

## MODUL 10

### Graphical User Interface (GUI) Java NetBeans & Aplikasi CRUD (Create, Read, Update, Delete)

#### 10.1 NetBeans

NetBeans IDE adalah GUI Editor yang memudahkan programmer untuk melakukan pembelajaran dan pengembangan software atau dalam tahapan membuat software atau aplikasi, yang sebelumnya banyak GUI Editor dalam bahasa pemrograman seperti notepad atau text editor, yang sebenarnya jika kita memakai notepad atau text editor maka pengetahuan kita akan lebih berkembang dibanding memakai GUI Editor.

NetBeans mengacu pada dua hal, yakni platform untuk pengembangan aplikasi desktop java dan sebuah Integrated Development Environment (IDE) yang dibangun menggunakan platform NetBeans.

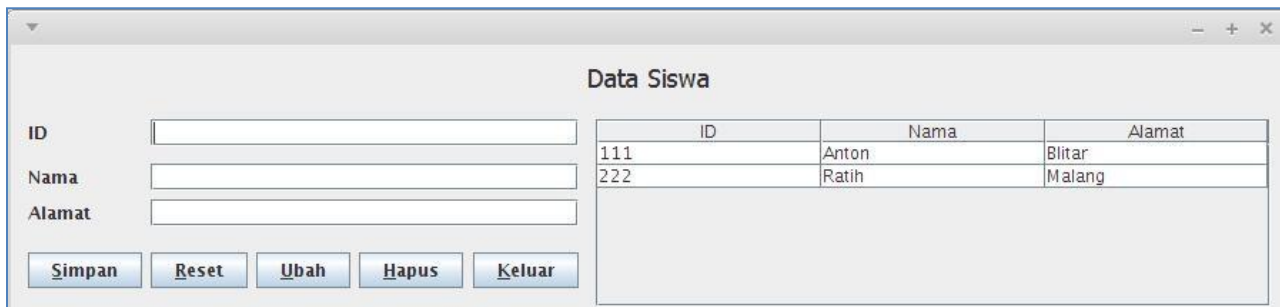
NetBeans IDE adalah sebuah lingkungan pengembangan untuk pemrogram, menulis, mengompilasi, mencari kesalahan dan menjalankan program. NetBeans IDE ditulis dalam bahasa Java, namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas NetBeans IDE. NetBeans IDE adalah sebuah produk bebas dengan tanpa batasan.

NetBeans adalah Integrated Development Environment (IDE) berbasis Java yang berawal pada tahun 1997 yang diawali dari Xelfi, sebuah proyek mahasiswa di bawah bimbingan Fakultas Matematika dan Fisika Universitas Charles, Praha dan akhirnya Sun Microsystems membelinya. Pada tahun 1999, dikembangkan NetBeans Integrated Development Environment (IDE) berbasis Java dari Sun Microsystems yang berjalan di atas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan di berbagai macam platform seperti Windows, Linux, Mac OS X and Solar dengan kata lain, java dengan NetBeans dapat dikatakan sebagai pemrograman multi platform karena fleksibel dengan segala sistem operasi.

Pada tahun 2000, sun menjadikan NetBeans sebagai software development yang Open Source, dengan kata lain software ini gratis tanpa biaya karena software ini di bawah pengembangan bersama. Software NetBeans dapat di download di <http://www.netbeans.com> atau juga bisa di download di situs yang lain.

## 10.2 Aplikasi CRUD

Aplikasi CRUD sederhana ini memiliki logika yang sangat sederhana, yaitu pertama saat aplikasi dijalankan, maka aplikasi CRUD akan meload semua data dan menampilkannya di tabel, serta mengisi text field dengan data – data yang sama pada tabel yang sudah siap untuk diubah. Selanjutnya jika melakukan perubahan berarti memerintahkan aplikasi untuk mengupdate data ke database, sedangkan jika melakukan penambahan data maka aplikasi akan menambah data tersebut dengan perintah insert ke database. Dan terakhir melakukan perintah delete, baris yang dipilih pada tabel akan dihapus secara permanent. Aplikasi CRUD pada data siswa seperti gambar di bawah ini.



