

PEMROGRAMAN BERORIENTASI OBJEK
SEMESTER GENAP 2022/2023

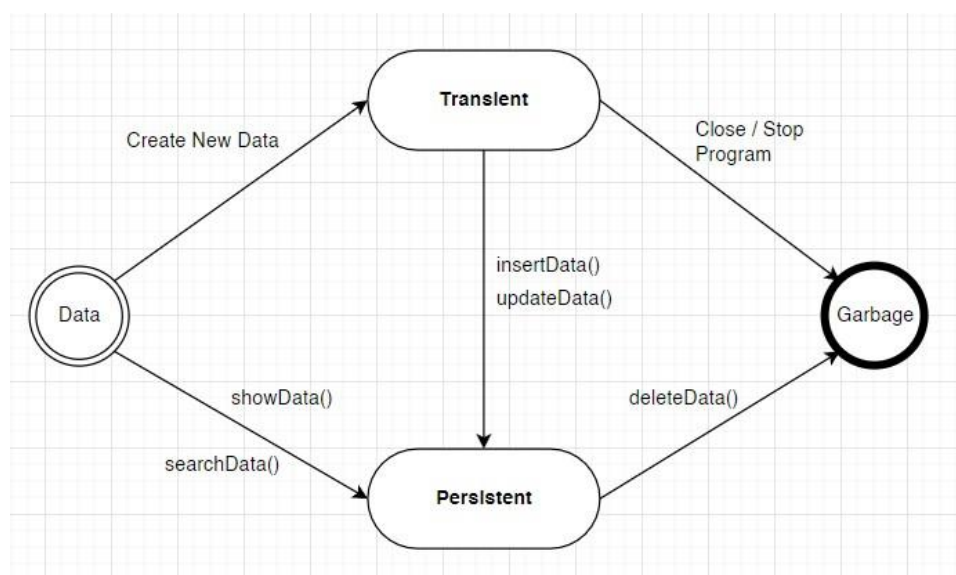


MODUL OBJECT PERSISTENCE
BAGIAN 1

PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ATMA JAYA YOGYAKARTA

Dasar Teori

Sejauh ini, kita telah membuat bermacam-macam program yang dapat menerima input dari user dan datanya akan tersimpan selama program berjalan atau sedang dalam status *running* untuk digunakan sesuai dengan keperluan. Ketika program berada dalam status *terminated*, maka data yang diinputkan tadi beserta hasil operasinya akan hilang. Hal ini dikarenakan data-data yang didapat pada saat program berstatus *running* disimpan pada Memori Primer, yaitu RAM. Bayangkanlah Objek yang kita pelajari seperti data di atas. Sebuah objek dapat menyimpan status dalam media penyimpanan. Berdasarkan media penyimpanannya, objek dibedakan menjadi *Transient Object* dan *Persistent Object*. *Transient Object* adalah objek yang menyimpan status (data) nya dalam penyimpanan primer (memori), di mana status objek akan hilang ketika program diterminasi atau berhenti berjalan. Sedangkan *Persistent Object* adalah objek yang menyimpan status (data) nya dalam penyimpanan sekunder (file, basis data), di mana status objek nya akan tetap ada ketika program sudah berhenti berjalan.



Gambar Ilustrasi Transient dan Persistent Object

Untuk mengembangkan aplikasi yang fungsional, penggunaan jenis Objek yang benar harus diperhatikan. Contohnya, tidak mungkin orang akan menggunakan aplikasi POS yang tidak bisa menyimpan data pembelian pada satu hari. Di sinilah Keunggulan dari *Persistent Object* dapat kita lihat. Dengan menerapkan Object Persistence, data atau status yang telah disimpan dalam media penyimpanan (storage), dapat diambil (restore) ketika aplikasi dijalankan kembali.

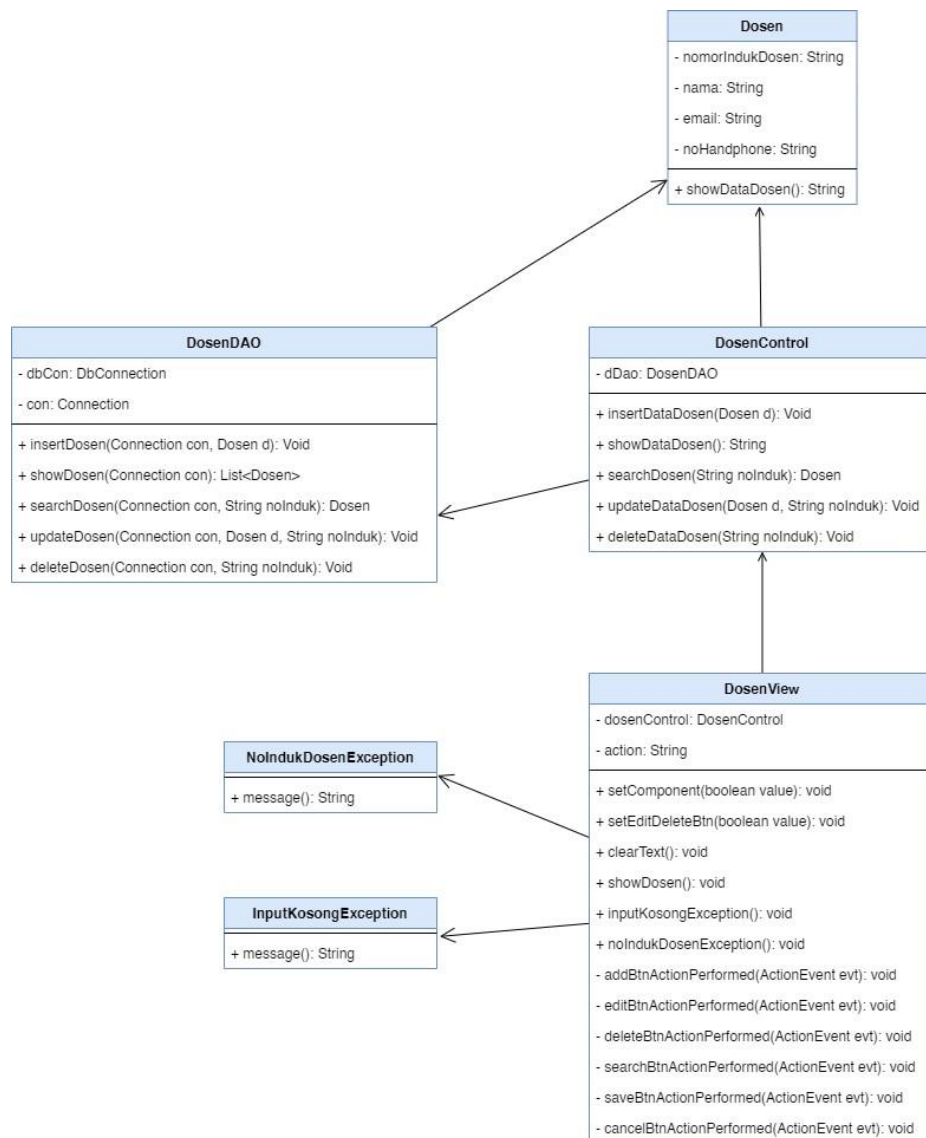
Guided

Ketentuan Pengumpulan Guided

Dikumpulkan dalam bentuk zip dengan nama GD9_X_YYYYY.zip dengan,

X = Kelas, YYYYYY = 5 digit NPM terakhir

Pastikan penamaan database dan tabel sama persis dengan yang ada pada Guided!



Gambar Class Diagram

dosen	
PK	nomor_induk_dosen varchar(5)
	nama varchar(255)
	email varchar(255)
	no_handphone varchar(14)

Gambar Tabel Dosen

Universitas Atma Jaya Yogyakarta ingin membuat sebuah sistem yang mampu mengolah data dosen dan mata kuliah. Sistem yang diberi nama RuangAtma ini akan dibangun dengan menggunakan bahasa pemrograman Java dan basis data MySQL. Pada guided kali ini, kalian akan dipandu untuk membuat basis data serta kode program untuk pengelolaan data dosen.

Data dosen yang ingin dikelola adalah nomorIndukDosen(String), nama(String), email(String) dan noHandphone(String). Dari data-data tersebut, buatlah sistem yang dapat memenuhi fitur-fitur seperti berikut :

- Menambahkan data dosen baru ke dalam sistem
- Menampilkan seluruh data-data dosen yang ada di dalam sistem
- Mencari data dosen yang ada di dalam sistem berdasarkan **nomor induk dosen**.
- Mengubah data dosen berdasarkan **nomor induk dosen**.
- Menghapus data dosen berdasarkan **nomor induk dosen**.

Seluruh data dalam sistem akan terhubung dengan basis data MySQL yang dimiliki oleh universitas sehingga data akan tetap tersedia jika program dijalankan ulang. Pihak universitas juga meminta untuk membuat exception untuk error handling dengan ketentuan semua inputan **tidak boleh kosong**, inputan nomor induk dosen **harus 5 digit**. Berikut adalah langkah-langkahnya :

1. Membuat database baru

- Buka XAMPP control panel dan hidupkan service Apache dan MySQL dengan menekan tombol “Start” untuk masing-masing module Apache maupun MySQL.
- Klik tombol “Admin” pada module MySQL untuk membuka phpMyAdmin.
- Klik “New” yang letaknya ada di sebelah kiri atas layar.

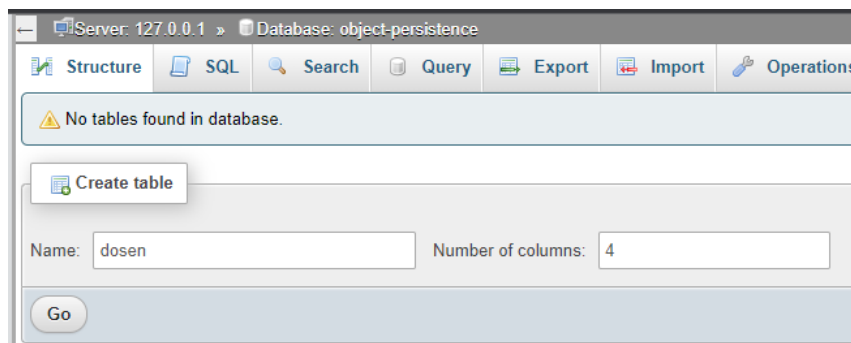


- Masukkan nama database “object-persistence” pada kolom database name lalu klik “Create”.

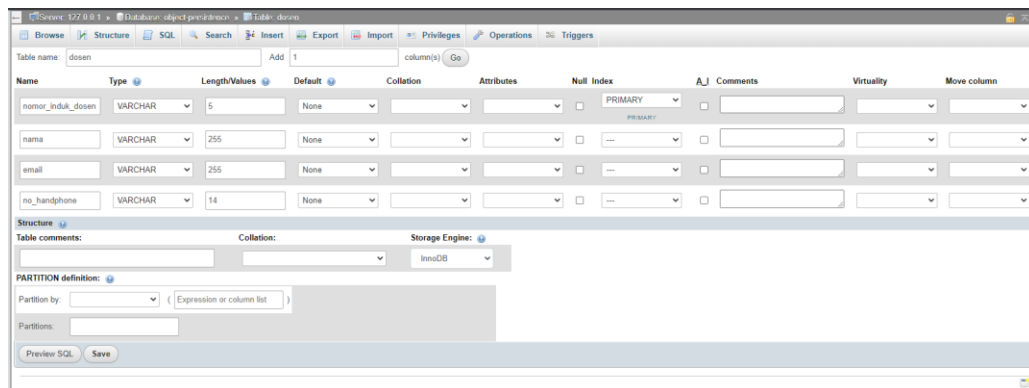


2. Membuat tabel dosen

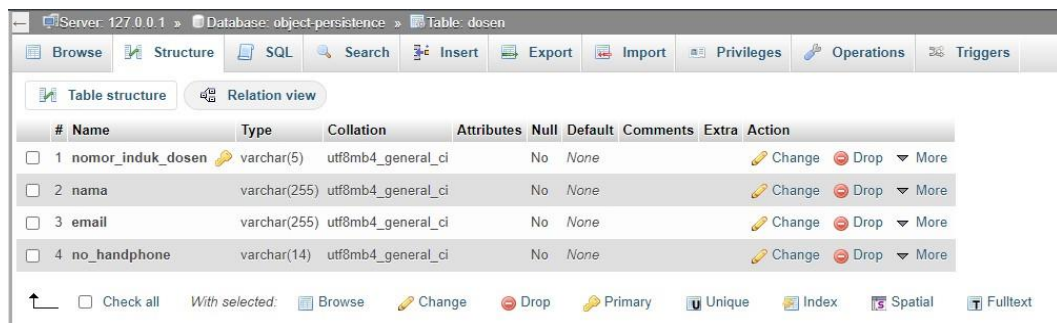
- Setelah itu masukkan nama tabel "dosen" pada kolom name lalu klik "Go".



- Lalu masukkan nama kolom beserta tipe sesuai pada diagram yang sudah diberikan sebelumnya. Setelah itu, untuk mengatur **nomor_induk_dosen** menjadi primary key, ubah nilai pada kolom **Index** pada baris nomor_induk_dosen, menjadi **primary** dan klik "Go" pada window yang muncul. Maka setelah itu pastikan tampilan menjadi seperti ini.

