LAPORAN TUGAS KULIAH PEMROGRAMAN BERORIENTASI OBJEK



Dosen Pengajar:

Bayu Adhi Nugroho, Ph.D

Disusun oleh:

ARRIO EKA FIRMANSYAH (09020622021)

UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA

FAKULTAS SAINS DAN TEKNOLOGI

SISTEM INFORMASI

SURABAYA

2023

TUGAS: Membuat java GUI yang berisi tampilan data, insert data, ubah data, dan hapus data.

Pilihan menu:

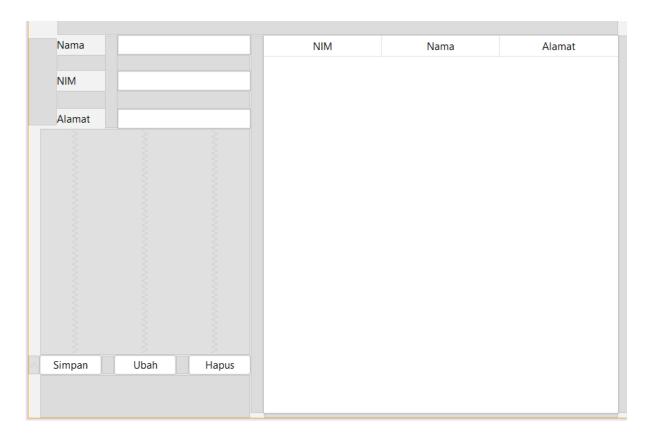
langkah 1 yaitu dengan membuat class mahasiswa, yang berisi data data yang akan digunakan.

```
public class Mahasiswa {
     private String nama;
     private String nim;
     private String alamat;
   /**

* @return the nama

*/
戸
    public String getNama() {
      return nama;
    * @param nama the nama to set
   public void setNama(String nama) {
     this.nama = nama;
* @return the nim
*/
口
    public String getNim() {
      return nim;
口
     * @param nim the nim to set */
口
     public void setNim(String nim) {
      this.nim = nim;
     * @return the alamat
*/
戸
     public String getAlamat() {
     return alamat;
```

kemudian buat class java interface, buat desain sesuai kebutuhan. Dibawah ini adalah desain yang saya buat



kemudian klik source, kemudian isi source code sesuai dengan keguanaan tombol tombol, dan hal-hal yang dibutuhkan

untuk menampilkan data pada setiap eksekusi

```
private void tampil(Connection conn) {
    dataMahasiswa.clear();
    try {
        String sql = "select * from mahasiswa";
       Statement st = conn.createStatement();
        ResultSet rs = st.executeQuery(sql);
        while (rs.next()) {
           Mahasiswa data = new Mahasiswa();
            data.setNim(nim:String.valueOf(obj:rs.getObject(columnIndex:1)));
            data.setNama( nama:String.valueOf( obj:rs.getObject( columnIndex: 2)));
            data.setAlamat( alamat: String.valueOf( obj:rs.getObject( columnIndex: 3) ));
            dataMahasiswa.add(e:data);
        DefaultTableModel model = (DefaultTableModel) jTableMahasiswa.getModel();
        model.setRowCount(rowCount:0);
        for (Mahasiswa data : dataMahasiswa) {
            Object[] baris = new Object[3];
            baris[0] = data.getNim();
            baris[1] = data.getNama();
            baris[2] = data.getAlamat();
            model.addRow(rowData:baris);
    } catch (SQLException ex) {
        Logger.getLogger(name:Tampilan.class.getName()).log(level:Level.SEVERE, msg:null, thrown:ex);
```

method sintaks sesuai kegunaan (INSERT, UPDATE, DELETE)

```
private int masukkanData(Connection conn, String nim, String nama, String alamat) throws SQLException {
   PreparedStatement pst = conn.prepareStatement( sql: "INSERT INTO MAHASISWA (nim, nama, alamat) VALUES(?,?,?)");
   pst.setString(parameterIndex:1, x:nim);
   pst.setString(parameterIndex: 2, x:nama);
   pst.setString(parameterIndex: 3, x:alamat);
   return pst.executeUpdate();
private int ubahData (Connection conn, String nama, String alamat, String nim) throws SQLException {
   PreparedStatement pst = conn.prepareStatement(sql:"UPDATE MAHASISWA SET mama = 2, alamat = 2 WHERE nim = 2");
   pst.setString(parameterIndex:1, x:nama);
   pst.setString(parameterIndex: 2, x:alamat);
   pst.setString(parameterIndex: 3, x:nim);
   return pst.executeUpdate();
private int hapusData(Connection conn, String nim) throws SQLException {
   PreparedStatement pst = conn.prepareStatement(sql:"DELETE FROM MAHASISWA WHERE NIM = ? ");
   pst.setString(parameterIndex:1, x:nim);
   return pst.executeUpdate();
```

source code pada button simpan (INSERT). yang berisi nama, nim, dan alamat

```
private void jButtonSimpanActionPerformed(java.awt.event.ActionEvent evt) {
         TODO add your handling code her
      String nim = jTextFieldNIM.getText().trim();
      String nama = jTextFieldNama.getText();
      String alamat = jTextFieldAlamat.getText();
      if (!nim.isEmpty() && !nama.isEmpty()) {
              // TODO add your handling code here:
              Connection conn = DriverManager.getConnection(url:"jdbc:postgresql://localhost:5432/postgres", user:"postgres", password:"lllll");
              int k = masukkanData(conn, nim, nama, alamat);
              if (k > 0) {
                 tampil(conn);
                  this.peringatan(pesan: "Simpan Berhasil");
                this.peringatan(pesan: "Simpan Gagal");
          } catch (SQLException ex) {
             Logger.getLogger( name: Tampilan.class.getName()).log( level: Level.SEVERE, msg: null, thrown: ex);
      } else {
         this.peringatan(pesan: "Input Data Mahasiswa Gagal");
      jTextFieldNIM.setText( t:"");
      iTextFieldNama.setText( t: "");
      jTextFieldAlamat.setText(t:"");
```

source code pada button ubah (UPDATE). yang berisi nama, nim, dan alamat

source code pada button hapus (DELETE). yang berisi nama, nim, dan alamat

OUTPUT:

