

Step by Step

Pada tahap 1 :

1. Instal aplikasi yang akan kita gunakan sebagai server lokal. Pada praktikum ini, kita akan menggunakan XAMPP sebagai server lokal
2. kemudian XAMPP di jalankan Apache dan MySQL.
3. Buka phpmyadmin, kemudian buat basis data baru dengan nama.
4. Buat tabel baru dalam database dengan detail sebagai berikut:

A. Nama tabel: Arsip Dokumen

B. Struktur tabel:

Tahap 2 :

1. Buka aplikasi Netbeans , kemudian buat project baru dengan nama UTS
2. Setelah anda membuat project baru, package baru package juga
3. Buka lokasi project anda, kemudian buat folder baru dengan nama lib.
4. Copy-Paste driver yang telah anda download (pilih salah satu, mysql-connector-java5.1.47.jar atau mysql-connector-java-5.1.47-bin.jar) ke dalam folder lib.
5. Tambahkan library yang telah anda siapkan (dalam folder lib) ke dalam project.

Tahap 3:

Buatlah desain dengan detail sebagai berikut:

A. Type: JFrame Form

B. Nama Kelas/Class Name: JavaMySQL

C. Desainlah tampilan di sini saya tidak perjelas karena waktu sudah mepet.

Tahap 4:

Membuat Kelas KoneksiDB Buatlah kelas baru dengan detail sebagai berikut:

A. Type: Java Class

B. Nama Kelas/Class Name: KoneksiDB

C. Ubah kode program pada kelas KoneksiDB, sehingga menjadi seperti di bawah ini.

```

12  * @author Lenovo
13  */
14  public class KoneksiDB {
15      public static Connection sambungDB() {
16          String JDBC_DRIVER_CLASS = "com.mysql.cj.jdbc.Driver";
17          String username = "root";
18          String host = "localhost";
19          String port = "3306";
20          String password = "";
21          String dbName = "arsip dokumen";
22          String url = "jdbc:mysql://" + host + ":" + port + "/" + dbName + "?useTimezone=true&serverTimezone=UTC";
23          Connection kon = null;
24          try {
25              Class.forName(JDBC_DRIVER_CLASS).newInstance();
26              kon=
27              DriverManager.getConnection(url,username,password);
28              System.out.println("Koneksi: Sukses!");
29              return kon;
30          } catch (ClassNotFoundException | IllegalAccessException |
31                  InstantiationException | SQLException e) {
32              System.err.println("Error: " + e.getMessage());
33              System.exit(0);
34          }
35          return null;
36      }
37  }
38

```

D. Pastikan bahwa username, password, host, port, dan nama database telah sesuai dengan konfigurasi pada DBMS

Tahap 5: Menguji Koneksi ke Database

1. Buka kelas JavaMySQL.java, pilih Tab Source.
2. Panggil method sambungDB yang ada di kelas KoneksiDB di dalam konstruktor

Tahap 6: Deklarasi Variabel Global

1. Buka kelas JavaMySQL.java, kemudian buka Tab Source.
2. Lakukan deklarasi variabel yang dibutuhkan:

```
public class JavaMySQL extends javax.swing.JFrame {  
    int idBaris=0;  
    String role;  
    DefaultTableModel model;
```

Tahap 7: Pembuatan Method-Method

1. Buka kelas JavaMySQL.java, kemudian buka Tab Source.

2. Tambahkan method-method di bawah ini:

```
private void aturModelTabel(){
    Object[] kolom = {"no", "Kode Dok", "Nama Dok", "Kategori Dok", "Lokasi Dok", "Deskripsi Dok", "Tanggal"}
    model = new DefaultTableModel(kolom,0) {
        boolean[] canEdit = new boolean [] {
            false, false, false, false, false, false, false
        };
        @Override
        public boolean isCellEditable(int rowIndex, int columnIndex) {
            return canEdit [columnIndex];
        }
    };
    tbldokumen.setModel(model);
    tbldokumen.setRowHeight(20);
    tbldokumen.getColumnModel().getColumn(0).setMinWidth(0);
    tbldokumen.getColumnModel().getColumn(0).setMaxWidth(0);
}

private void showForm(boolean b){
    areaSplit.setDividerLocation(0.3);
    areaSplit.getLeftComponent().setVisible(b);
}

private void resetForm(){
    tbldokumen.clearSelection();
    txtkode.setText("");
    txtnama.setText("");
    cmbkategori.setSelectedIndex(0);
    txtlokasi.setText("");
    txtdesk.setText("");
    txttanggal.setText("");
    txtkode.requestFocus();
}
```

```

private void kategori() {
    cmbkategori.removeAllItems();
    cmbkategori.addItem("pilih jenis surat");
    cmbkategori.addItem("Resmi");
    cmbkategori.addItem("Umum");
    cmbkategori.addItem("Pribadi");
}

private void showData(String key) {
    model.getDataVector().removeAllElements();
    String where = "";
    if (!key.isEmpty()) {
        where += "WHERE Kode_Dokumen LIKE '%" + key + "%' "
            + "OR nama_dk LIKE '%" + key + "%' "
            + "OR kategori_dk LIKE '%" + key + "%' "
            + "OR lokasi_dk LIKE '%" + key + "%' "
            + "OR deskripsi_dk LIKE '%" + key + "%' "
            + "OR tanggal LIKE '%" + key + "%' ";
    }
    String sql = "SELECT * FROM dokumen " + where;
    Connection con;
    Statement st;
    ResultSet rs;
    int baris = 0;
    try {
        con = KoneksiDB.sambungDB();
        st = con.createStatement();
        rs = st.executeQuery(sql);
        while (rs.next()) {
            Object id = rs.getInt(1);
            Object kode = rs.getString(2);

```

```

        Object id = rs.getString(1);
        Object kode = rs.getString(2);
        Object nama = rs.getString(3);
        Object kategori = rs.getString(4);
        Object lokasi = rs.getString(5);
        Object deskripsi = rs.getString(6);
        Object tanggal = rs.getString(7);
        Object[] data = {id, kode, nama, kategori, lokasi, deskripsi, tanggal};
        model.insertRow(baris, data);
        baris++;
    }
    st.close();
    con.close();
    tblDokumen.revalidate();
    tblDokumen.repaint();
} catch (SQLException e) {
    System.err.println("showData(): " + e.getMessage());
}
}

private void resetView() {
    resetForm();
    showForm(false);
    showData("");
    btnHapus.setEnabled(false);
    idBaris = 0;
}

private void pilihData(String n) {
    btnHapus.setEnabled(true);

    String sql = "SELECT * FROM dokumen WHERE id='" + n + "'";
    Connection con;

```

```

Statement st;
ResultSet rs;
try {
    con = KoneksiDB.sambungDB();
    st = con.createStatement();
    rs = st.executeQuery(sql);
    while (rs.next()) {
        int id = rs.getInt(1);
        String Kode_Dokumen = rs.getString(2);
        String nama_dk = rs.getString(3);
        Object kategori_dk = rs.getString(4);
        String lokasi_dk = rs.getString(5);
        String deskripsi_dk = rs.getString(6);
        String tanggal = rs.getString(7);
        idBaris = id;
        txtkode.setText(Kode_Dokumen);
        txtnama.setText(nama_dk);
        cmbkategori.setSelectedItem(kategori_dk);
        txtlokasi.setText(lokasi_dk);
        txtdesk.setText(deskripsi_dk);
        txttanggal.setText(tanggal);
    }
    st.close();
    con.close();
    showForm(true);
} catch (SQLException e) {
    System.err.println("pilihData(): " + e.getMessage());
}
}

```

```

private void simpanData() {
    String kode = txtkode.getText();
    String nama = txtnama.getText();
    int kategori = cmbkategori.getSelectedIndex();
    String lokasi = txtlokasi.getText();
    String desk = txtdesk.getText();
    String tanggal = txttanggal.getText();
    if (kode.isEmpty() || nama.isEmpty() || kategori == 0 || lokasi.isEmpty()
        || desk.isEmpty() || tanggal.isEmpty()) {
        JOptionPane.showMessageDialog(this, "Mohon lengkapi data!");
    } else {
        String kategori1 = cmbkategori.getSelectedItem().toString();

        String sql
            = "INSERT INTO dokumen (Kode_Dokumen,nama_dk,kategori_dk,"
            + "lokasi_dk,deskripsi_dk,tanggal) "
            + "VALUES (" + kode + "\",\"" + nama + "\",\"
            + "\"" + kategori1 + "\",\"" + lokasi + "\",\"" + desk
            + "\",\"" + tanggal + "\"))";

        Connection con;
        Statement st;
        try {
            con = KoneksiDB.sambungDB();
            st = con.createStatement();
            st.executeUpdate(sql);
            st.close();
            con.close();
        }
    }
}