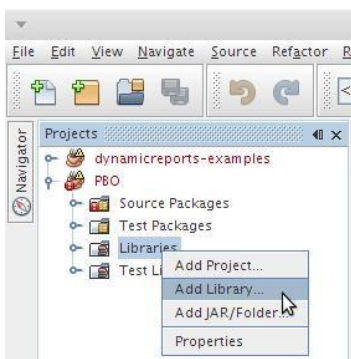
ID	Nama	Alamat
111	Anton	Blitar
222	Ratih	Malang

## 10.3 Latihan

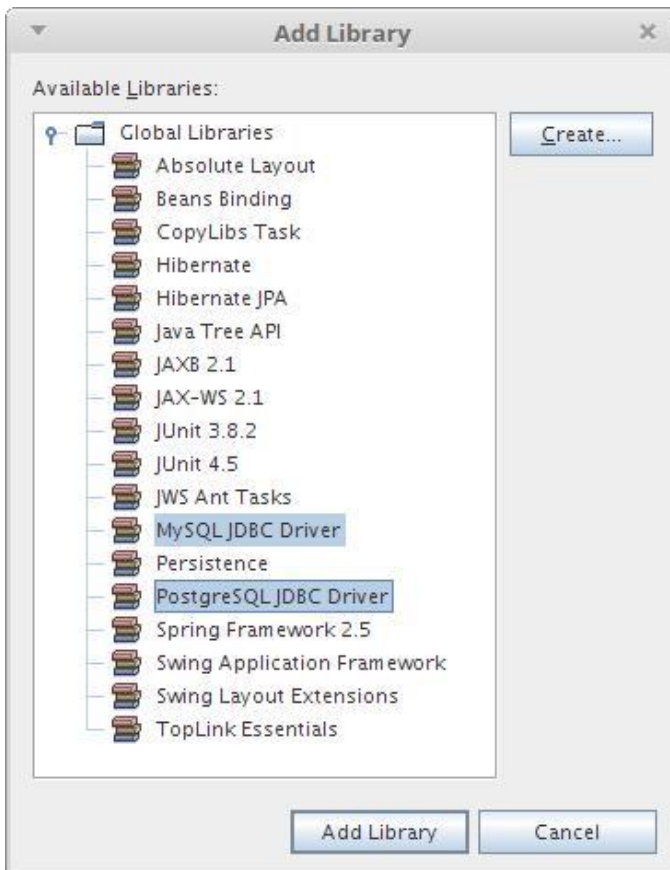
A. Buat database mysql dengan nama crud+nim misalnya 'crud15650001' dan buat tabel 'siswa'!

```
1 CREATE TABLE siswa (  
2   id varchar(20) NOT NULL,  
3   nama varchar(100) DEFAULT NULL,  
4   alamat text,  
5   PRIMARY KEY (id)  
6 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
7  
8 insert into siswa(id,nama,alamat) values  
   ('111','Anton','Blitar'),('222','Ratih','Malang');
```

B. Buka NetBeans dan klik menu **"Window > Projects"**. Pilih salah satu project lalu klik kanan **"Libraries > Add Library"**.



Pilih **"MySQL JDBC Driver"** untuk driver MySQL dan pilih **"PostgreSQL JDBC Driver"** untuk driver PostgreSQL. Klik tombol **"Add Library"**.



### C. Source code koneksi MySQL di NetBeans : Nama file : **KoneksiMysql.java**

```

1  package crud;
2  import java.sql.Connection;
3  import java.sql.DriverManager;
4  import java.sql.SQLException;
5  public class KoneksiMysql {
6      private Connection connect;
7      private String driverName = "com.mysql.jdbc.Driver"; // Driver Untuk Koneksi Ke MySQL
8      private String jdbc = "jdbc:mysql://";
9      private String host = "localhost"; // Bisa Menggunakan IP Anda, Cntr : 192.168.100.100
10     private String port = "3306/"; // Port ini port MySQL
11     private String database = "crud15650001"; // Ini Database yang akan digunakan
12     private String url = jdbc + host + port + database;
13     private String username = "root"; // username default mysql
14     private String password = "";
15     public Connection getKoneksi() throws SQLException {
16         if (connect == null) {
17             try {
18                 Class.forName(driverName);
19                 System.out.println("Class Driver Ditemukan");
20                 try {
21                     connect = DriverManager.getConnection(url, username, password);
22                     System.out.println("Koneksi Database Sukses");
23                 } catch (SQLException se) {
24                     System.out.println("Koneksi Database Gagal : " + se);
25                     System.exit(0);
26                 }
27             } catch (ClassNotFoundException cnfe) {
28                 System.out.println("Class Driver Tidak Ditemukan, Terjadi Kesalahan Pada : " + cnfe);
29                 System.exit(0);
30             }
31         }
32         return connect;
33     }
34 }

```

Untuk database Postgresql :

Nama file : **KoneksiPostgresql.java**

```
1 package crud;
2 import java.sql.Connection;
3 import java.sql.DriverManager;
4 import java.sql.SQLException;
5 public class KoneksiPostgresql {
6     private Connection connect;
7     private String driverName = "org.postgresql.Driver"; // Driver Untuk Koneksi Ke PostgreSQL
8     private String jdbc = "jdbc:postgresql://";
9     private String host = "localhost"; // Host ini Bisa Menggunakan IP Anda, Contoh : 192.168.100.100
10    private String port = "5432/"; // Port Default PostgreSQL
11    private String database = "crud15650001"; // Ini Database yang akan digunakan
12    private String url = jdbc + host + port + database;
13    private String username = "postgres"; //
14    private String password = "";
15    public Connection getKoneksi() throws SQLException {
16        if (connect == null) {
17            try {
18                Class.forName(driverName);
19                System.out.println("Class Driver Ditemukan");
20                try {
21                    connect = DriverManager.getConnection(url, username, password);
22                    System.out.println("Koneksi Database Sukses");
23                } catch (SQLException se) {
24                    System.out.println("Koneksi Database Gagal : " + se);
25                    System.exit(0);
26                }
27            } catch (ClassNotFoundException cnfe) {
28                System.out.println("Class Driver Tidak Ditemukan, Terjadi Kesalahan Pada : " + cnfe);
29                System.exit(0);
30            }
31        }
32        return connect;
33    }
34 }
```

D. Source code class CRUD Nama file : **CRUD.java**

```
1 package crud;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.Statement;
6 import java.sql.ResultSet;
7 import java.sql.SQLException;
8
9 public class CRUD {
10     private String id, nama, alamat;
11     private Connection CRUDkoneksi;
12     private PreparedStatement CRUDpsmt;
13     private Statement CRUDstat;
14     private ResultSet CRUDhasil;
15     private String CRUDquery;
16
17     public CRUD(){
18         try { KoneksiMysql connection = new KoneksiMysql();
19             CRUDkoneksi = connection.getKoneksi();
20         } catch (SQLException ex){
21             System.out.println(ex);
22         }
23     }
24
25     public void setID(String id)
26     { this.id = id;
27     }
28
29     public String getID()
30     { return id;
31     }
32
33     public void setNama(String nama)
34     { this.nama = nama;
35     }
36 }
```



```

37     public String getNama()
38     { return nama;
39     }
40
41     public void setAlamat(String alamat)
42     { this.alamat = alamat;
43     }
44
45     public String getAlamat()
46     { return alamat;
47     }
48
49     public ResultSet tampilData()
50     { CRUDquery = "select * from siswa";
51       try { CRUDstat = CRUDkoneksi.createStatement();
52             CRUDhasil = CRUDstat.executeQuery(CRUDquery);
53         } catch (Exception e) {
54         }
55       return CRUDhasil;
56     }
57
58     public void simpanData(String id,String nama,String alamat)
59     { CRUDquery = "insert into siswa values(?,?,?)";
60       try {
61         CRUDpsmt = CRUDkoneksi.prepareStatement(CRUDquery);
62         CRUDpsmt.setString(1, id);
63         CRUDpsmt.setString(2, nama);
64         CRUDpsmt.setString(3, alamat);
65         CRUDpsmt.executeUpdate();
66         CRUDpsmt.close();
67       } catch (Exception e) {
68         System.out.println(e);
69       }
70     }
71

```

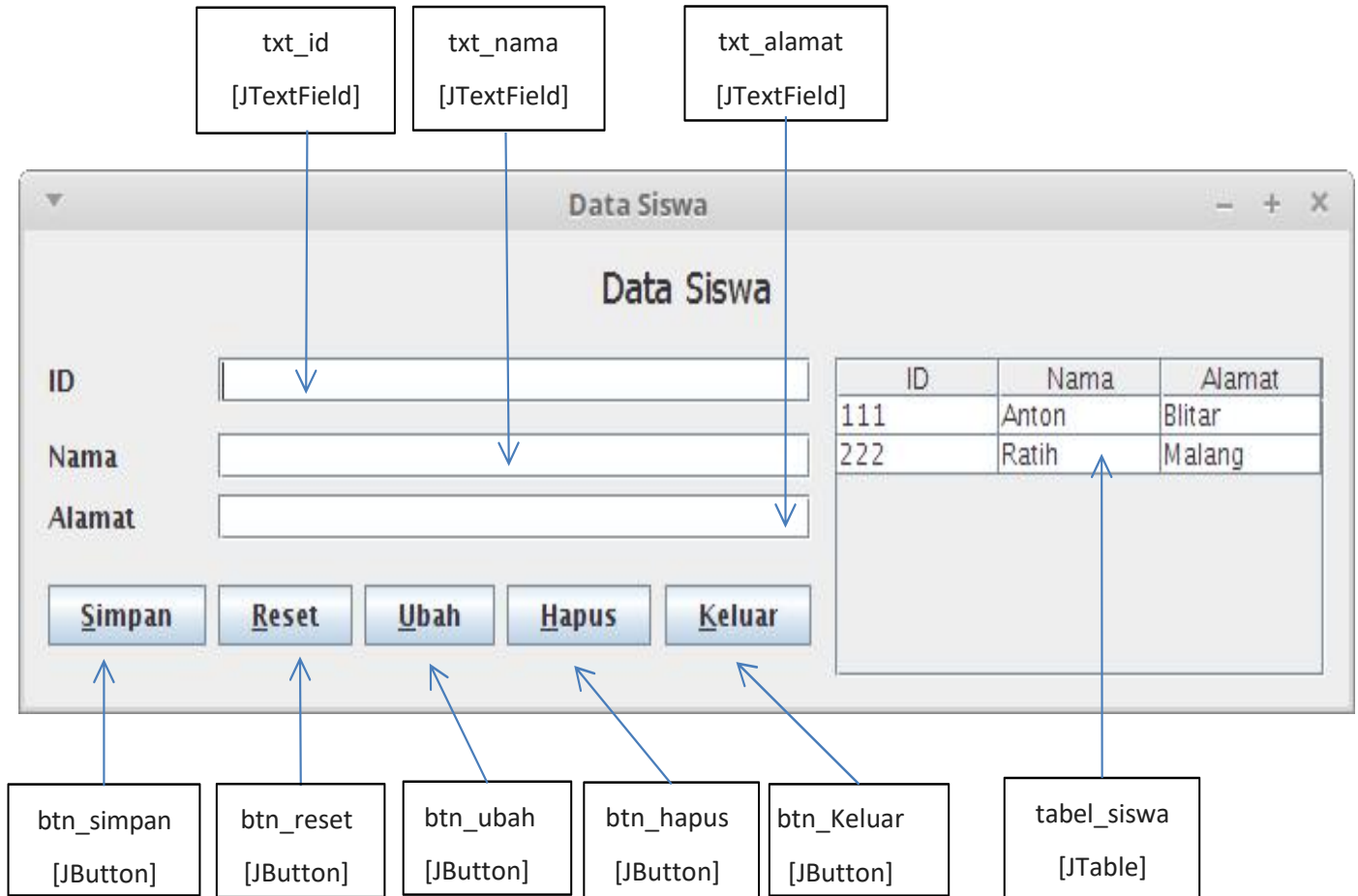
```

72     public void ubahData(String id,String nama,String alamat)
73     { CRUDquery = "update siswa set nama=?, alamat=? where id=?";
74       try {
75         CRUDpsmt = CRUDkoneksi.prepareStatement(CRUDquery);
76         CRUDpsmt.setString(1, nama);
77         CRUDpsmt.setString(2, alamat);
78         CRUDpsmt.setString(3, id);
79         CRUDpsmt.executeUpdate();
80         CRUDpsmt.close();
81       } catch (Exception e) {
82         System.out.println(e);
83       }
84     }
85
86     public void hapusData(String id)
87     { CRUDquery = "delete from siswa where id=?";
88       try {
89         CRUDpsmt = CRUDkoneksi.prepareStatement(CRUDquery);
90         CRUDpsmt.setString(1, id);
91         CRUDpsmt.executeUpdate();
92         CRUDpsmt.close();
93       } catch (Exception e) {
94         System.out.println(e);
95       }
96     }
97
98
99 }
100

```

E. Buat Form Siswa dengan JFrame, klik kanan package “**crud > New > JFrame Form**” dan beri nama dengan Form\_Siswa. Lihat catatan JFrame pada halaman 10.