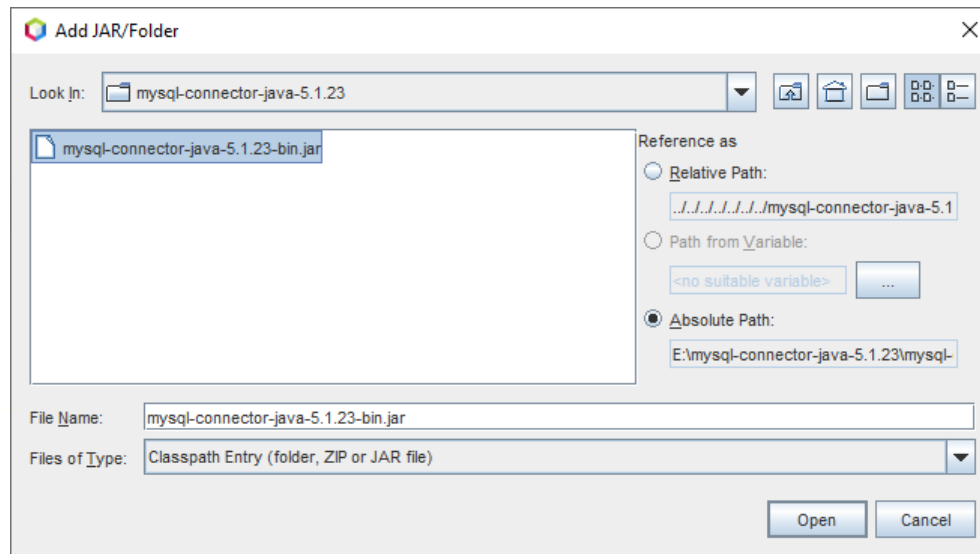


- Klik "Save" pada bagian kiri bawah dan tabel akan terbentuk.

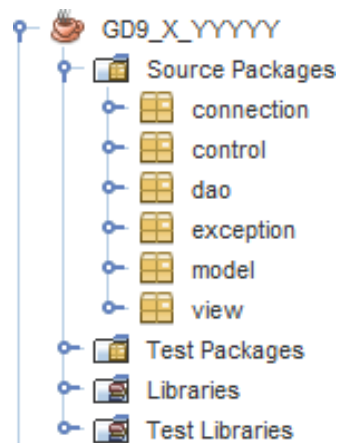


3. Menyiapkan project pada netbeans

- Buatlah sebuah project baru dengan nama GD9_X_YYYYY (X = Kelas, YYYYY = 5 digit terakhir NPM). Uncheck create main class (karena tidak akan digunakan dalam project kali ini).
- Import library untuk koneksi ke database. Klik kanan pada folder Libraries pada project yang baru dibuat lalu pilih Add JAR/Folder dan arahkan menuju library yang telah disediakan oleh asisten lalu klik “Open”.



- Langkah selanjutnya adalah membuat struktur package untuk memisahkan fungsi dari masing-masing kelas. Buat 6 package yang berbeda dalam 1 project yaitu **connection, dao, model, control, view, dan exception** dengan cara klik kanan pada Source Packages → New → Folder → beri nama → Finish.



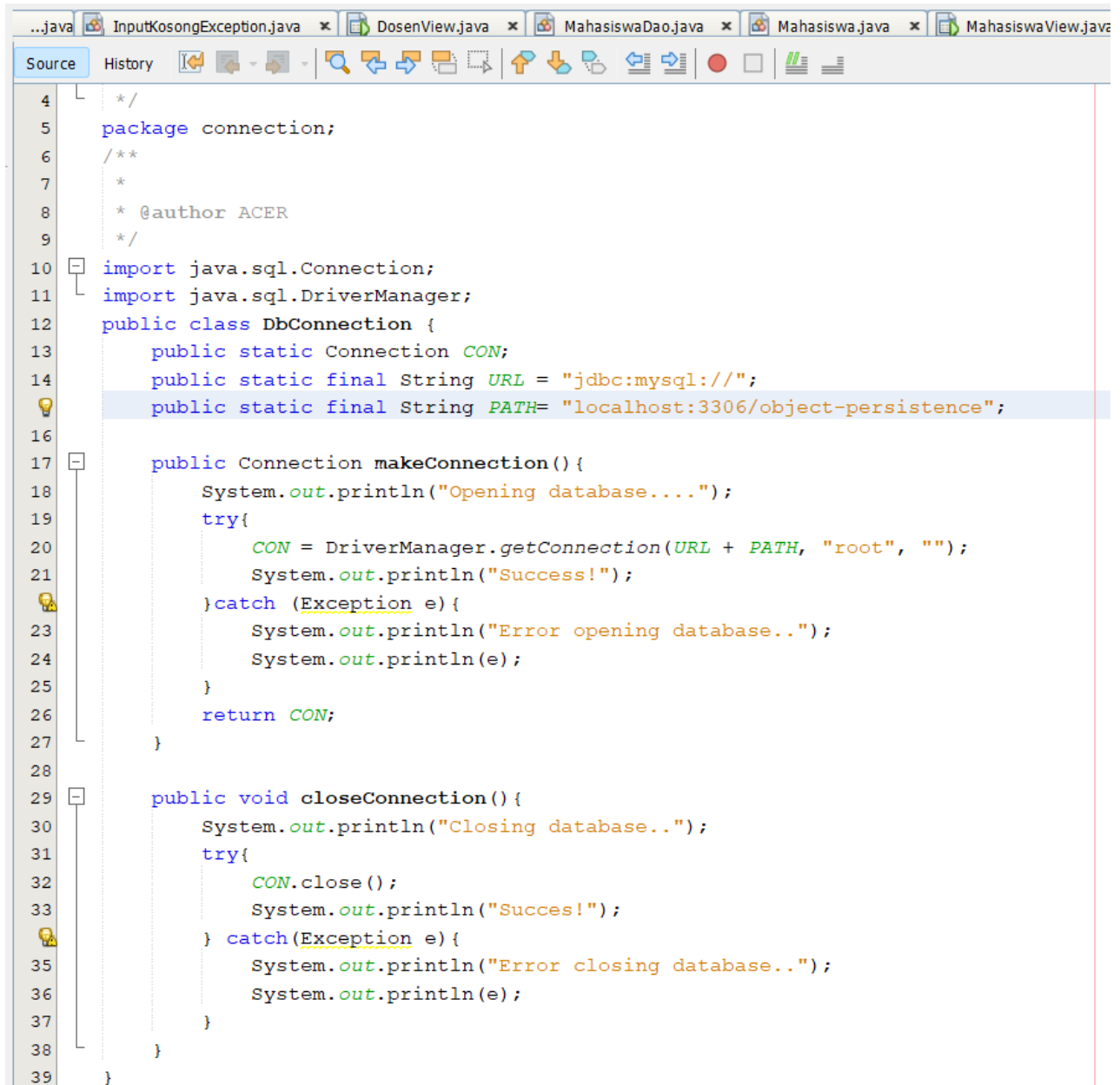
4. Membuat entity class

Entity class adalah kelas yang **merepresentasikan data** yang akan disimpan. Dalam konteks pembuatan aplikasi ini, entity class yang akan dibuat adalah kelas Dosen. Kelas dosen ini merupakan bagian dari package **model**. Berikut codenya:

```
Source History
1  /**
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5   package model;
6
7   /**
8    *
9    * @author ACER
10   */
11  public class Dosen {
12      private String nomorIndukDosen;
13      private String nama;
14      private String email;
15      private String noHandphone;
16
17      public Dosen(String nomorIndukDosen, String nama, String email, String noHandphone){
18          this.nomorIndukDosen=nomorIndukDosen;
19          this.nama=nama;
20          this.email=email;
21          this.noHandphone=noHandphone;
22      }
23
24      public String getNomorIndukDosen(){
25          return nomorIndukDosen;
26      }
27
28      public void setNomorIndukDosen(String nomorIndukDosen){
29          this.nomorIndukDosen=nomorIndukDosen;
30      }
31
32      public String getNama(){
33          return nama;
34      }
35
36      public void setNama(String nama){
37          this.nama=nama;
38      }
39
40      public String getEmail(){
41          return email;
42      }
43
44      public void setEmail(String email){
45          this.email=email;
46      }
47
48      public String getNoHandphone(){
49          return noHandphone;
50      }
51
52      public void setNoHandphone(String noHandphone){
53          this.noHandphone=noHandphone;
54      }
55
56      public String showDataDosen(){
57          return
58              this.nomorIndukDosen + " | "
59              + this.nama + " | "
60              + this.email + " | "
61              +this.noHandphone;
62      }
63  }
64
```

5. Membuat koneksi ke database

Untuk membuat koneksi ke database, buatlah class baru pada package **connection** dan beri nama **DbConnection**. Setelah itu buat method `makeConnection` dan `closeConnection` yang berfungsi untuk membuka dan menutup koneksi ke database dengan masukkan kode seperti berikut ini :



```
4  */
5  package connection;
6  /**
7   *
8   * @author ACER
9   */
10 import java.sql.Connection;
11 import java.sql.DriverManager;
12 public class DbConnection {
13     public static Connection CON;
14     public static final String URL = "jdbc:mysql://";
15     public static final String PATH= "localhost:3306/object-persistence";
16
17     public Connection makeConnection() {
18         System.out.println("Opening database....");
19         try{
20             CON = DriverManager.getConnection(URL + PATH, "root", "");
21             System.out.println("Success!");
22         } catch (Exception e) {
23             System.out.println("Error opening database..");
24             System.out.println(e);
25         }
26         return CON;
27     }
28
29     public void closeConnection() {
30         System.out.println("Closing database..");
31         try{
32             CON.close();
33             System.out.println("Success!");
34         } catch (Exception e) {
35             System.out.println("Error closing database..");
36             System.out.println(e);
37         }
38     }
39 }
```

Ketika membuat koneksi, yang perlu diperhatikan adalah pada variabel **path**. Dimana path ini memiliki struktur **localhost:X/Y** (X = port MySQL yang dapat dilihat pada XAMPP control panel dan Y = nama database pada phpMyAdmin).

Salah satu cara menyimpan data dari program Java ke basis data yaitu dengan menggunakan Java Database Connectivity (JDBC). JDBC memiliki kemampuan untuk melakukan koneksi pada basis data, mengeksekusi statement SQL dan menerima hasil query. Analoginya bisa dibayangkan seperti driver printer untuk menggunakan sebuah printer melalui komputer. Driver JDBC sendiri merupakan koleksi class-class Java yang dikumpulkan dalam satu atau beberapa file .jar. Perlu diperhatikan bahwa setiap database memerlukan JDBC yang berbeda.

6. Membuat Data Access Object Class (DAO Class)

Data Access Object (DAO) digunakan untuk memisahkan data access logic dari business logic dan presentation logic. DAO dapat menambah **reusability** dan **maintainability** dari data. DAO digunakan untuk melakukan operasi akses data ke basis data. Berikut adalah langkah-langkahnya:

- Buatlah class baru pada package dao dengan nama DosenDAO. Setelah itu, import beberapa library yang akan digunakan dalam pembuatan dao.

```
package dao;
```

```
import connection.DbConnection;
import java.sql.Connection;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
import model.Dosen;
```

- Kemudian buatlah objek baru dari class DbConnection serta objek Connection karena kita perlu mengakses database untuk setiap method pada class DosenDAO.

```
public class DosenDAO {
    private DbConnection dbCon = new DbConnection();
    private Connection con;
```

- Langkah selanjutnya adalah membuat method untuk menambahkan data dosen ke dalam tabel dosen yang ada pada database, sesuai dengan atribut yang telah dibuat dari tabel dosen dengan query insert data seperti gambar berikut.

```
// Method untuk menyimpan data dosen
public void insertDosen(Dosen d) {
    con = dbCon.makeConnection();

    String sql = "INSERT INTO dosen(nomor_induk_dosen, nama, email, no_handphone) "
        + "VALUES ('" + d.getNomorIndukDosen() + "', '" + d.getNama() + "', '"
        + d.getEmail() + "', '" + d.getNoHandphone() + "')";

    System.out.println("Adding Dosen...");

    try {
        Statement statement = con.createStatement();
        int result = statement.executeUpdate(sql);
        System.out.println("Added " + result + " Dosen");
        statement.close();
    } catch (Exception e) {
        System.out.println("Error adding Dosen...");
        System.out.println(e);
    }
    dbCon.closeConnection();
}
```

- Lanjutkan dengan membuat method untuk menampilkan semua data dari tabel dosen pada database.

```
// Method untuk mengambil semua data dosen
public List<Dosen> showDosen() {
    con = dbCon.makeConnection();

    String sql = "SELECT * FROM dosen";
    System.out.println("Mengambil data Dosen...");

    List<Dosen> list = new ArrayList();

    try {
        Statement statement = con.createStatement();
        ResultSet rs = statement.executeQuery(sql);

        if (rs != null) {
            while (rs.next()) {
                Dosen d = new Dosen(
                    rs.getString("nomor_induk_dosen"),
                    rs.getString("nama"),
                    rs.getString("email"),
                    rs.getString("no_handphone")
                );
                list.add(d);
            }
        }
        rs.close();
        statement.close();
    } catch (Exception e) {
        System.out.println("Error reading database...");
        System.out.println(e);
    }
    dbCon.closeConnection();

    return list;
}
```

