

LAPORAN TUGAS KULIAH
PEMROGAMAN BEORIENTASI OBYEK
Pertemuan Kelima (19 September 2023)



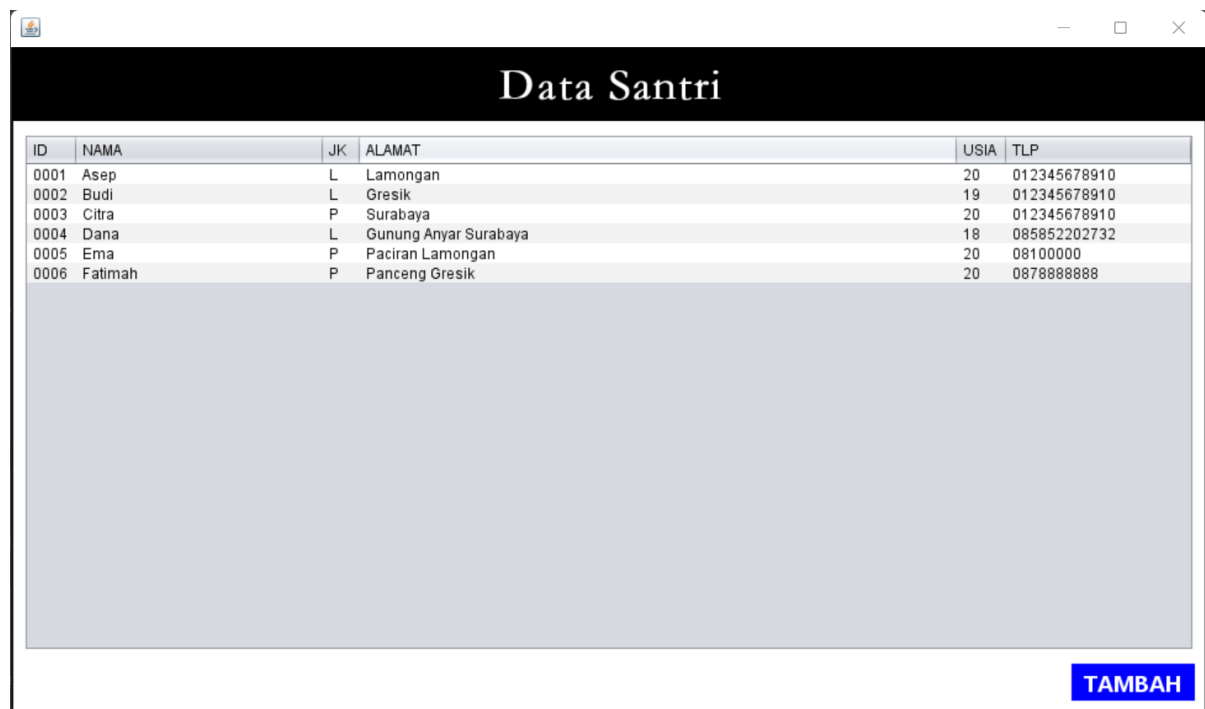
UIN SUNAN AMPEL
S U R A B A Y A

Dosen Pembimbing
Bayu Adi Nugroho, PH.D

Disusun Oleh:
Nur Fatiq
09040622071

TAHUN PEMBELAJARAN 2023/2024
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
FAKULTAS SAINS DAN TEKNOLOGI
SISTEM INFORMASI
2023/2024

- **Tampilan awal GUI**



Berikut adalah tampilan awal ketika program dirun. Menampilkan sebuah tabel dengan data yang diambil dari database, dan sebuah tombol dengan label tambah.

- **Konstruktor**

```
/**
 *
 * @author fatiq
 */
public class Main extends javax.swing.JFrame {

    /**
     * Creates new form Main
     */
    private final String url = "jdbc:postgresql://localhost:5432/PBO";
    private final String user = "postgres";
    private final String pass = "5127";
    private Connection conn = null;

    public Main() throws SQLException {
        initComponents();
        conn = DriverManager.getConnection(url,user, password:pass);
        tampilkan();
        setColWidht();
    }
}
```

Pada class Main ini saya menggunakan JFrame Form. Sebelum masuk ke konstruktor saya membuat tiga final String yaitu url, user dan pass, dan membuat Connection dengat set null. Didalam konstruktor bawaan JFrame saya menambahkan deklarasi dari Connection dan memanggil methode tampilkan() dan setColWidht().

- **Methode setColWidht()**

```
private void setColWidht() {
    int[] noCol = {0,1,2,4,5};
    int[] noColW = {40,200,30,40,150};

    TableColumnModel tblModel = tbl_isi.getColumnModel();
    for(int i = 0; i < noCol.length; i++){
        tblModel.getColumn(noCol[i]).setPreferredWidth(noColW[i]);
        tblModel.getColumn(noCol[i]).setMaxWidth(noColW[i]);
        tblModel.getColumn(noCol[i]).setMinWidth(noColW[i]);
    }
}
```

Methode ini berfungsi untuk mengatur lebar colom pada tabel. Disini saya mengubah lebar colom dengan index 0,1,2,4,5. Methode ini tidak ada hubungan dengan database, namun dengan methode ini tabel yang ditampilkan pada GUI akan lebih rapi.

- **Methode tampilkan()**

```
private void tampilkan(){
    PreparedStatement st = null;
    ArrayList<Santri> list = new ArrayList();
    ResultSet rs = null;
    String query = "SELECT * FROM santri ";
    try{
        st = conn.prepareStatement( sql.query);
        rs = st.executeQuery();
        while(rs.next()){
            Santri sn = new Santri();

            sn.setId( id:rs.getString( columnLabel:"id_santri"));
            sn.setNama( nama:rs.getString( columnLabel:"nama"));
            sn.setJk( jk:rs.getString( columnLabel:"jenis_kelamin"));
            sn.setAlamat( alamat:rs.getString( columnLabel:"alamat"));
            sn.setTlp( tlp:rs.getString( columnLabel:"no_tlp"));
            sn.setUsia( usia:Integer.parseInt( s:rs.getString( columnLabel:"usia")));

            list.add( e:sn);
        }

        DefaultTableModel tbl = (DefaultTableModel) tbl_isi.getModel();
        tbl.setRowCount( rowCount:0);
        int i = 1;
        for(Santri data : list){
            Object[] baris = new Object[6];

            baris[0] = data.getId();
            baris[1] = data.getNama();
            baris[2] = data.getJk();
            baris[3] = data.getAlamat();
            baris[4] = data.getUsia();
            baris[5] = data.getTlp();

            tbl.addRow( rowData:baris);
        }
    } catch (SQLException e){
        e.printStackTrace();
    }
    btn_hapus.setVisible( aFlag:false);
    btn_batal.setVisible( aFlag:false);
    if(lb_tambah.getText().equals( anObject:"UBAH")){
        lb_tambah.setText( text:"TAMBAH");
    }
}
```

Method ini berfungsi sebagai pemanggil data dari database dan di cetak pada tabel pada GUI. Setelah query select dijalankan maka hasil akan disimpan dalam ArrayList dengan tipe data Santri yang merupakan sebuah class dengan atribut yang telah disesuaikan dengan field pada database. Setelah selesai memanggil dan memasukkan ke dalam tabel, method akan menghilangkan tombol hapus dan batal, dan mengubah lebel pada tombol tabah jika lebel bertuliskan “UBAH”. Tombol ubah hapus dan batal akan muncul ketika user telah memilih salah satu item pada tabel.

• Class Santri

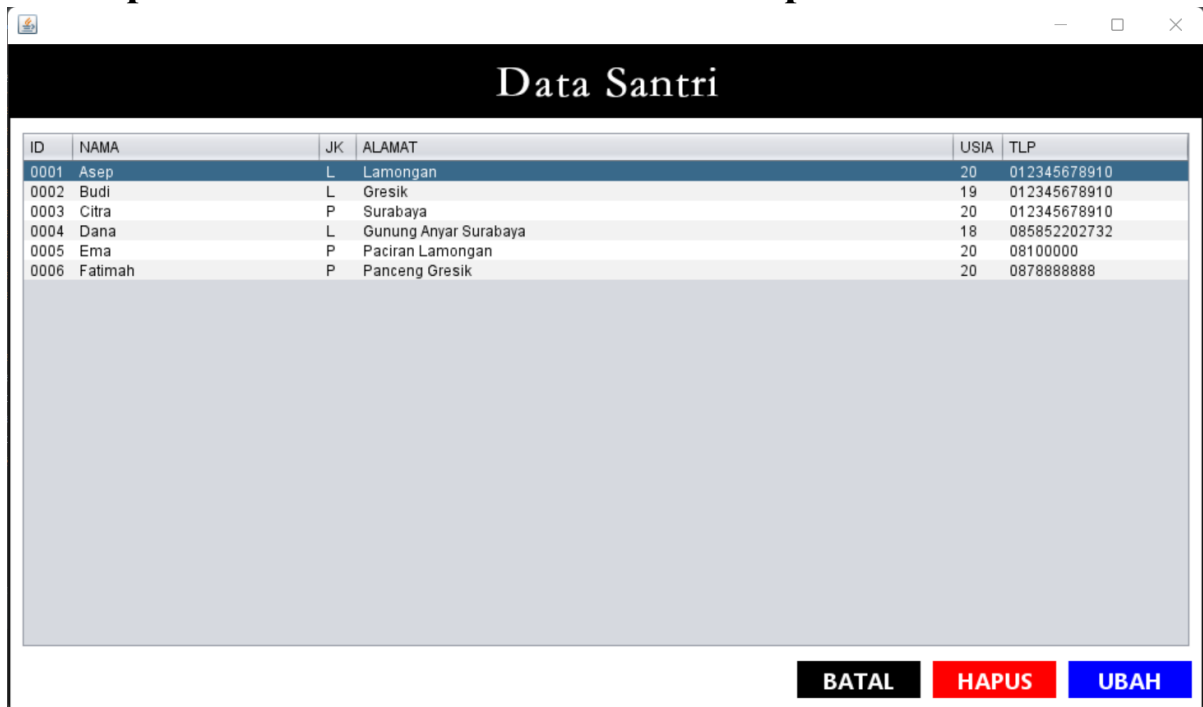
```
public class Santri {  
    private String id;  
    private String nama;  
    private String jk;  
    private String alamat;  
    private String tlp;  
    private int usia;  
  
    public String getId() {  
        return id;  
    }  
  
    public void setId(String id) {  
        this.id = id;  
    }  
  
    public String getNama() {  
        return nama;  
    }  
  
    public void setNama(String nama) {  
        this.nama = nama;  
    }  
  
    public String getJk() {  
        return jk;  
    }  
  
    public void setJk(String jk) {  
        this.jk = jk;  
    }  
  
    public String getAlamat() {  
        return alamat;  
    }  
  
    public void setAlamat(String alamat) {  
        this.alamat = alamat;  
    }  
  
    public String getTlp() {  
        return tlp;  
    }  
  
    public void setTlp(String tlp) {  
        this.tlp = tlp;  
    }  
  
    public int getUsia() {  
        return usia;  
    }  
  
    public void setUsia(int usia) {  
        this.usia = usia;  
    }  
}
```

Class ini berfungsi sebagai wadah untuk mempermudah dalam menangkap data baik dari database saat perintah select atau pun dari inputan user seperti saat perintah insert dan update.

Class telah diberi atribut yang telah disesuaikan dengan database, seperti penggunaan tipe data untuk char dan varchar pada database pada program dapat menggunakan String.

Untuk akses atribut, class telah dilengkapi dengan method getter dan setter.

- **Tampilan GUI ketika user memilih item pada tabel**



ID	NAMA	JK	ALAMAT	USIA	TLP
0001	Asep	L	Lamongan	20	012345678910
0002	Budi	L	Gresik	19	012345678910
0003	Citra	P	Surabaya	20	012345678910
0004	Dana	L	Gunung Anyar Surabaya	18	085852202732
0005	Ema	P	Paciran Lamongan	20	08100000
0006	Fatimah	P	Panceng Gresik	20	0878888888

BATAL **HAPUS** **UBAH**

Ketika user menekan salah satu item pada tabel, maka tombol yang awalnya TAMBAH berubah menjadi UBAH. Dan juga menampilkan tombol HAPUS dan BATAL.

- **Method `tbl_isiMouseClicked()`**

```
private void tbl_isiMouseClicked(java.awt.event.MouseEvent evt) {  
    btn_hapus.setVisible( aFlag:true);  
    btn_batal.setVisible( aFlag:true);  
    if(lb_tambah.getText().equals( anObject: "TAMBAH")) {  
        lb_tambah.setText( text: "UBAH");  
    }  
}
```

Ketika user melakukan klik tabel pada GUI, `btn_hapus` dan `btn_batal` diset menjadi visible dan meset lebel pada `btn_tambah` (`lb_tambah`) menjadi “UBAH” jika lebel sebelumnya “TAMBAH”.

- **Method `btn_batalMouseClicked()`**

```
private void btn_batalMouseClicked(java.awt.event.MouseEvent evt) {  
    tampilkan();  
}
```

Ketika user menekan `btn_batal`, tombol akan memanggil metode `tampilkan()`. Pada metode `tampilkan()` tampilan GUI akan direfresh.

- **Methode btn_hapusMouseClicked()**

```
private void btn_hapusMouseClicked(java.awt.event.MouseEvent evt) {  
    hapus((String) tbl_isi.getValueAt (row:tbl_isi.getSelectedRow(), column:0));  
    tampilkan();  
}
```

Ketika user menekan btn_hapus, tombol akan memanggil metode hapus() dengan paramater id dengan tipe data String yang diambil dari tabel. Id yang diambil dari tabel berdasarkan yang dipilih user. Setelah memanggil metode hapus(), akan dilanjutkan dengan memanggil metode tampilkan() yang berfungsi sebagai refresh.

- **Methode hapus()**

```
private void hapus(String id){  
    String query = "DELETE FROM santri WHERE id_santri = ?";  
    try{  
        PreparedStatement stmt = null;  
        stmt = conn.prepareStatement (sql:query);  
  
        stmt.setString( parameterIndex:1, x:id);  
  
        stmt.executeUpdate();  
    } catch (SQLException e){  
        e.printStackTrace();  
    }  
}
```

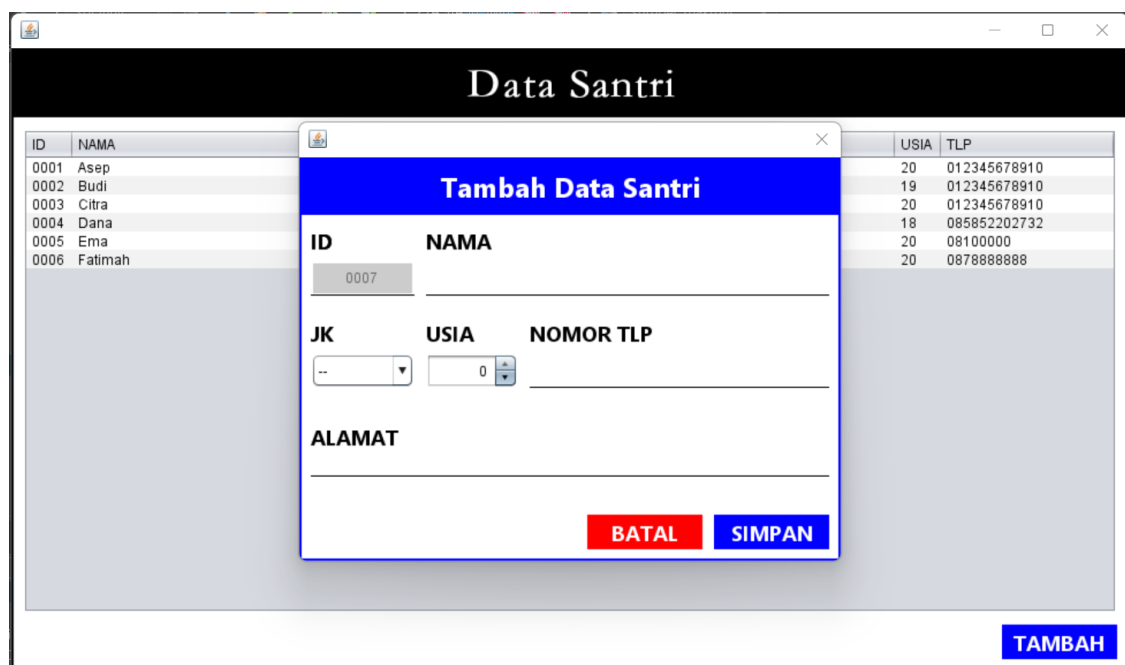
Ketika metode ini dipanggil dengan parameter id, metode akan menghapus data pada database sesuai dengan id yang diberikan.

- **Methode btn_tambahMouseClicked()**

```
private void btn_tambahMouseClicked(java.awt.event.MouseEvent evt) {  
    try {  
        if(lb_tambah.getText().equalsIgnoreCase (anotherString: "TAMBAH")){  
            new Tambah (parent:null, modal:true).setVisible (b:true);  
        } else {  
            int index = tbl_isi.getSelectedRow();  
            Santri sn = new Santri();  
            sn.setId((String) tbl_isi.getValueAt (row:index, column:0));  
            sn.setNama((String) tbl_isi.getValueAt (row:index, column:1));  
            sn.setJk((String) tbl_isi.getValueAt (row:index, column:2));  
            sn.setAlamat((String) tbl_isi.getValueAt (row:index, column:3));  
            sn.setUsia((int) tbl_isi.getValueAt (row:index, column:4));  
            sn.setTlp((String) tbl_isi.getValueAt (row:index, column:5));  
            new Tambah (parent:null, modal:true, hsn:sn).setVisible (b:true);  
        }  
    } catch (SQLException ex) {  
        Logger.getLogger( name:Main.class.getName()).log ( level:Level.SEVERE, msg:null, thrown:ex);  
    }  
    tampilkan();  
}
```

Ketika user menekan btn_tambah dan lb_tambah bertuliskan “TAMBAH”, maka metode akan memanggil JDialog Tambah dengan konstruktor tanpa parameter Santri. Sedangkan jika ketika ditekan user lb_tambah bertuliskan “UBAH”, maka metode akan membuat instansi dari class Santri dan mengisinya sebelum memanggil JDialog Tambah dengan konstruktor yang menggunakan parameter Santri.

- User menekan tombol dengan label “TAMBAH”



Ketika user menekan tombol dengan label “TAMBAH”, maka JDialog Tambah akan muncul seperti pada gambar di atas. Tampilan GUI JDialog seperti gambar di atas, memiliki form id, nama, jenis kelamin, usia, nomor telpon dan alamat. Khusus id akan digenerate otomatis dan tidak dapat diubah.