```



```

        con.close();

        resetView();
        JOptionPane.showMessageDialog(this, "Data telah disimpan!");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

private void ubahData() {
    String kode = txtkode.getText();
    String nama = txtnama.getText();
    int kategori = cmbkategori.getSelectedIndex();
    String lokasi = txtlokasi.getText();
    String desk = txtdesk.getText();
    String tanggal = txttanggal.getText();
    if (kode.isEmpty() || nama.isEmpty() || kategori == 0 || lokasi.isEmpty()
        || desk.isEmpty() || tanggal.isEmpty()) {
        JOptionPane.showMessageDialog(this, "data Lengkapi!");
    } else {
        String kategori1 = cmbkategori.getSelectedItem().toString();

        String sql
            = "UPDATE dokumen "
            + "SET Kode_Dokumen=\"" + kode + "\", "
            + "nama_dk=\"" + nama + "\", "
            + "kategori_dk=\"" + kategori1 + "\", "
            + "lokasi_dk=\"" + lokasi + "\", "
            + "deskripsi_dk=\"" + desk + "\", "
            + "tanggal=\"" + tanggal + "\" where id = \"" + idBaris + "\"";
        Connection con;
    }
}

```

```

        con.close();

        resetView();
        JOptionPane.showMessageDialog(this, "Data telah diubah!");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}

private void hapusData(int baris){
    Connection con;
    Statement st;
    try {
        con = KoneksiDB.sambungDB();
        st = con.createStatement();
        st.executeUpdate("DELETE FROM dokumen WHERE id=" + baris);
        st.close();
        con.close();

        resetView();
        JOptionPane.showMessageDialog(this, "Data telah dihapus");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}
}

```

Tahap 8: Menambahkan event pada Komponen

1. Buka kelas JavaMySQL.java, kemudian buka Tab Design
2. Disini saya jadikan satu biar lebih mudah.

```
@SuppressWarnings("unchecked")
Generated Code

private void btnTambahActionPerformed(java.awt.event.ActionEvent evt) {
    role = "Tambah";
    btnSimpan.setText("Simpan");
    idBaris = 0;
    resetForm();
    showForm(true);
    btnHapus.setEnabled(false);
}

private void btnHapusActionPerformed(java.awt.event.ActionEvent evt) {
    if (idBaris == 0) {
        JOptionPane.showMessageDialog(this, "Pilih data yang akan"
            + " dihapus");
    } else {
        hapusData(idBaris);
    }
}

private void btnSimpanActionPerformed(java.awt.event.ActionEvent evt) {
    if (role.equals("Tambah")) {
        simpanData();
    } else if (role.equals("Ubah")) {
        ubahData();
    }
}
```

```

private void btnTutupActionPerformed(java.awt.event.ActionEvent evt) {
    resetForm();
    showForm(false);
    btnHapus.setEnabled(false);
    idBaris = 0;
}

private void txtCariKeyReleased(java.awt.event.KeyEvent evt) {
    String key = txtCari.getText();
    showData(key);
}

private void formComponentResized(java.awt.event.ComponentEvent evt) {
    areaSplit.setDividerLocation(0.3);
}

private void tbldokumenMouseClicked(java.awt.event.MouseEvent evt) {
    role = "Ubah";
    int row = tbldokumen.getRowCount();
    if (row > 0) {
        int sel = tbldokumen.getSelectedRow();
        if (sel != -1) {
            pilihData(tbldokumen.getValueAt(sel, 0).toString());
            btnSimpan.setText("UBAH DATA");
        }
    }
}
}

```

```

        con.close();

        resetView();
        JOptionPane.showMessageDialog(this, "Data telah diubah!");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}




private void hapusData(int baris) {
    Connection con;
    Statement st;
    try {
        con = KoneksiDB.sambungDB();
        st = con.createStatement();
        st.executeUpdate("DELETE FROM dokumen WHERE id=" + baris);
        st.close();
        con.close();

        resetView();
        JOptionPane.showMessageDialog(this, "Data telah dihapus");
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(this, e.getMessage());
    }
}
}

```

•

[illegible]

Kode Dok	Nama Dok	Kategori Dok	Lokasi Dok	Deskripsi Dok
1	1	Resmi	12	1

Tambah Data

Hapus Data

Cari

Kode Dok

Nama Dok

Kategori Dok

pilih jenis surat ▾

Lokasi Dok

Deskripsi

Tanggal

Tutup

Simpan

Kode Dok	Nama Dok	Kategori Dok	Lokasi Dok	Deskripsi Dok	Tanggal
1	1	Resmi	12	1	1

Tambah Data

Hapus Data

Cari

Kode Dok	Nama Dok	Kategori Dok	Lokasi Dok	Deskripsi Dok	Tanggal
1	1	Resmi	12	1	1

Message

Data telah dihapus

OK



Tambah Data

Hapus Data

Cari

Kode Dok	Nama Dok	Kategori Dok	Lokasi Dok	Deskripsi Dok	
1	1	Resmi	12	1	1