- Selain mengambil semua data, program juga mampu untuk menampilkan data berdasarkan nomor induk dosen. Maka buatlah method untuk menampilkan data dosen berdasarkan nomor induk dosen tertentu.

```
// Method untuk menampilkan dosen berdasarkan nomor induk
public Dosen searchDosen(String noInduk) {
    con = dbCon.makeConnection();

    String sql = "SELECT * FROM dosen WHERE nomor_induk_dosen = '" + noInduk + "'";
    System.out.println("Searching Dosen...");
    Dosen d = null;

    try {
        Statement statement = con.createStatement();
        ResultSet rs = statement.executeQuery(sql);

        if (rs != null) {
            while (rs.next()) {
                d = new Dosen(
                    rs.getString("nomor_induk_dosen"),
                    rs.getString("nama"),
                    rs.getString("email"),
                    rs.getString("no_handphone")
                );
            }
            rs.close();
            statement.close();
        } catch (Exception e) {
            System.out.println("Error reading database...");
            System.out.println(e);
        }
        dbCon.closeConnection();

        return d;
    }
}
```

- Berikutnya buatlah method untuk mengubah data dosen berdasarkan nomor induk dosen. Data pada tabel dosen masing-masing nilainya akan diubah berdasarkan inputan pengguna.

```
// Method untuk mengubah data dosen
public void updateDosen(Dosen d, String noInduk) {
    con = dbCon.makeConnection();

    String sql = "UPDATE dosen SET nama = '" + d.getNama() + "', "
        + "email = '" + d.getEmail() + "', "
        + "no_handphone = '" + d.getNoHandphone() + "' "
        + "WHERE nomor_induk_dosen = '" + noInduk + "'";
    System.out.println("Editing Dosen...");

    try {
        Statement statement = con.createStatement();
        int result = statement.executeUpdate(sql);
        System.out.println("Edited " + result + " Dosen " + noInduk);
        statement.close();
    } catch (Exception e) {
        System.out.println("Error editing dosen...");
        System.out.println(e);
    }
    dbCon.closeConnection();
}
```

- Method terakhir yang perlu dibuat adalah method untuk menghapus data dosen berdasarkan nomor induk dosen.

```
// Method untuk menghapus data dosen
public void deleteDosen(String noInduk) {
    con = dbCon.makeConnection();

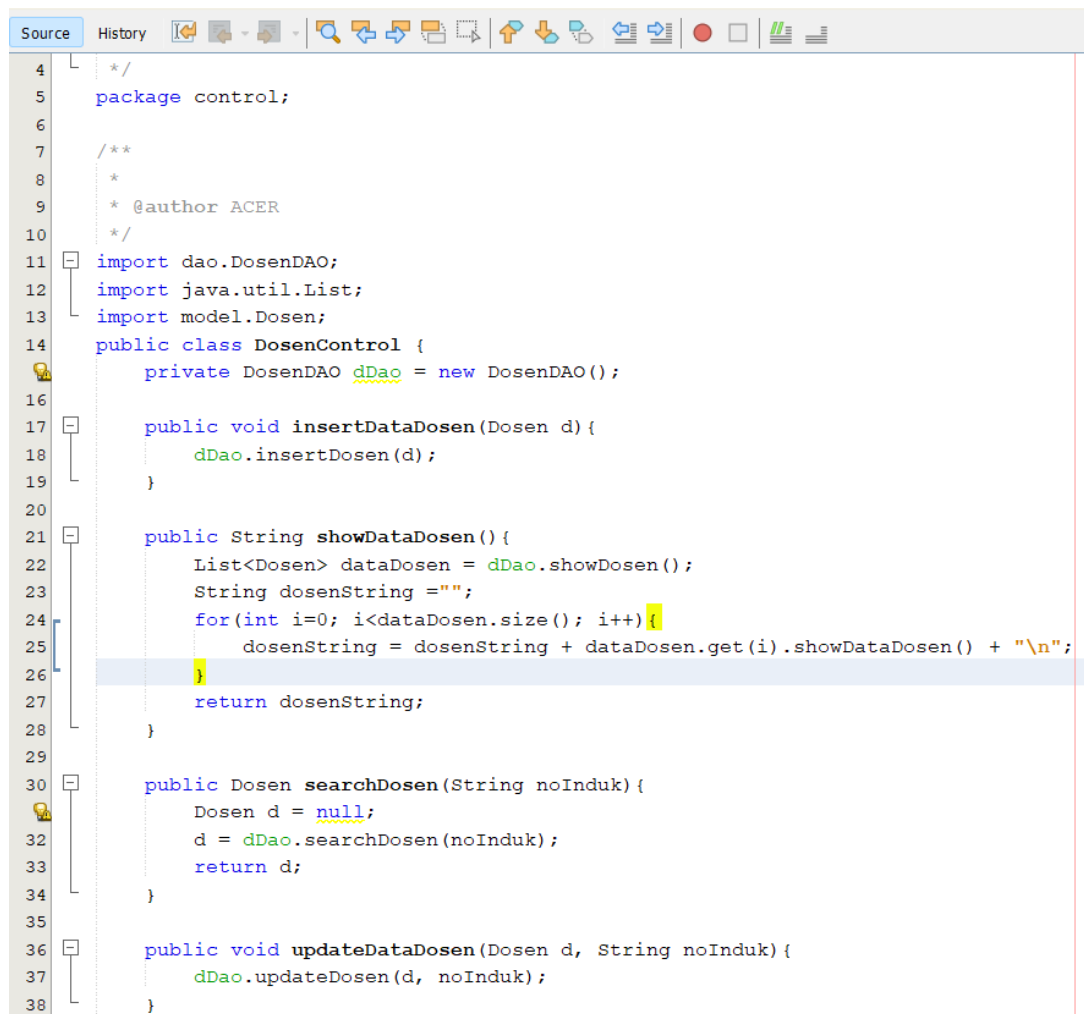
    String sql = "DELETE FROM dosen WHERE nomor_induk_dosen = '" + noInduk + "'";
    System.out.println("Deleting Dosen...");

    try {
        Statement statement = con.createStatement();
        int result = statement.executeUpdate(sql);
        System.out.println("Delete " + result + " Dosen " + noInduk);
        statement.close();
    } catch (Exception e) {
        System.out.println("Error deleting dosen...");
        System.out.println(e);
    }
    dbCon.closeConnection();
}
```

Apabila diperhatikan, ketika hendak melakukan perubahan pada basis data (seperti insert, update dan delete) maka digunakan `statement.executeUpdate()` yang memiliki tipe balikan berupa integer. Sedangkan ketika proses pengambilan data digunakan `statement.executeQuery()` yang memiliki tipe balikan berupa `ResultSet`.

7. Membuat control class

Control class merupakan layer yang berisi tentang business logic dari sebuah aplikasi. Hal pertama yang harus dilakukan setelah membuat class baru pada package control dengan nama DosenControl adalah mengimport package dao dan model yang akan digunakan. Langkah selanjutnya adalah membuat body dari kelas DosenControl sehingga keseluruhan codenya menjadi seperti ini :



```
4  */
5  package control;
6
7  /**
8   *
9   * @author ACER
10  */
11  import dao.DosenDAO;
12  import java.util.List;
13  import model.Dosen;
14  public class DosenControl {
15      private DosenDAO dDao = new DosenDAO();
16
17      public void insertDataDosen(Dosen d) {
18          dDao.insertDosen(d);
19      }
20
21      public String showDataDosen() {
22          List<Dosen> dataDosen = dDao.showDosen();
23          String dosenString = "";
24          for(int i=0; i<dataDosen.size(); i++){
25              dosenString = dosenString + dataDosen.get(i).showDataDosen() + "\n";
26          }
27          return dosenString;
28      }
29
30      public Dosen searchDosen(String noInduk) {
31          Dosen d = null;
32          d = dDao.searchDosen(noInduk);
33          return d;
34      }
35
36      public void updateDataDosen(Dosen d, String noInduk) {
37          dDao.updateDosen(d, noInduk);
38      }
39  }
```

```
...ava InputKosongException.java x DosenView.java x MahasiswaDao.java x Mahasiswa.java x MahasiswaView.java
Source History
13 import model.Dosen;
14 public class DosenControl {
15     private DosenDAO dDao = new DosenDAO();
16
17     public void insertDataDosen(Dosen d) {
18         dDao.insertDosen(d);
19     }
20
21     public String showDataDosen() {
22         List<Dosen> dataDosen = dDao.showDosen();
23         String dosenString = "";
24         for(int i=0; i<dataDosen.size(); i++){
25             dosenString = dosenString + dataDosen.get(i).showDataDosen() + "\n";
26         }
27         return dosenString;
28     }
29
30     public Dosen searchDosen(String noInduk) {
31         Dosen d = null;
32         d = dDao.searchDosen(noInduk);
33         return d;
34     }
35
36     public void updateDataDosen(Dosen d, String noInduk) {
37         dDao.updateDosen(d, noInduk);
38     }
39
40     public void deleteDataDosen(String noInduk) {
41         dDao.deleteDosen(noInduk);
42     }
43 }
44
```

Secara keseluruhan, pada masing-masing method akan melakukan pemanggilan terhadap method yang ada pada class DosenDAO dengan membawa beberapa data yang dibutuhkan ke database pada parameter.

8. Membuat exception class

Langkah selanjutnya adalah mengimplementasikan exception handling untuk menangani error ketika program dijalankan dengan membuat kelas baru pada package exception.

- Buatlah kelas InputKosongException lalu masukkan kode seperti berikut :

```
5      package exception;
6
7      /**
8       *
9       * @author ACER
10     */
11     public class InputKosongException extends Exception{
12         public String message() {
13             return "FIELD INPUT TIDAK BOLEH KOSONG!";
14         }
15     }
16
```

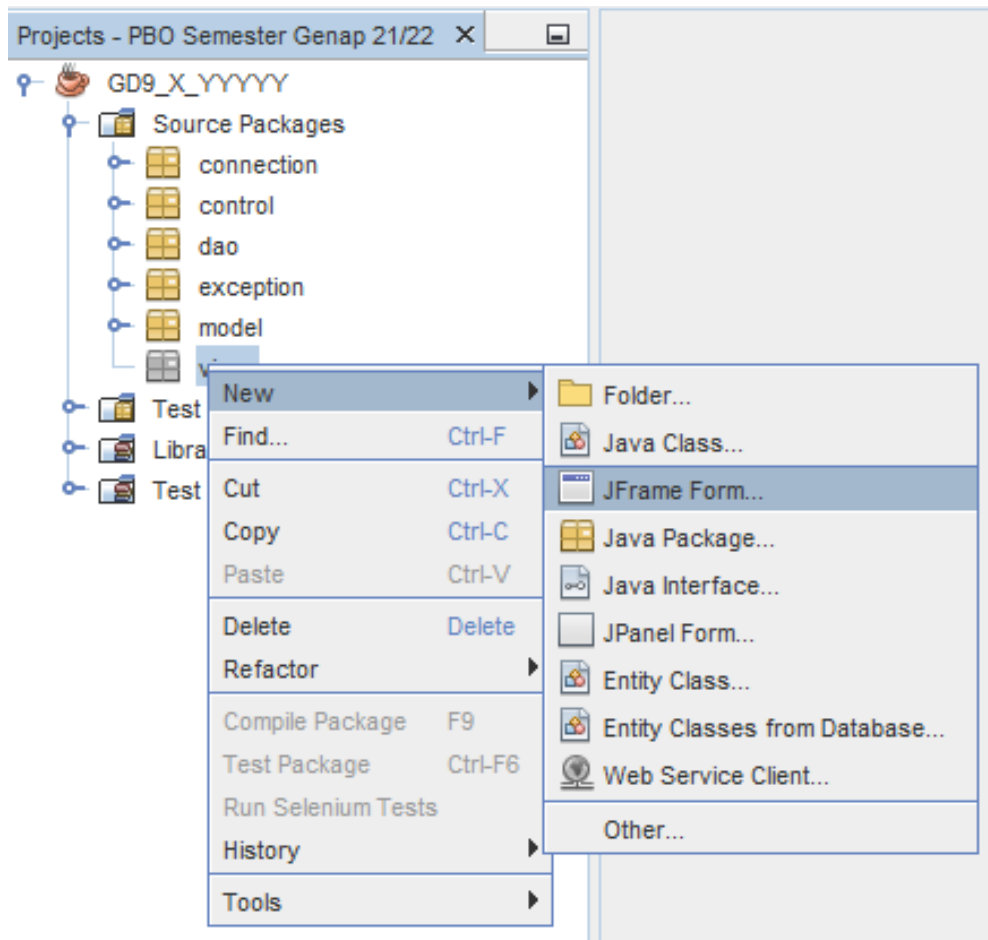
- Buatlah kelas NoIndukDosenException lalu masukkan kode seperti berikut :

```
5      package exception;
6
7      /**
8       *
9       * @author ACER
10     */
11     public class NoIndukDosenException extends Exception{
12         public String message() {
13             return "Nomor Induk dosen harus terdiri dari 5 karakter!";
14         }
15     }
16
```