- Konstruktor JDialog

```
public class Tambah extends javax.swing.JDialog {

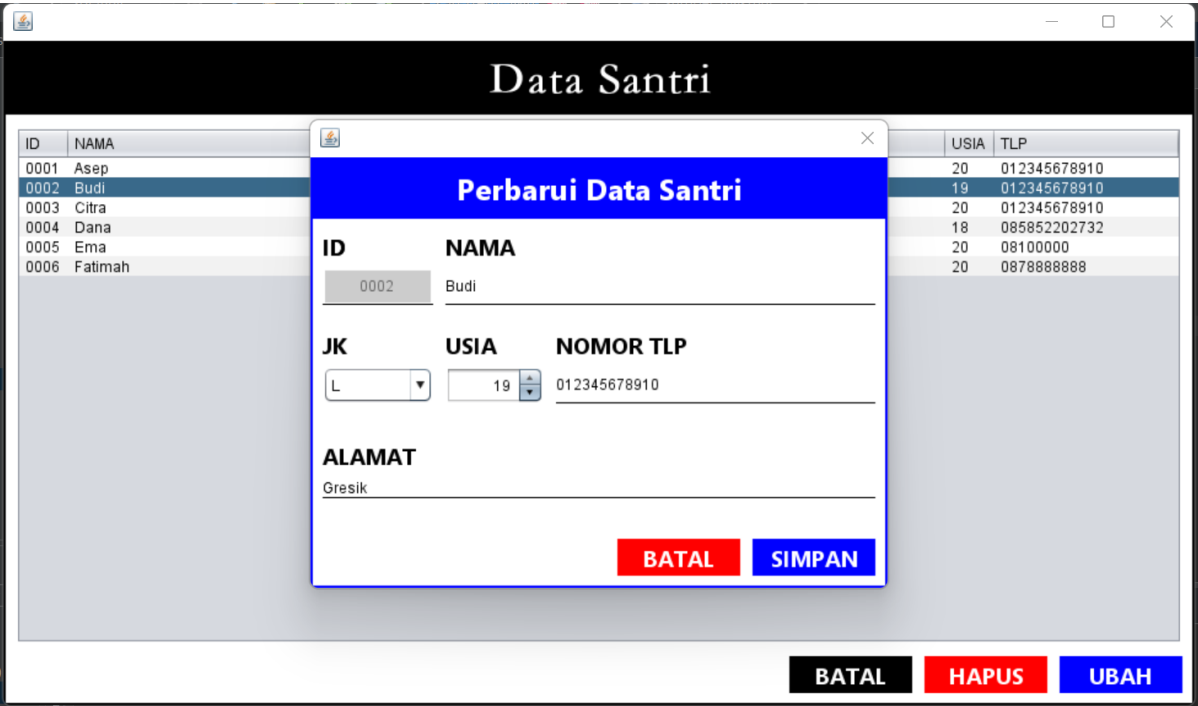
    /**
     * Creates new form Tambah
     */
    private final String url = "jdbc:postgresql://localhost:5432/PBO";
    private final String user = "postgres";
    private final String pass = "5127";
    private Connection conn = null;
    private Santri sn = new Santri();

    public Tambah(java.awt.Frame parent, boolean modal) throws SQLException {
        super(frame:parent, bin:modal);
        initComponents();
        conn = DriverManager.getConnection(url,user, password:pass);
        generateNomer();
    }

    public Tambah(java.awt.Frame parent, boolean modal, Santri hsn) throws SQLException {
        super(frame:parent, bin:modal);
        initComponents();
        conn = DriverManager.getConnection(url,user, password:pass);
        lb_judul.setText( text: "Perbarui Data Santri");
        setData(hsn);
    }
}
```

Pada class Tambah ini saya menggunakan JDialog. Sebelum masuk ke konstruktor saya membuat tiga final String yaitu url, user dan pass, membuat Connection dengat set null, dan mendeklarasikan instansi dari Santri. Dalam konstruktor bawaan JDialog (tanpa parameter Santri) saya menambahkan deklarasi dari Connection dan memanggil method generateNomer(). Untuk kostruktor yang menggunakan parameter Santri saya menambahkan deklarasi dari Connection dan memanggil method generateNomer().

- User menekan tombol dengan lebel “UBAH”



Ketika user menekan tombol dengan lebel “UBAH”, maka JDialog Tambah akan muncul seperti pada gambar di atas. Tampilan GUI JDialog seperti gambar di atas, memiliki form id, nama, jenis kelamin, usia, nomor telpon dan alamat. Tampilan sama persis seperti waktu menekan tombol dengan lebel “TAMBAH”, perbedaanya hanya ada pada form yang telah diisi sesuai dengan yang dipilih.

- **Methode setData()**

```
private void setData(Santri hsn){
    tf_id.setText( t:hsn.getId());
    tf_nama.setText( t:hsn.getNama());
    tf_alamat.setText( t:hsn.getAlamat());
    tf_tlp.setText( t:hsn.getTlp());
    cbx_jk.setSelectedItem( anObject:hsn.getJk());
    sp_usia.setValue( value:hsn.getUsia());
}
```

Methode ini dipanggil ketika konstruktor yang digunakan adalah konstruktor dengan parameter Santri. Methode ini berfungsi sebagai pengisi form, sehingga form yang ditampilkan akan sesuai dengan data yang dikirim.