Nama file : Form\_Siswa.java Tab Design



#### Tab Source

```
package crud;
```

```
import java.sql.ResultSet; import java.sql.SQLException;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.table.DefaultTableModel;
```

```
public class Form_Siswa extends javax.swing.JFrame { private DefaultTableModel tabmode;
```

```
private ResultSet hasil;
```

```
// membuat objek baru dari class CRUD yang sudah terkoneksi dengan database CRUD aa = new CRUD();
```

```

public Form_Siswa() throws SQLException { initComponents();
tampil_database();
}

```

```

// method untuk menampilkan data ke tabel_siswa [JTable] public void tampil_database(){
Object [] baris = {"ID","Nama","Alamat"}; tabmode = new DefaultTableModel(null, baris);

```

```

Tabel_siswa.setModel(tabmode); try {
hasil = aa.tampilData(); while (hasil.next()){
aa.setID(hasil.getString("id")); aa.setNama(hasil.getString("nama")); aa.setAlamat(hasil.getString("alamat"));
String[] data = {aa.getID(), aa.getNama(), aa.getAlamat()}; tabmode.addRow(data);
}
} catch (Exception e) {
}
}

```

```

// method untuk reset text public void reset_text() {
txt_id.setText(""); txt_nama.setText(""); txt_alamat.setText("");
}

```

```

// klik double tombol btn_simpan untuk menyimpan data
private void btn_simpanActionPerformed(java.awt.event.ActionEvent evt) { if
(txt_id.getText().trim().equals("")){
JOptionPane.showMessageDialog(null,"Maaf, ID belum diisi !"); txt_id.requestFocus();
} else if (txt_nama.getText().trim().equals("")){ JOptionPane.showMessageDialog(null,"Maaf, Nama belum
diisi !"); txt_nama.requestFocus();
} else if (txt_alamat.getText().trim().equals("")){ JOptionPane.showMessageDialog(null,"Maaf, Alamat belum
diisi !"); txt_alamat.requestFocus();
} else {
try { aa.setID(txt_id.getText()); aa.setNama(txt_nama.getText()); aa.setAlamat(txt_alamat.getText());
aa.simpanData(aa.getID(), aa.getNama(), aa.getAlamat());

```

```
JOptionPane.showMessageDialog(null, "Data berhasil tersimpan" , "Informasi",
JOptionPane.INFORMATION_MESSAGE);

tampil_database(); reset_text();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "Data gagal tersimpan" , "Informasi",
JOptionPane.INFORMATION_MESSAGE);

}

}

}
```



```

// klik double tombol btn_reset untuk reset text

private void btn_resetActionPerformed(java.awt.event.ActionEvent evt) { reset_text();
}

// klik double tombol btn_ubah untuk mengubah data

private void btn_ubahActionPerformed(java.awt.event.ActionEvent evt) { if
(txt_id.getText().trim().equals("")){

JOptionPane.showMessageDialog(null,"Maaf, ID belum diisi !"); txt_id.requestFocus();

} else if (txt_nama.getText().trim().equals("")){ JOptionPane.showMessageDialog(null,"Maaf, Nama belum
diisi !"); txt_nama.requestFocus();

} else if (txt_alamat.getText().trim().equals("")){ JOptionPane.showMessageDialog(null,"Maaf, Alamat belum
diisi !"); txt_alamat.requestFocus();

} else {

try { aa.setID(txt_id.getText()); aa.setNama(txt_nama.getText()); aa.setAlamat(txt_alamat.getText());
aa.ubahData(aa.getID(), aa.getNama(), aa.getAlamat());

JOptionPane.showMessageDialog(null, "Data berhasil diubah", "Informasi",
JOptionPane.INFORMATION_MESSAGE);

tampil_database();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "Data gagal diubah", "Informasi",
JOptionPane.INFORMATION_MESSAGE);

}

}

}

// klik double tombol btn_hapus untuk menghapus data

private void btn_hapusActionPerformed(java.awt.event.ActionEvent evt) { if
(txt_id.getText().trim().equals("")){

JOptionPane.showMessageDialog(null,"Maaf, ID belum diisi !"); txt_id.requestFocus();

} else {

if (JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan menghapus data ini ?", "Warning", 2) ==
JOptionPane.YES_OPTION){

String id = "";