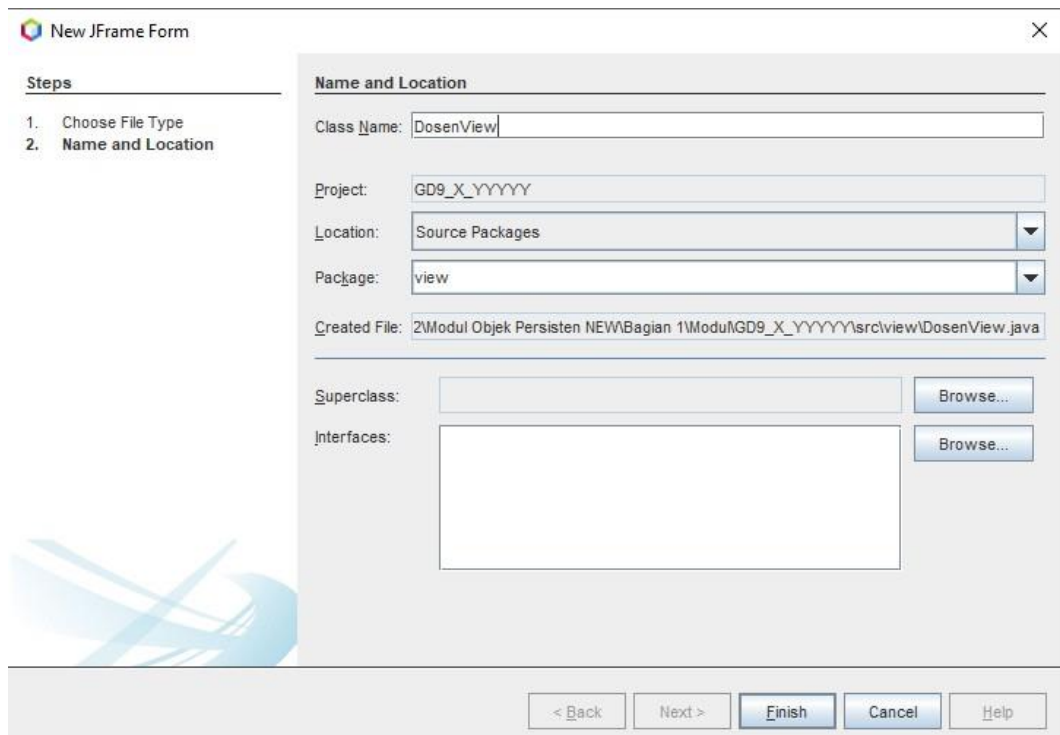
9. Membuat user interface class

User interface merupakan bagian presentation logic sebuah aplikasi yang digunakan sebagai antarmuka kepada user. Kita akan masuk ke pembuatan UI. Berikut adalah langkah-langkahnya:

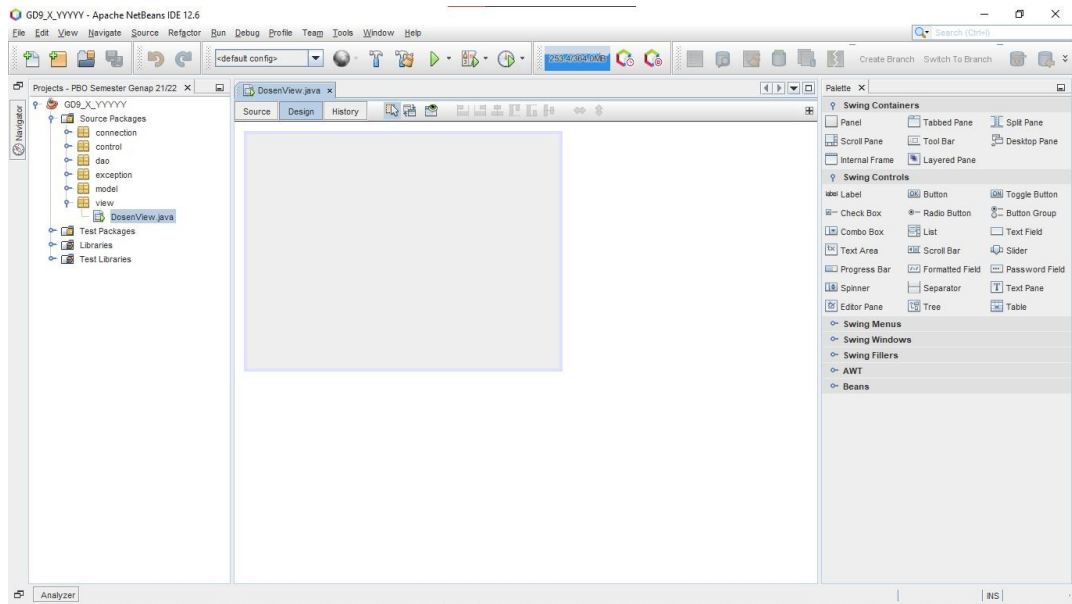
- Buatlah class baru dengan tipe JFrameForm pada package view seperti terlihat pada gambar di bawah ini



- Pada windows yang muncul berilah nama class tersebut dengan nama DosenView, lalu klik Finish



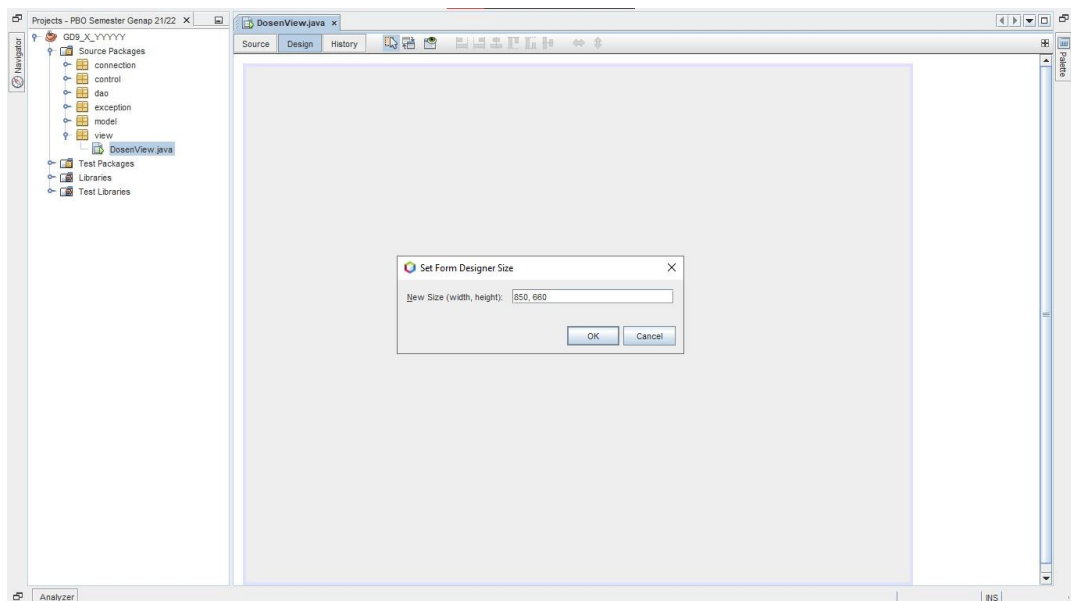
- Setelah itu maka akan muncul halaman baru seperti gambar di bawah ini



Proses pembuatan UI dapat dibagi menjadi 2 tahap, pertama styling atau mendesain UI, kedua mengintegrasikan logic dengan UI yang telah dibuat.

I. Styling desain komponen UI

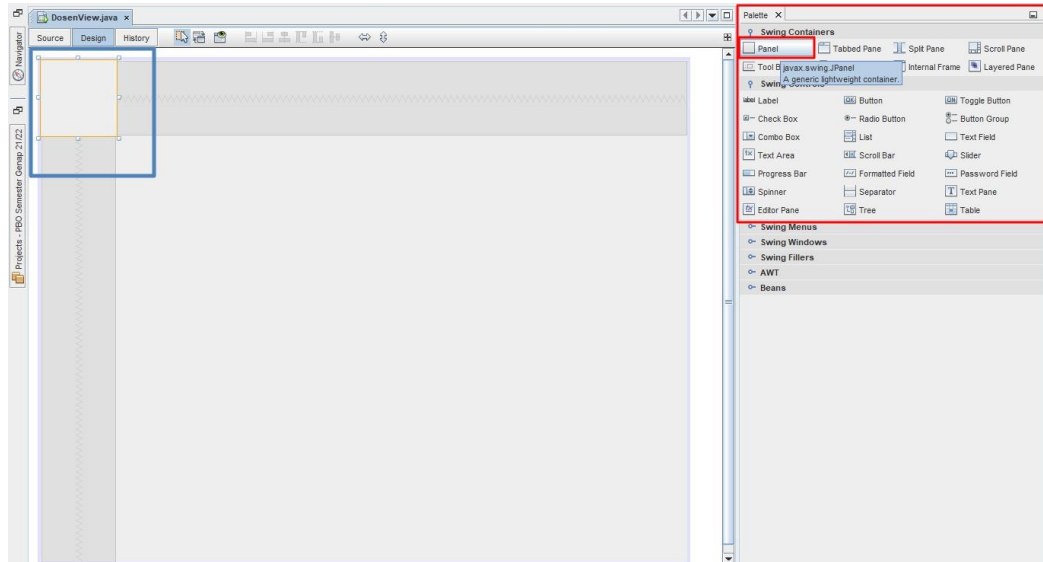
Sebelum masuk lebih dalam silahkan atur ukuran frame menjadi 850 x 660 dengan cara drag pada bagian sisi atas dan bawahnya, atau double click pada bordernya.



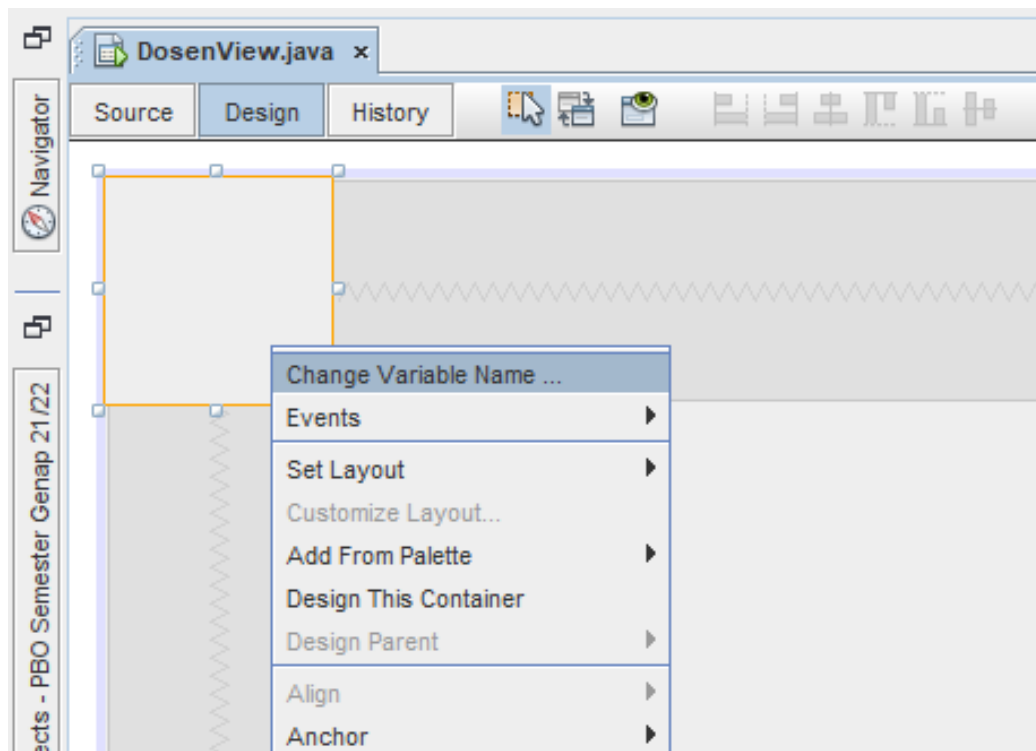
Ikuti langkah-langkah berikut ini secara teliti untuk mendesain tampilan UI.

PEMBUATAN SIDEBAR

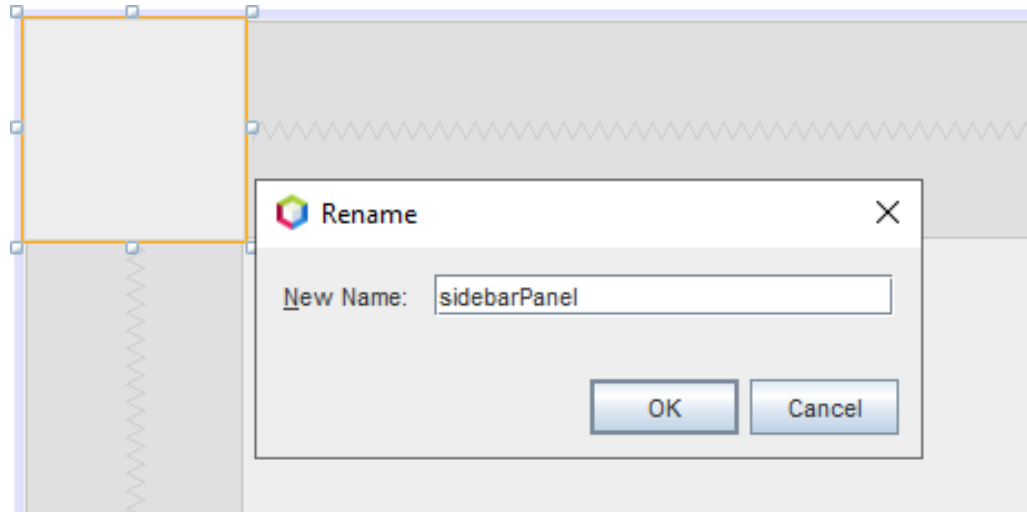
- a. Pada bagian pertama kita akan membuat komponen sidebar. Pada bagian Palette klik komponen Panel (perhatikan kotak merah) dan drag ke dalam JFrame kalian (perhatikan kotak biru).



