(model.getValueAt(index, 4).toString());  
  
}
```

# Koding update data

- Buat kode seperti berikut

```
private void
jRubahActionPerformed(java.awt.event.ActionEvent evt)
{
    // TODO add your handling code here:
    String nama = textNama.getText();
    String jurusan = textJurusan.getText();
    String email = textEmail.getText();
    String alamat = textAlamat.getText();
    String id = textkode.getText();

    try{
        Connection c = Koneksi.getKoneksi();
        String sql = "update mahasiswa set nama =
        ?,jurusan = ?,email = ?,alamat = ? where id = ?";
```

```
        PreparedStatement p = c.prepareStatement(sql);
        p.setString(1, nama);
        p.setString(2, jurusan);
        p.setString(3, email);
        p.setString(4, alamat);
        p.setString(5, id);
        p.executeUpdate();
        p.close();
        JOptionPane.showMessageDialog(this, "data berhasil dirubah");
        jRubah.setEnabled(false);
        jHapus.setEnabled(false);
        jSimpan.setEnabled(true);
    } catch(SQLException e) {
        System.out.println("Terjadi Kesalahan");
    }
    refreshTabel();
    loadData();
    refreshText();
}
```

# Koding delete

- Silahkan tambahkan kode delete

# Latihan

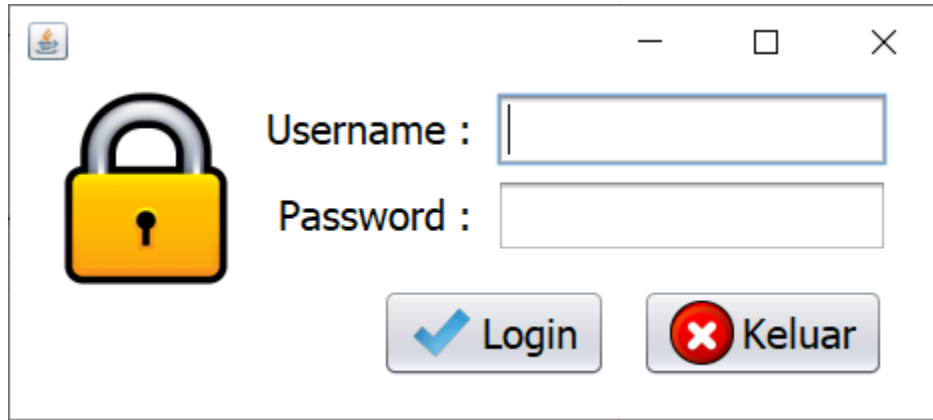
- Buat table baru dengan nama user

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
user	varchar(30)	NO		NULL	
pass	varchar(30)	NO		NULL	

3 rows in set (0.02 sec)



- Isikan data di table tersebut

- Buat form login



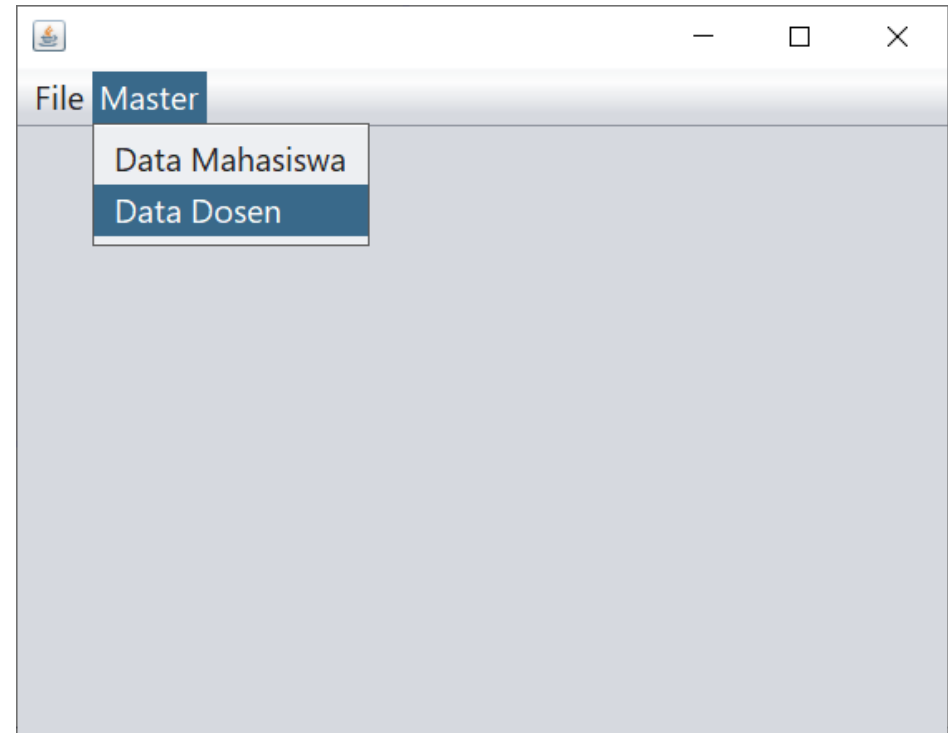
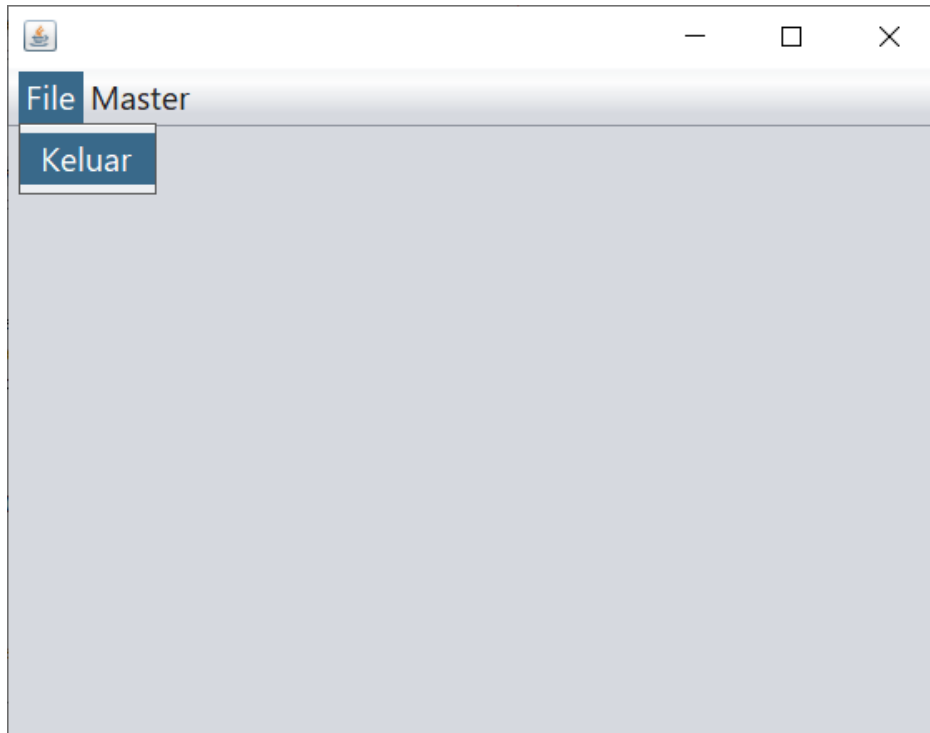
Username :

Password :

 Login  Keluar



- User password yang diinputkan ketika sesuai dengan isi ditabel akan menuju menu utama



- Menu data mahasiswa, silahkan langsung dipanggil inputData yang sudah dibuat sebelumnya
- Menu data dosen silahkan tambahkan tabel dosen dan buat form input data dosen

# Tugas

1. Deskripsikan hasil praktikum yang dilakukan ke dalam laporan!