```

```

try { aa.setID(txt_id.getText()); aa.hapusData(aa.getID());

JOptionPane.showMessageDialog(null, "Data berhasil dihapus" , "Informasi",
JOptionPane.INFORMATION_MESSAGE);

tampil_database(); reset_text();

} catch (Exception e) {

JOptionPane.showMessageDialog(null, "Data gagal dihapus" , "Informasi",
JOptionPane.INFORMATION_MESSAGE);

}

}

}

}

// klik kanan tabel_siswa [JTable] > Events > Mouse > MouseClicked

// fungsinya ketika di klik salah satu data maka akan muncul di [JTextField] id, nama, alamat private void
tabel_siswaMouseClicked(java.awt.event.MouseEvent evt) {

try { int row = tabel_siswa.rowAtPoint(evt.getPoint());

String id = tabel_siswa.getValueAt(row, 0).toString(); String nama = tabel_siswa.getValueAt(row, 1).toString();
String alamat = tabel_siswa.getValueAt(row, 2).toString();

txt_id.setText(String.valueOf(id)); txt_nama.setText(String.valueOf(nama));
txt_alamat.setText(String.valueOf(alamat));

} catch (Exception e) {

}

}

// klik double tombol btn_keluar untuk keluar program

private void btn_keluarActionPerformed(java.awt.event.ActionEvent evt) {

if (JOptionPane.showConfirmDialog(null, "Apakah Anda yakin akan keluar ?", "Warning", 2)
== JOptionPane.YES_OPTION){ System.exit(0);

```

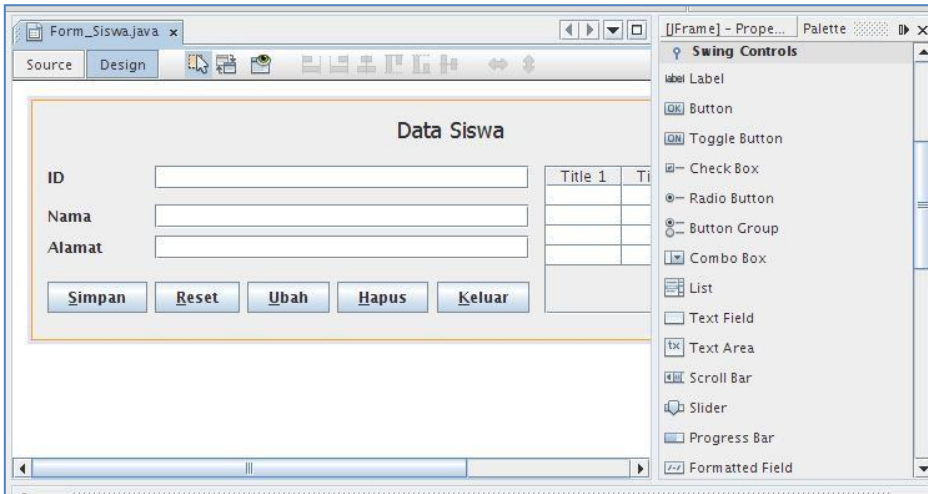
```
}
```

```
}
```

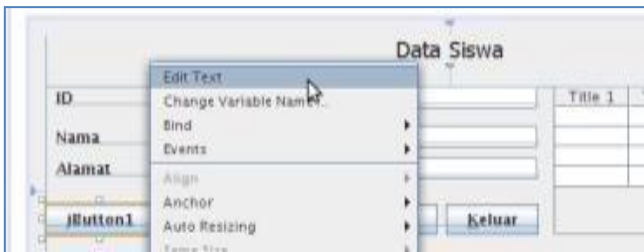
```
// End of Form_Siswa class
```

### Catatan JFrame:

1. Desainlah Form Data Siswa pada tab “Design” menggunakan Palette, klik menu **Window > Palette**. Drag JLabel, JButton, JTextField dan JTable ke area JFrame.



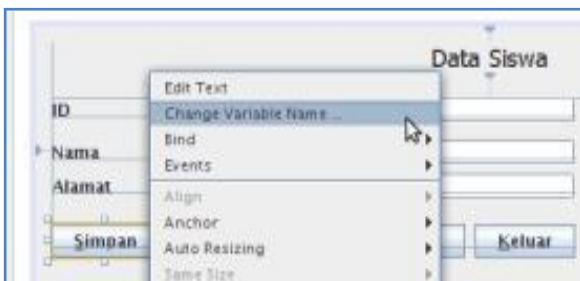
2. Untuk edit text, misalnya tombol “JButton1” ingin diubah menjadi tombol “Simpan”, klik kanan tombol “JButton1” dan pilih “Edit Text”.



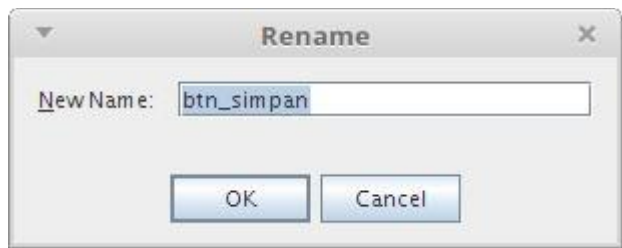
Ubah “JButton1” menjadi “Simpan”. Terapkan pula pada semua JLabel, JButton, JTextField dan JTable !