- b. Pada JPanel tersebut, silahkan ganti nama variabel dengan cara klik kanan pada JPanel pilih Change variable name...



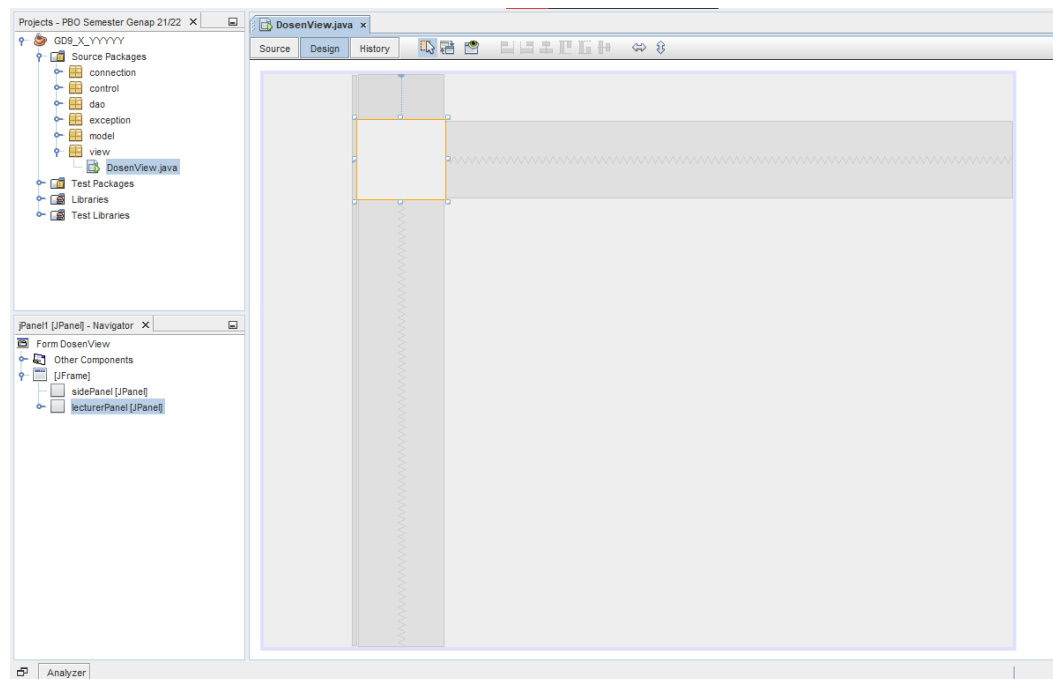
- c. Setelah itu akan muncul sebuah window dan gantilah nama variabel dengan nama sidebarPanel. **Lakukan penggantian variabel setiap kali menambahkan komponen baru dengan tujuan memudahkan mengenali komponen yang kalian buat ketika membuat logic dari UI.**



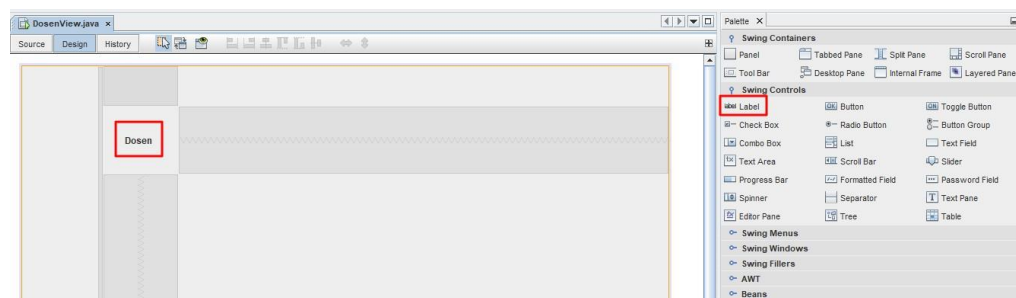
- d. Setelah itu letakan sidebarPanel di sebelah kiri JFrame dan set ukuran menjadi 100 x 660 (panjang sama dengan JFrame).



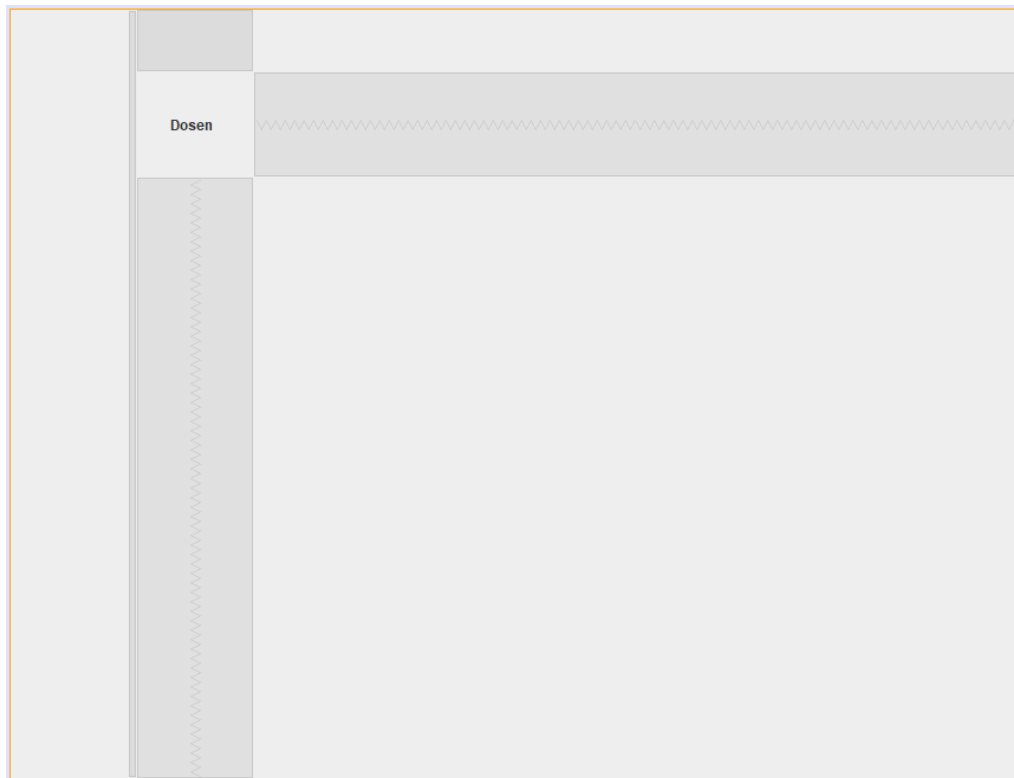
- e. Selanjutnya kita akan membuat menu yang ada pada sidebarPanel. Tambah komponen panel dan beri nama variabel `lecturerPanel`. Setelah itu set ukurannya menjadi 100 x 90 hingga tertampil seperti ini.



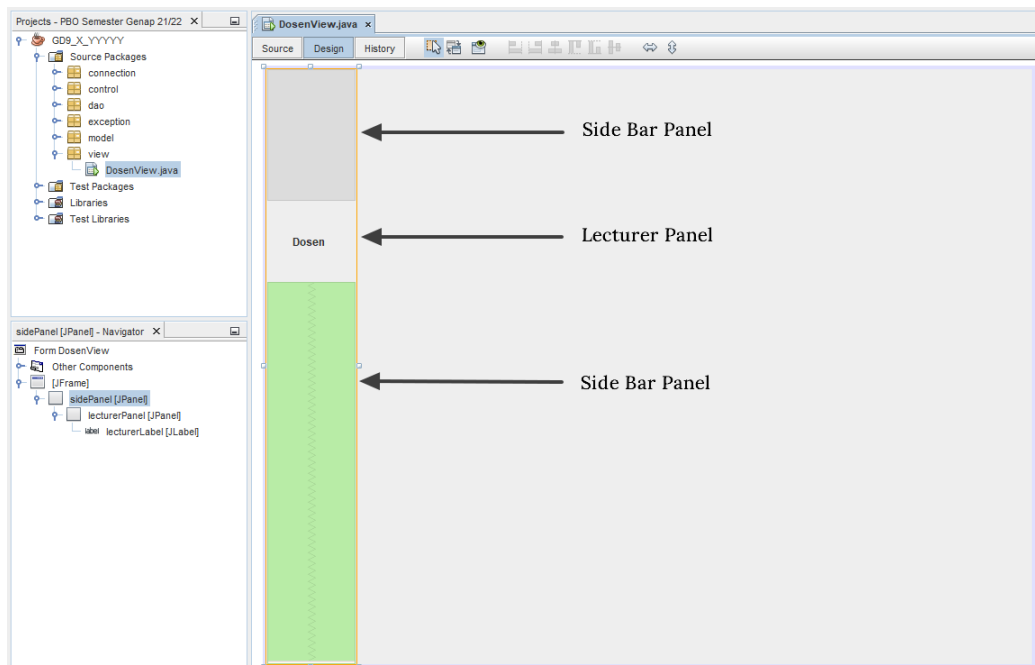
- f. Tambahkan komponen Label ke dalam komponen `lecturerPanel` dan ubah nama variabel label menjadi `lecturerLabel`. Atur panjang `lecturerLabel` menjadi sama dengan `lecturerPanel` dan ganti isi teks label dengan "Dosen".



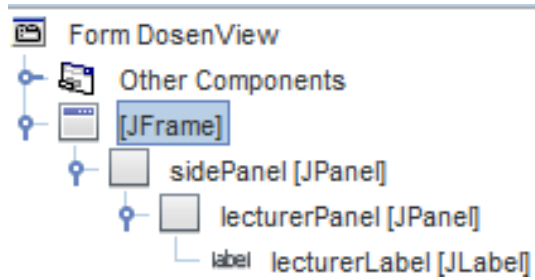
Pastikan saat ini kalian memiliki tampilan yang sama seperti gambar di bawah ini.



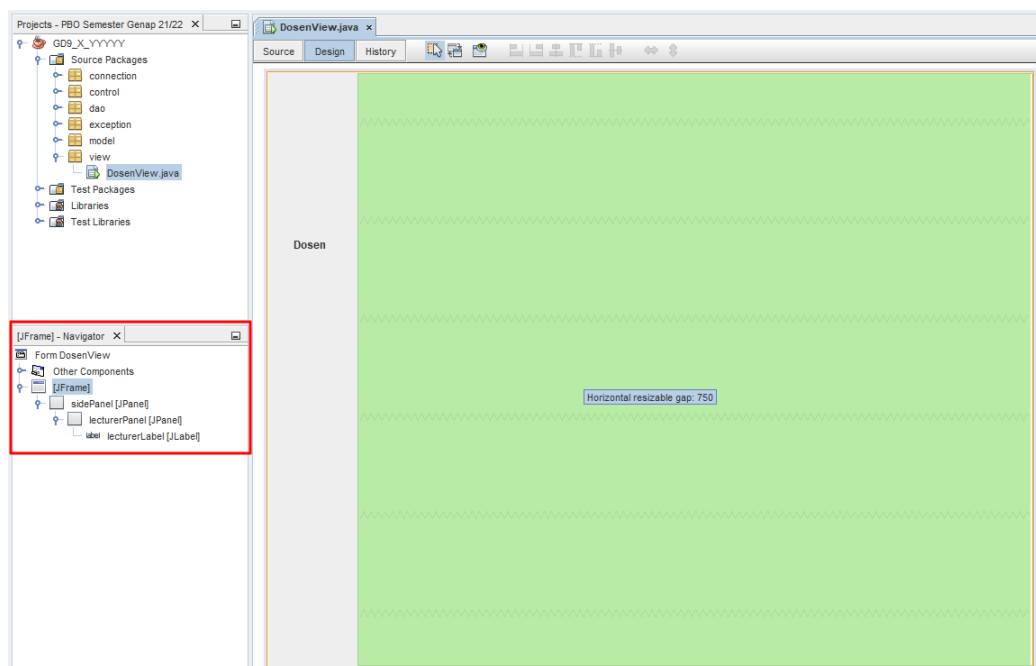
- g. Masukkan komponen `lecturerPanel` ke dalam komponen `sidebarPanel` dan posisikan sedemikian rupa seperti pada gambar di bawah ini.



h. Pastikan struktur urutan komponen UI kalian sama seperti gambar di bawah ini

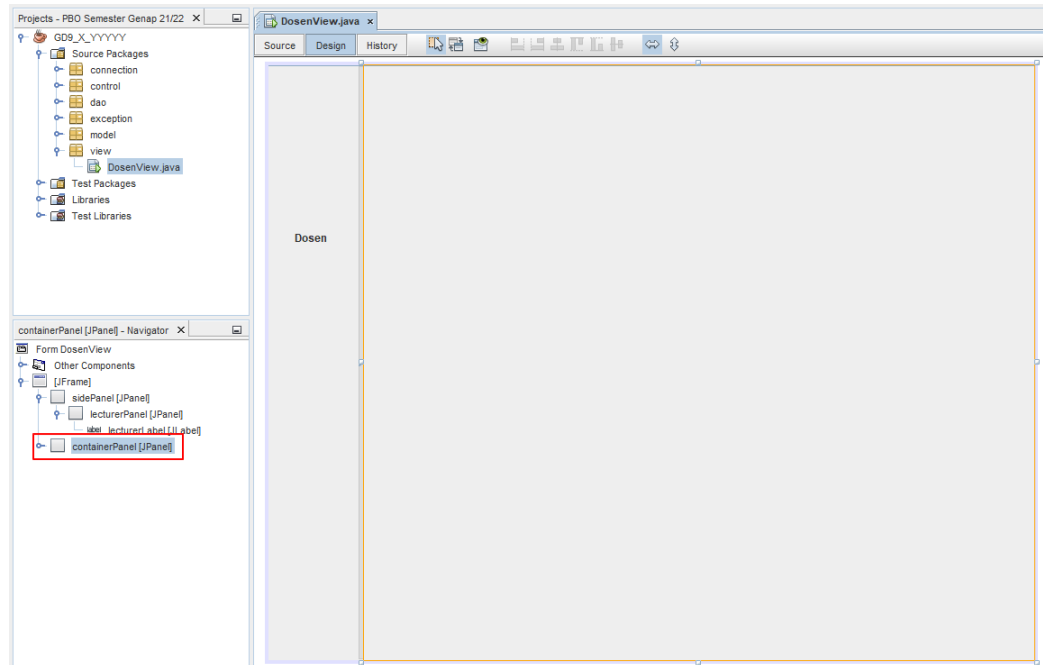


i. Kalian juga dapat mengubah struktur urutan setiap komponen dengan melakukan *drag and drop* pada bagian navigator.