- **Methode generateNomer()**

```
private void generateNomer(){
    PreparedStatement st = null;
    ResultSet rs = null;
    String query = "SELECT id_santri AS nomor FROM santri ORDER BY id_santri DESC LIMIT 1";
    String urutan = null;

    try{
        st = conn.prepareStatement( sql:query);
        rs = st.executeQuery();
        if(rs.next()){
            int nomor = Integer.parseInt( s:rs.getString( columnLabel:"nomor"));
            nomor++;
            urutan = String.format( format:"%04d", args:nomor);
        } else {
            urutan = "0001";
        }
        tf_id.setText( t:urutan);
    } catch (SQLException e){
        e.printStackTrace();
    }
}
```

Methode ini berfungsi sebagai pengisi form id, karena id tidak dapat di ubah user melalui tampilan GUI. Methode akan memanggil data dari database dengan urutan terakhir, kemudian melakukan setText pada form id sesuai output ditambah satu.

- **Ketika user menekan tombol batal**

```
private void btn_hapusMouseClicked(java.awt.event.MouseEvent evt) {
    dispose();
}
```

Metode ini berfungsi untuk close JDialog ketika user menekan tombol batal.

- **Ketika user menekan tombol simpan**

```
private void btn_simpanMouseClicked(java.awt.event.MouseEvent evt) {  
    setSn();  
    if(lb_judul.getText().equals("Perbarui Data Santri")){  
        perbarui();  
    } else {  
        tambah();  
    }  
    dispose();  
}
```

Method ini akan memanggil method setSn(). Dan memanggil method perbarui(), ketika judul bertuliskan “Perbarui Data Santri”. Dan jika tidak method ini akan memanggil method tambah(). Dilanjut dengan close JDialog.

- **Method setSn()**

```
private void setSn(){  
    this.sn.setId(id:tf_id.getText());  
    this.sn.setNama(nama:tf_nama.getText());  
    this.sn.setAlamat(alamat:tf_alamat.getText());  
    this.sn.setTlp(tlp:tf_tlp.getText());  
    this.sn.setJk(jk:cbx_jk.getSelectedItem().toString());  
    this.sn.setUsia(usia:Integer.parseInt(s:sp_usia.getValue().toString()));  
}
```

Method ini akan memasukan nilai-nilai pada form yang telah diisi user menjadi instansi dari class Santri yang telah dideklarasikan diawal JDialog.

- **Method tambah()**

```
private void tambah(){  
    String query = "INSERT INTO santri "  
        + "(id_santri,nama,jenis_kelamin,alamat,usia,no_tlp) "  
        + "values (?,?,?,?,?,?)";  
    try{  
        PreparedStatement stmt = null;  
        stmt = conn.prepareStatement(sql:query);  
  
        stmt.setString(parameterIndex:1, x:sn.getId());  
        stmt.setString(parameterIndex:2, x:sn.getNama());  
        stmt.setString(parameterIndex:3, x:sn.getJk());  
        stmt.setString(parameterIndex:4, x:sn.getAlamat());  
        stmt.setInt(parameterIndex:5, x:sn.getUsia());  
        stmt.setString(parameterIndex:6, x:sn.getTlp());  
  
        stmt.executeUpdate();  
    } catch (SQLException e){  
        e.printStackTrace();  
    }  
}
```

Methode ini akan menkonfersi data dalam bentuk instansi dari Santri menjadi query insert untuk database. Setelah query tercipta, query langsung dijalankan.

- **Methode perbarui()**

```
private void perbarui() {
    String query = "UPDATE santri SET nama = ?, jenis_kelamin = ?, "
        + "alamat = ?, usia = ?, no_tlp = ? WHERE id_santri = ?";
    try{
        PreparedStatement stmt = null;
        stmt = conn.prepareStatement( sql:query);

        stmt.setString( parameterIndex:6, x:Sn.getId());
        stmt.setString( parameterIndex:1, x:Sn.getNama());
        stmt.setString( parameterIndex:2, x:Sn.getJk());
        stmt.setString( parameterIndex:3, x:Sn.getAlamat());
        stmt.setInt( parameterIndex:4, x:Sn.getUsia());
        stmt.setString( parameterIndex:5, x:Sn.getTlp());

        stmt.executeUpdate();
    } catch (SQLException e){
        e.printStackTrace();
    }
}
```

Methode ini akan menkonfersi data dalam bentuk instansi dari Santri menjadi query update untuk database. Setelah query tercipta, query langsung dijalankan.