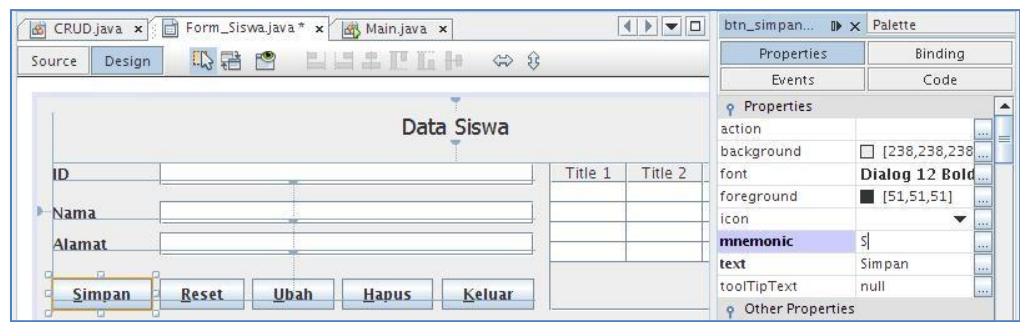
3. Untuk memberi nama variable, misalnya variable tombol “Simpan”, klik kanan tombol “Simpan” dan pilih “Change Variable Name”.



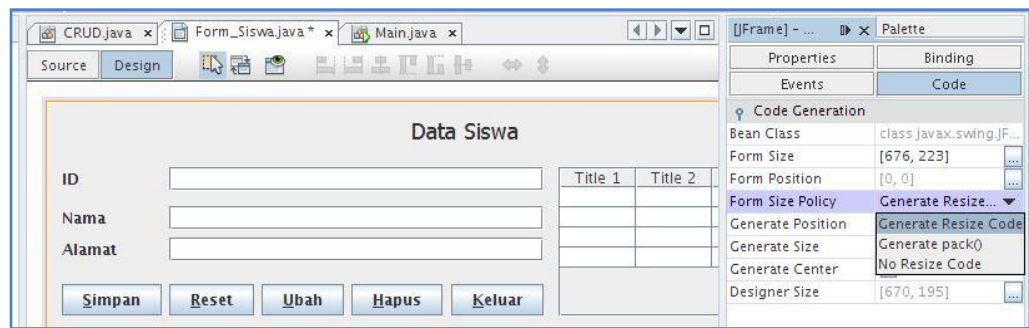
Ganti “New Name” yang awalnya “jButton1” menjadi “btn\_simpan”. Terapkan pula pada semua JLabel, JButton, JTextField dan JTable !



4. Agar tombol memiliki shortcut keyboard, misalnya tombol “Simpan” akan dijalankan ketika mengetikkan Alt+S. Caranya, klik tombol “Simpan”. Klik Menu **Window > Properties**. Klik Tab “Properties” dan pada “mnemonic” tambahkan huruf “S”. Tambahkan shortcut keyboard pula pada tombol yang lain.



5. Agar JFrame berada di posisi tengah, klik area JFrame. Klik Menu **Window > Properties**. Klik Tab “Code” dan pada “Form Size Policy” pilih “Generate Resize Code”.



6. Tulis source code yang berwarna biru pada tab “Source”.



F. Class Main untuk memanggil Form Siswa Nama file : **Main.java**

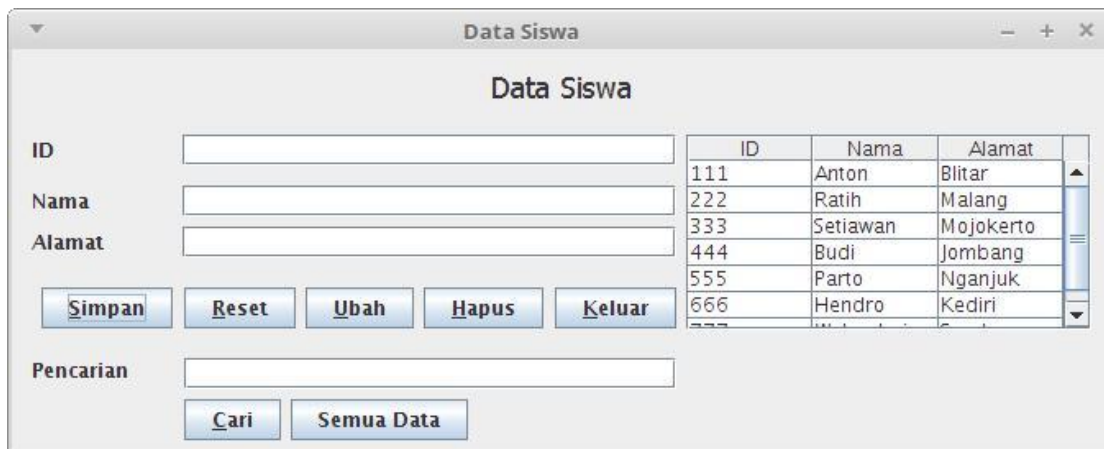
```
1 package crud;
2
3 public class Main {
4     public static void main(String[] args) {
5         try{
6             Form_Siswa form = new Form_Siswa();
7             form.setVisible(true);
8         }catch(Exception ex){
9             System.out.println(ex.toString());
10        }
11    }
12 }
```