PEMBUATAN KONTEN

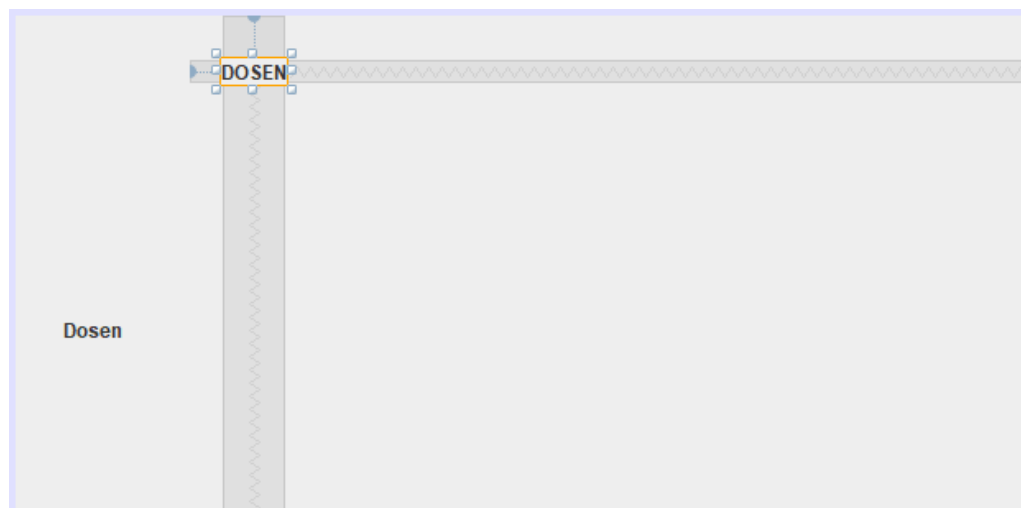
- a. Pada bagian kedua kita akan membuat komponen untuk konten. Pada bagian Palette, pilih komponen Panel dan beri nama variabel `containerPanel`. Aturlah ukurannya untuk mengisi sisa ruang dari JFrame.



- b. Tambahkan Label ke dalam panel `containerPanel` sebagai judul menu dan sesuai dengan aturan di bawah ini.

Nama variabel : `titleContent`

Text : DOSEN



- c. Tambahkan 3 komponen button ke dalam `containerPanel` dengan aturan di bawah ini.

Button-1

Nama variabel : `addBtn`

Text : Tambah

Button-2

Nama variabel : `editBtn`

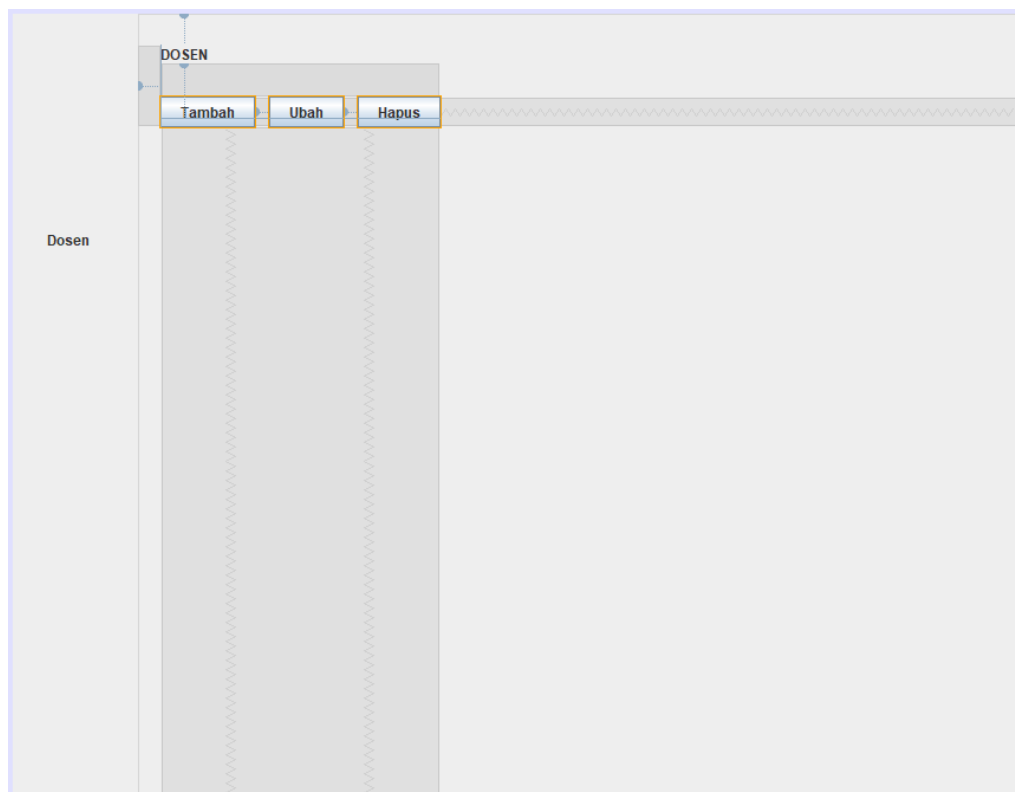
Text : Ubah

Button-3

Nama variabel : `deleteBtn`

Text : Hapus

Pastikan tampilan dan struktur komponen sudah sesuai dengan gambar di bawah ini



- d. Tambahkan komponen Button dan komponen Text Field ke dalam `containerPanel` dengan aturan di bawah ini.

Button

Nama variabel : `searchBtn`

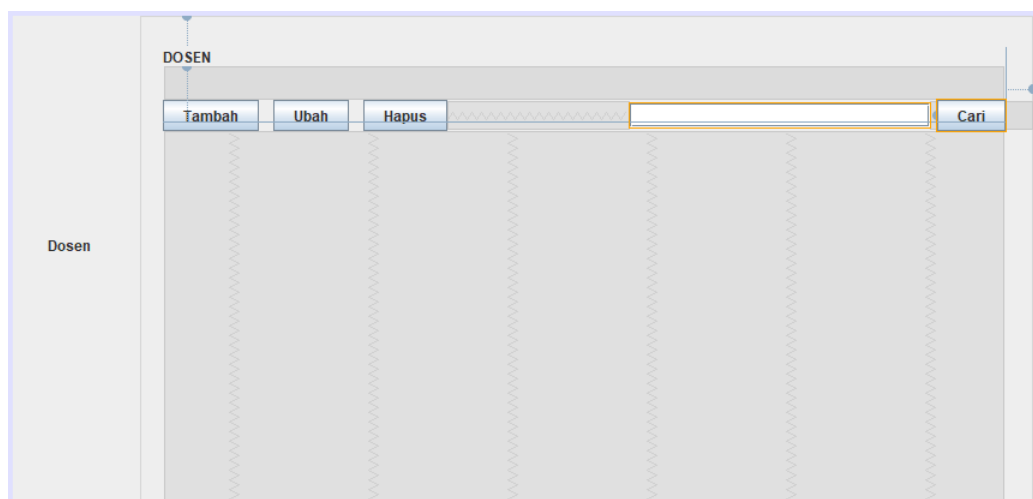
Text : Cari

Text Field

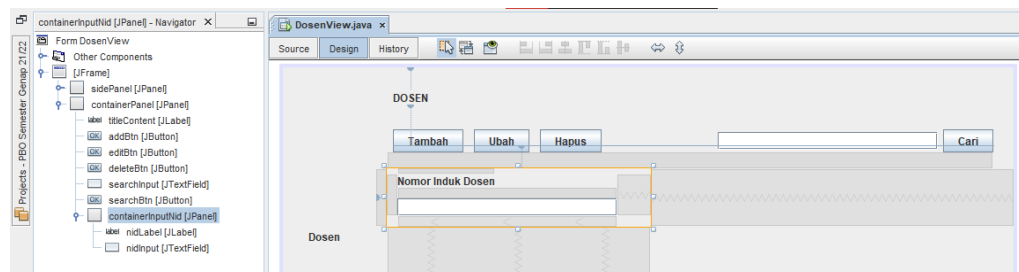
Nama variabel : `inputSearch`

Text :

Pastikan tampilan dan struktur komponen sudah sesuai dengan gambar di bawah ini



- e. Tambahkan Panel baru bernama `containerInputNID` ke dalam komponen `containerPanel` dan set ukuran menjadi 300 x 65. Kemudian Tambahkan komponen Label dan komponen Text Field ke dalam komponen `containerInputNID` dan beri text “Nomor Induk Dosen” serta posisikan seperti gambar berikut ini



Silahkan *copy* dan *paste* komponen `containerInputNID` dan buatlah komponen baru dengan aturan berikut ini.

a. Komponen 1 (NID)

Nama Container : `containerInputNID`

Nama variabel Label : `nidLabel`

Nama variabel Input : `nidInput`

b. Komponen 2 (Nama)

Nama Container : `containerInputNama`

Nama variabel Label : `namaLabel`

Nama variabel Input : `namaInput`

c. Komponen 3 (Email)

Nama Container : `containerInputEmail`

Nama variabel Label : `emailLabel`

Nama variabel Input : `emailInput`

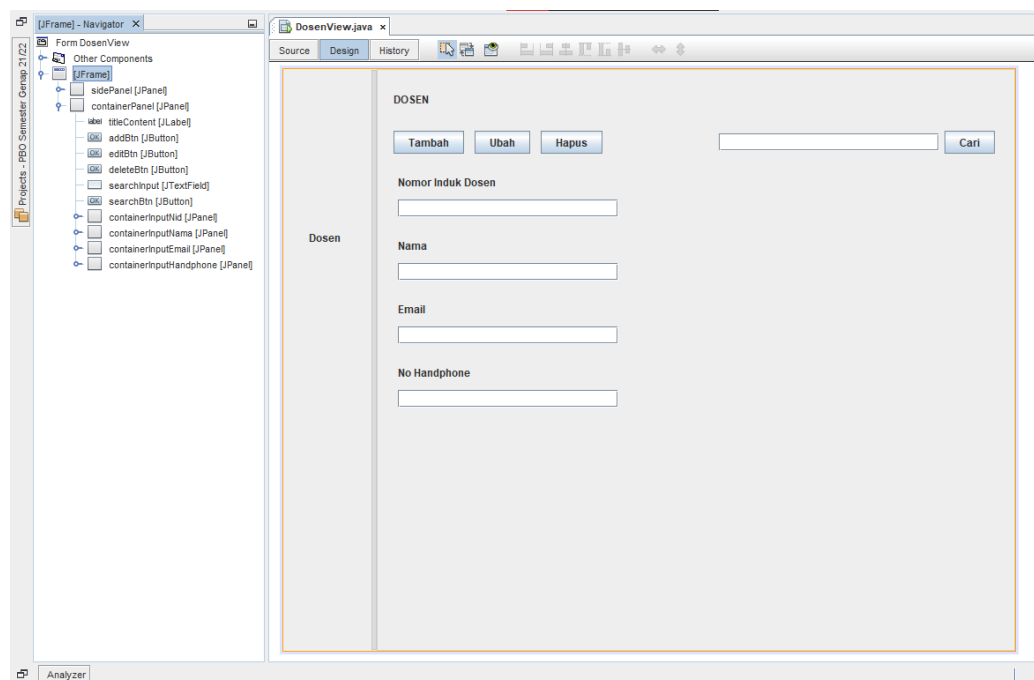
d. Komponen 4 (Handphone)

Nama Container : `containerInputHandphone`

Nama variabel Label : `handphoneLabel`

Nama variabel Input : `handphoneInput`

Pastikan tampilan dan struktur komponen sudah sesuai dengan gambar di bawah ini



- f. Selanjutnya tambahkan 2 button di sebelah kanan dengan aturan seperti di bawah ini.