## MODUL 10

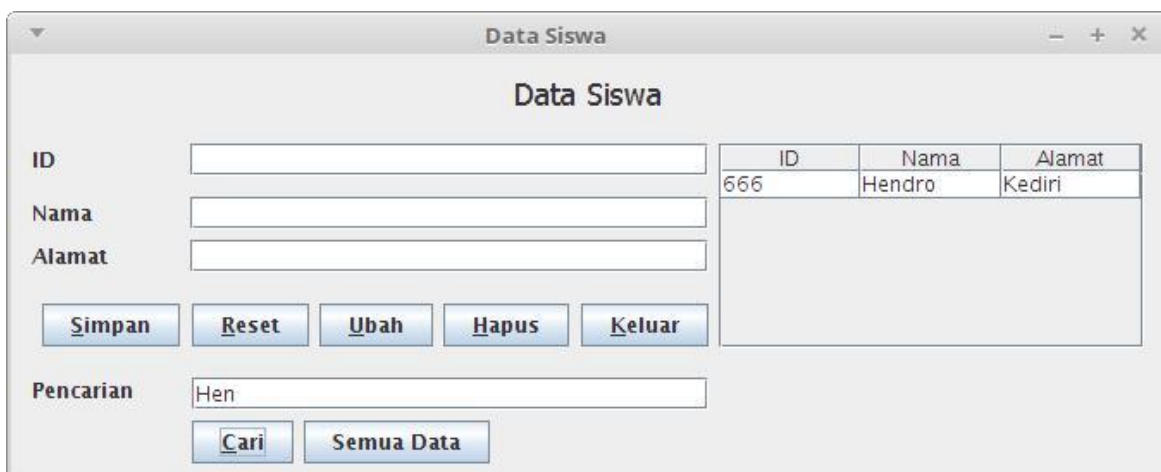
### Graphical User Interface (GUI) Java NetBeans & Aplikasi CRUD (Create, Read, Update, Delete) LATIHAN 1

1. Ubahlah method `simpanData`, `ubahData` dan `hapusData` pada class CRUD dari void menjadi String. Nilai String yang dikembalikan adalah pesan sukses jika data berhasil disimpan/diubah/dihapus atau pesan gagal jika data gagal disimpan/diubah/dihapus. String pesan sukses/gagal tadi akan ditangkap oleh `JOptionPane.showMessageDialog` pada class `Form_Siswa`.
2. Tambahkan Pencarian ID atau Nama pada Form Siswa dengan GUI sebagai berikut:



ID	Nama	Alamat
111	Anton	Blitar
222	Ratih	Malang
333	Setiawan	Mojokerto
444	Budi	Jombang
555	Parto	Nganjuk
666	Hendro	Kediri

Masukkan ID atau Nama Siswa, contoh “Hen”, pada pencarian lalu klik tombol “Cari” sehingga muncul di tabel siswa seperti berikut:



ID	Nama	Alamat
666	Hendro	Kediri

Query yang digunakan untuk pencarian “Hen” adalah sebagai berikut:

```
SELECT * FROM siswa WHERE id LIKE '%Hen%' OR nama LIKE '%Hen%'
```

Klik tombol “Semua Data” untuk menampilkan semua data seperti diawal ketika program dijalankan.

3.     Buatlah class Abstract yang diwarisi oleh class CRUD, dimana class Abstract tersebut memuat semua method setter dan getter pada class CRUD !
4.     Buatlah Interface yang diimplementasikan pada class CRUD, dimana Inteface tersebut memuat semua method yang berhubungan dengan database seperti method simpan/tampil/ubah/hapus/cari data pada class CRUD !