Button-1

Nama variabel : saveBtn

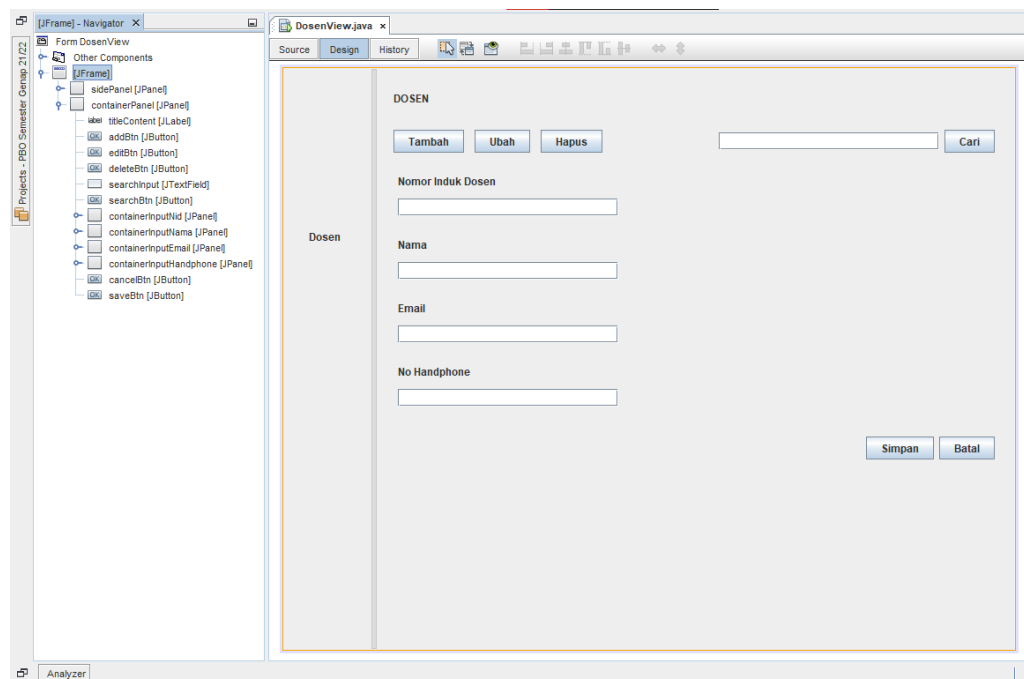
Text : Simpan

Button-2

Nama variabel : cancelBtn

Text : Batal

Pastikan tampilan dan struktur komponen sudah sesuai dengan gambar di bawah ini

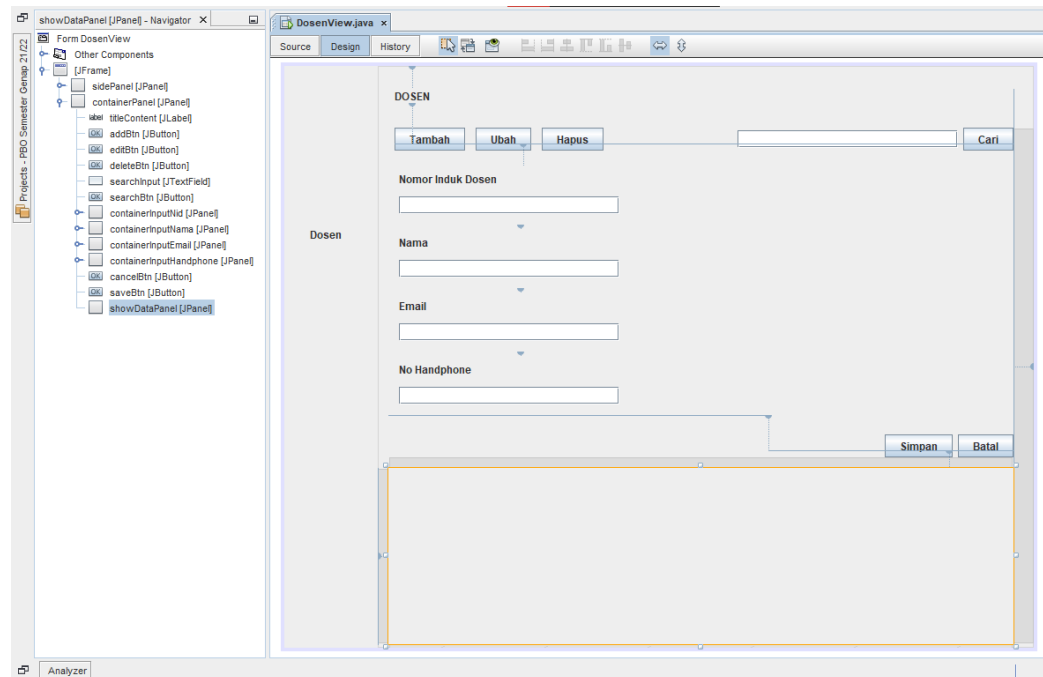


- g. Tambahkan komponen Panel ke dalam `containerPanel` dengan aturan berikut ini.

Nama variabel : `showDataPanel`

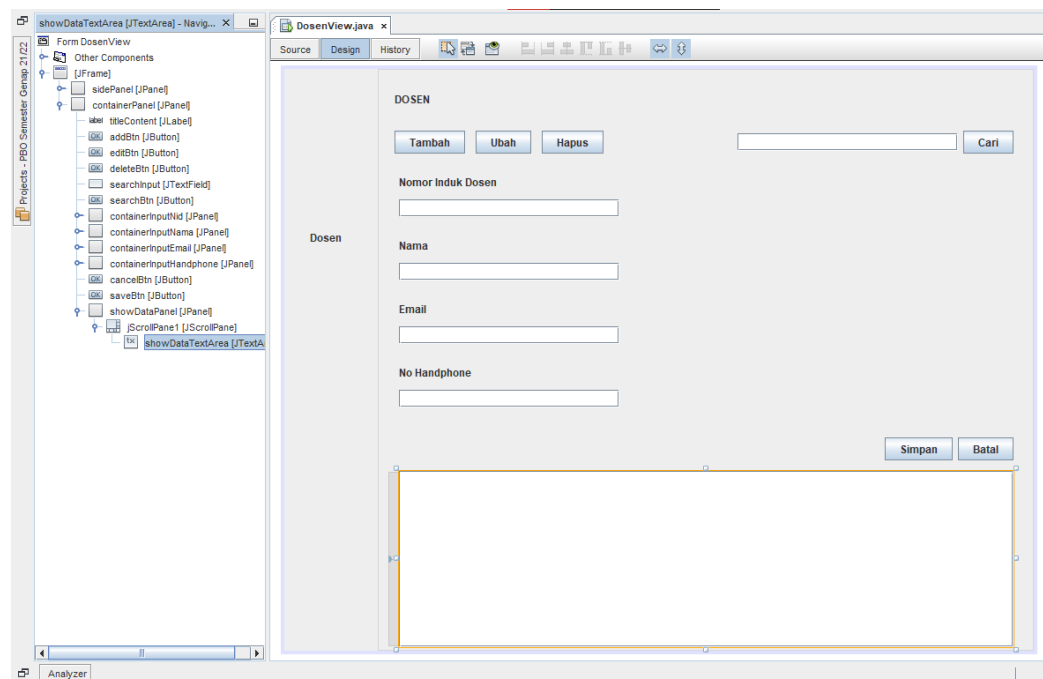
Width : Sesuaikan dengan perataan kanan kiri terhadap komponen di atasnya

Height : 200

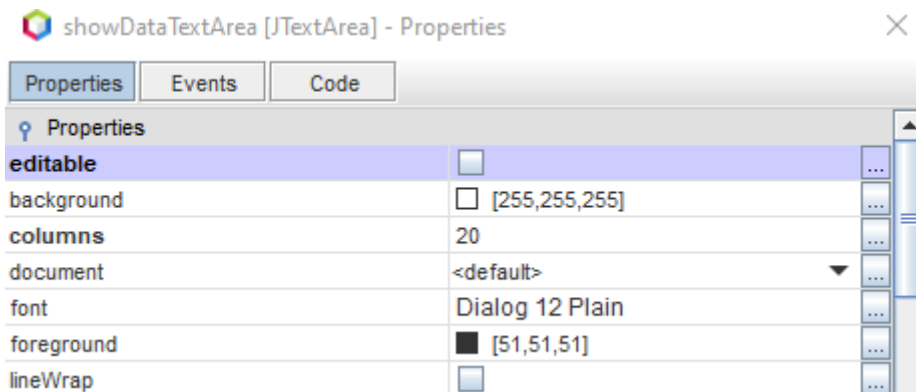


- h. Tambahkan komponen `TextArea` ke dalam komponen `showDataPanel` dengan aturan berikut ini.

Nama variabel : `showDataTextArea`

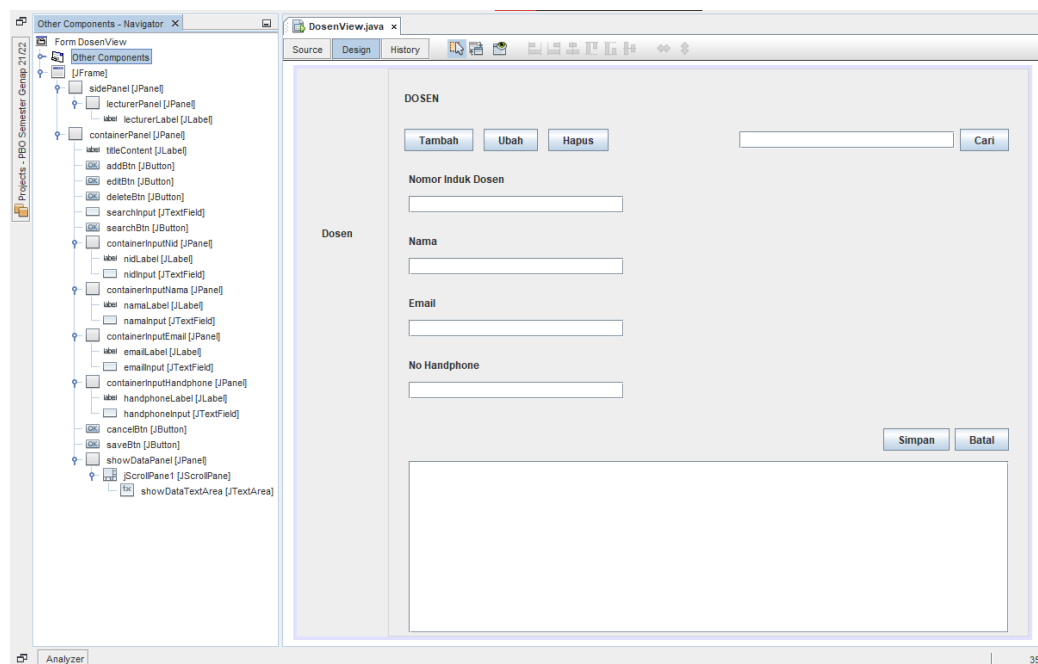


Klik kanan pada panel `showDataTextArea` dan klik properties. Lalu uncheck kolom editable.

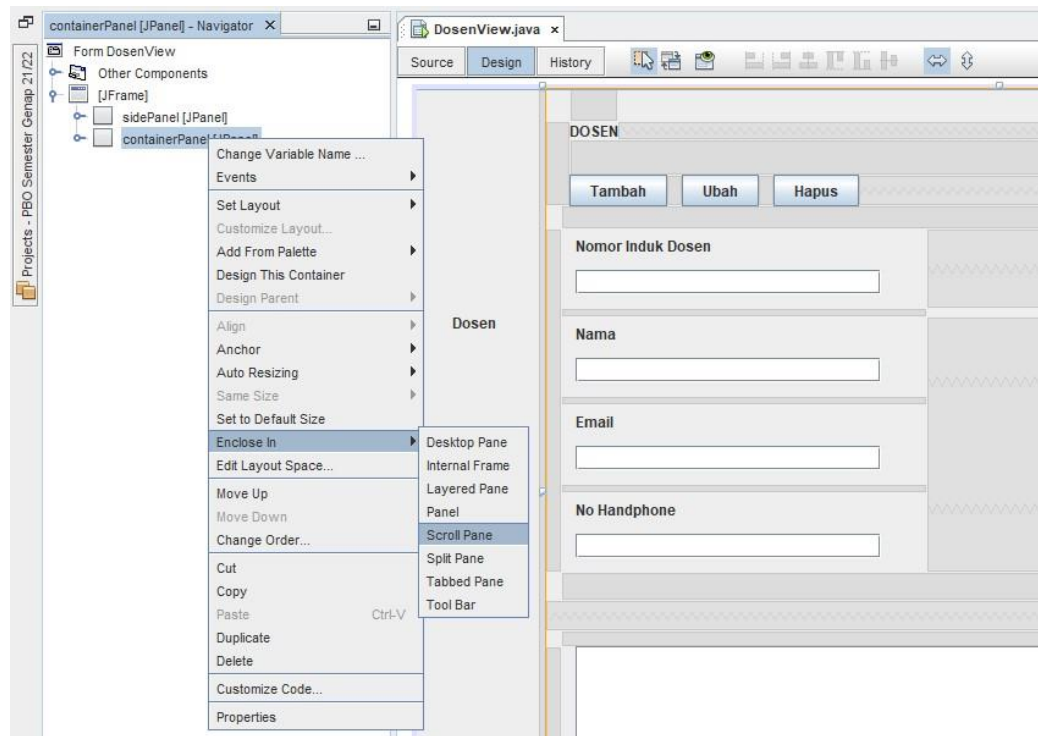


Setting ukuran panjang dan lebar komponen `showDataTextArea` memenuhi komponen `showDataPanel`.

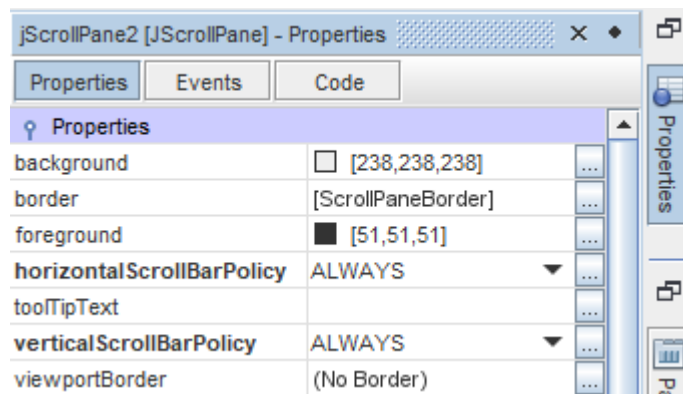
Pastikan tampilan dan struktur komponen sudah sesuai dengan gambar di bawah ini



Untuk langkah terakhir, klik `containerPanel` pada struktur komponen di bagian kiri bawah layar. Klik kanan, lalu klik **Enclose In** lalu **Scroll Pane**. Hal ini dilakukan untuk menambahkan scroll bar, supaya kita tetap dapat melihat konten di dalam form, walaupun ukuran form melebihi layar.



Lalu pada bagian kanan layar akan muncul bagian properties. Ubah nilai dari **horizontalScrollBarPolicy** dan **verticalScrollBarPolicy** menjadi **ALWAYS**.



II. Integrasi logic dengan UI

Setelah selesai membuat tampilan UI, maka langkah selanjutnya adalah melakukan “coding” untuk menghubungkan business logic dan presentation logic. Klik tab “Source” pada bagian kiri atas DosenView.



a. Import library dan kelas yang dibutuhkan :

```
package view;

import control.DosenControl;
import exception.InputKosongException;
import exception.NoIndukDosenException;
import javax.swing.JOptionPane;
import model.Dosen;
```

b. Lakukan deklarasi variabel dan pembuatan object DosenControl pada constructor:

```
public class DosenView extends javax.swing.JFrame {
    private DosenControl dosenControl;
    String action = null;

    /**
     * Creates new form DosenView
     */
    public DosenView() {
        initComponents();
        setComponent(false);
        setEditDeleteBtn(false);
        dosenControl = new DosenControl();
        showDosen();
    }
}
```

c. Tambahkan method seperti pada gambar berikut ini, untuk mematikan (disable) beberapa komponen UI yang tidak diperlukan saat program dijalankan serta method untuk membersihkan semua inputan.

```
// Method untuk mengeset komponen input dan button save, button cancel
public void setComponent(boolean value) {
    nidInput.setEnabled(value);
    namaInput.setEnabled(value);
    emailInput.setEnabled(value);
    handphoneInput.setEnabled(value);

    saveBtn.setEnabled(value);
    cancelBtn.setEnabled(value);
}

// Method untuk mengeset komponen button edit dan button delete
public void setEditDeleteBtn(boolean value) {
    editBtn.setEnabled(value);
    deleteBtn.setEnabled(value);
}

// Method untuk mengeset value text dari komponen Input
public void clearText() {
    nidInput.setText("");
    namaInput.setText("");
    emailInput.setText("");
    handphoneInput.setText("");
    searchInput.setText("");
}
```

- d. Tambahkan method `showDosen` untuk menampilkan data dosen pada text area.

```
// Method untuk menampilkan data dosen pada text area
public void showDosen() {
    showDataTextArea.setText(dosenControl.showDataDosen());
}
```

- e. Tambahkan juga method yang bernama `inputKosongException` serta `noIndukDosenException` untuk mengimplementasikan User Defined Exception yang telah dibuat sebelumnya pada package Exception. Fungsi dari `inputKosongException` adalah untuk memeriksa dari setiap field input memiliki inputan atau tidak. Begitu pula untuk `noIndukDosenException` dimana method ini akan melakukan pemeriksaan apabila nomor induk dosen yang dimasukkan terdiri dari 5 karakter atau tidak. Berikut adalah potongan kodenya :


```
// Method untuk memeriksa input kosong
public void inputKosongException() throws InputKosongException {
    if (nidInput.getText().isEmpty() || namaInput.getText().isEmpty()
        || emailInput.getText().isEmpty()
        || handphoneInput.getText().isEmpty()) {
        throw new InputKosongException();
    }
}

// Method untuk memeriksa nomor induk dosen
public void noIndukDosenException() throws NoIndukDosenException {
    if (nidInput.getText().length() != 5) {
        throw new NoIndukDosenException();
    }
}
```

- f. Klik 2x pada tombol addBtn, maka akan muncul tampilan seperti di bawah ini dan ketikkan code berikut :

```
private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // Ketika button 'Tambah' diklik, maka akan mengaktifkan component Input
    setComponent(true);
    clearText();
    searchInput.setText("");
    action = "Tambah";
}
```

- g. Klik 2x pada tombol editBtn, maka akan muncul tampilan seperti di bawah ini dan ketikkan code berikut :

```
private void editBtnActionPerformed(java.awt.event.ActionEvent evt) {
    // Ketika button 'Ubah' diklik, maka akan mengaktifkan component Input
    setComponent(true);
    nidInput.setEnabled(false);
    action = "Ubah";
}
```

- h. Klik 2x pada tombol deleteBtn, maka akan muncul tampilan seperti di bawah ini dan ketikkan code berikut:

```
private void deleteBtnActionPerformed(java.awt.event.ActionEvent evt) {
    int getAnswer = JOptionPane.showConfirmDialog(rootPane,
        "Apakah yakin ingin menghapus data ?", "Konfirmasi",
        JOptionPane.YES_NO_OPTION);
    //Jika jawaban user adalah Yes
    if (getAnswer == JOptionPane.YES_OPTION) {
        try {
            dosenControl.deleteDataDosen(nidInput.getText());
            clearText();
            showDosen();
            JOptionPane.showMessageDialog(null, "Data berhasil dihapus!");
        } catch (Exception e) {
            System.out.println("Error : " + e.getMessage());
        }
    }
    //Jika jawaban user adalah No
    else{JOptionPane.showMessageDialog(null, "Data tidak jadi dihapus!");}
}
```

- i. Klik 2x pada tombol searchBtn, maka akan muncul tampilan seperti di bawah ini dan ketikkan code berikut :

```
private void searchBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    // Ketika button 'Cari' diklik, maka akan mengaktifkan component Button Edit dan Delete  
    setEditDeleteBtn(true);  
    // Ketika button 'Cari' diklik, maka akan menonaktifkan component Input  
    setComponent(false);  
  
    try {  
        Dosen dosen = dosenControl.searchDosen(searchInput.getText());  
        if (dosen == null) {  
            //Ketika data yang dicari tidak ditemukan  
            clearText();  
            setEditDeleteBtn(false);  
            JOptionPane.showConfirmDialog(null, "Data tidak ditemukan",  
                "Konfirmasi", JOptionPane.DEFAULT_OPTION);  
        } else {  
            //Ketika data yang dicari ditemukan  
            nidInput.setText(dosen.getNomorIndukDosen());  
            namaInput.setText(dosen.getNama());  
            emailInput.setText(dosen.getEmail());  
            handphoneInput.setText(dosen.getNoHandphone());  
        }  
    } catch (Exception e) {  
        System.out.println("Error: " + e.getMessage());  
    }  
}
```

- j. Klik 2x pada tombol saveBtn, maka akan muncul tampilan seperti di bawah ini dan ketikkan code berikut :

```
private void saveBtnActionPerformed(java.awt.event.ActionEvent evt) {  
    try {  
        inputKosongException();  
        noIndukDosenException();  
  
        Dosen d = new Dosen(nidInput.getText(), namaInput.getText(),  
            emailInput.getText(), handphoneInput.getText());  
        if (action.equals("Tambah")) {  
            //Ketika nilai action = "Tambah" maka akan memanggil fungsi insert  
            dosenControl.insertDataDosen(d);  
        } else {  
            //Ketika nilai action = "Ubah" maka akan memanggil fungsi update  
            dosenControl.updateDataDosen(d, nidInput.getText());  
        }  
        clearText();  
        showDosen();  
        setComponent(false);  
        setEditDeleteBtn(false);  
    } catch (InputKosongException e) {  
        JOptionPane.showMessageDialog(this, e.message());  
    } catch (NoIndukDosenException e) {  
        JOptionPane.showMessageDialog(this, e.message());  
    }  
}
```

- k. Klik 2x pada tombol cancelBtn, maka akan muncul tampilan seperti di bawah ini dan ketikkan code berikut :

```
private void cancelBtnActionPerformed(java.awt.event.ActionEvent evt) {
    setComponent(false);
    setEditDeleteBtn(false);
    clearText();
}
```

Selamat! Kalian telah menyelesaikan modul pertama untuk Object Persistence. Cobalah untuk menjalankan program dan mencoba semua fungsionalitas. Perlu diingat, untuk melakukan aksi **ubah** atau **hapus** maka alurnya adalah memasukkan nomor induk dosen pada pencarian terlebih dahulu baru memilih aksi ubah atau hapus. Berikut adalah tampilan program yang sudah selesai:

The screenshot shows a Java Swing application window titled "Dosen". The window has a blue sidebar on the left with a "Dosen" button. The main area is light blue and contains a search bar, "Tambah", "Edit", and "Delete" buttons, and four input fields labeled "Nomor Induk Dosen", "Nama Dosen", "Email Dosen", and "No Handphone". At the bottom, there is a "Simpan" button and a list of two records: "INF-1 | Henri | henri@gmail.com | 08794561230" and "INF-2 | Krisna | krisna@gmail.com | 081378945262".

Dosen

INF-1

Cari

TambahEditDelete

Nomor Induk Dosen

INF-1

Nama Dosen

Henri

Email Dosen

henri@gmail.com

No Handphone

08794561230

SimpanBatal

INF-1 | Henri | henri@gmail.com | 08794561230
INF-2 | Krisna | krisna@gmail.com | 081378945262

Dosen

INF-1

Cari

TambahEditDelete

Nomor Induk Dosen

INF-1

Nama Dosen

Henri

Email Dosen

henri@gmail.com

No Handphone

08794561230

SimpanBatal

INF-1 | Henri | henri@gmail.com | 08794561230
INF-2 | Krisna | krisna@gmail.com | 081378945262

konfirmasi

?

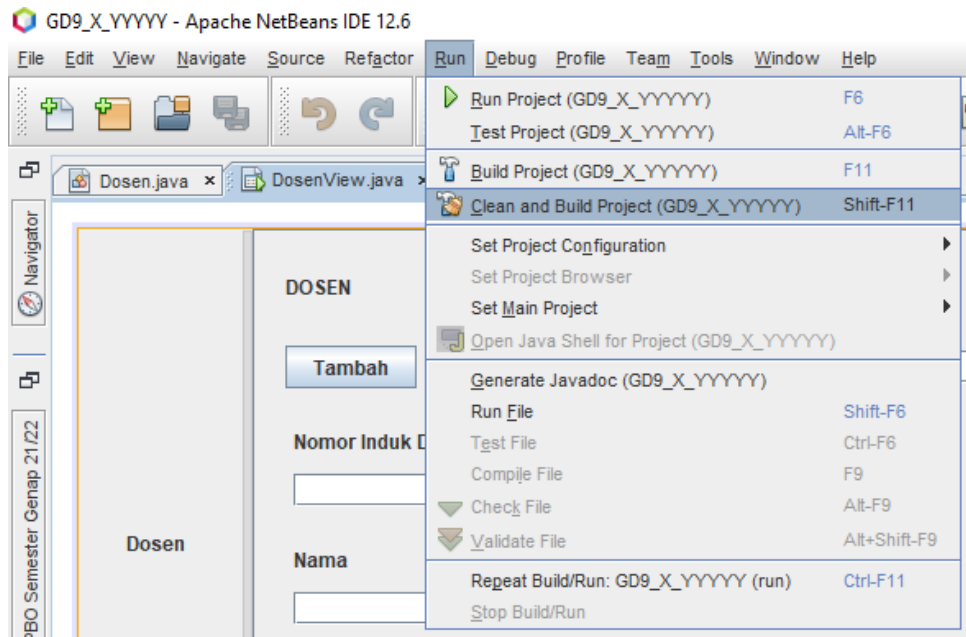
Apakah yakin ingin menghapus data?

YesNo

OPSIONAL

Saat ini, aplikasi yang telah kalian buat hanya bisa dijalankan ketika dibuka melalui IDE. Untuk membuat aplikasi yang bisa dibuka di desktop kalian, maka kalian perlu mengekspor program kalian menjadi .jar file yang merupakan executable file. Langkah-langkahnya adalah sebagai berikut:

1. Pastikan proyek kalian sudah bersih dari bug atau error.
2. Jika sudah siap ekspor, maka klik Run pada toolbar. Lalu klik “Clean and Build Project”



3. Jika proses build sudah selesai, maka bukalah folder proyek kalian di File Explorer
4. Cari folder bernama dist

Name	Type	Size
build	File folder	
dist	File folder	
nbproject	File folder	
src	File folder	
test	File folder	
build.xml	XML Document	4 KB
manifest.mf	MF File	1 KB

5. Klik 2x pada file .jar dan selamat kalian telah berhasil menjalankan aplikasi kalian di luar IDE.

Name	Type	Size
lib	File folder	
GD9_X_YYYYY.jar	JAR File	38 KB
README.TXT	Text Document	2 KB

REFERENSI UNTUK UGD

1. Pengaplikasian Radio Button dan Radio Button Group

<https://www.youtube.com/watch?v=QpDrXhzlEY0>