

PEMROGRAMAN BERORIENTASI OBJEK  
SEMESTER GENAP 2022/2023



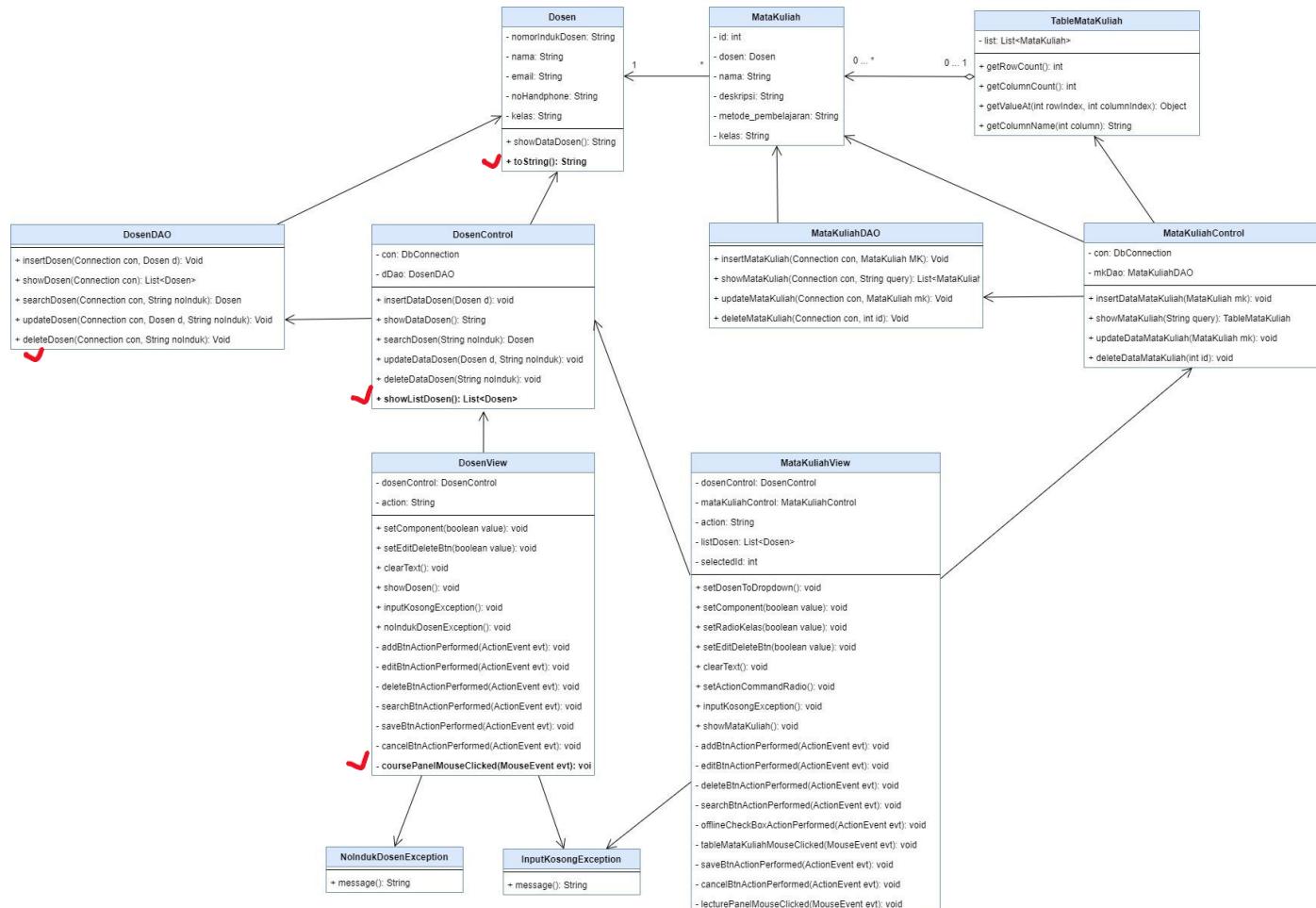
MODUL OBJECT PERSISTENCE  
BAGIAN 2

ASISTEN PBO 2022/2023  
PROGRAM STUDI INFORMATIKA FAKULTAS TEKNOLOGI INDUSTRI  
UNIVERSITAS ATMA JAYA YOGYAKARTA

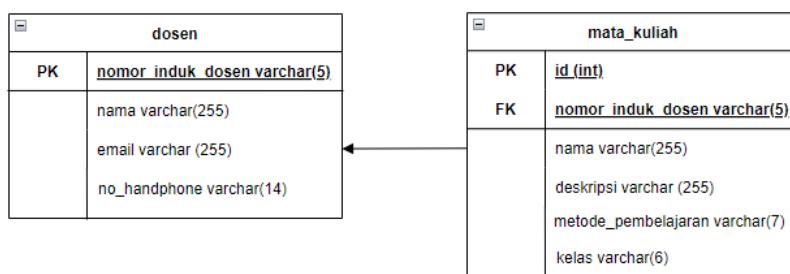
# Requirement

1. Netbeans 12.6
2. XAMPP
3. Project Modul 1 Object Persistence PBO 2022

# Guided



Gambar Class Diagram



Gambar Tabel Basis Data

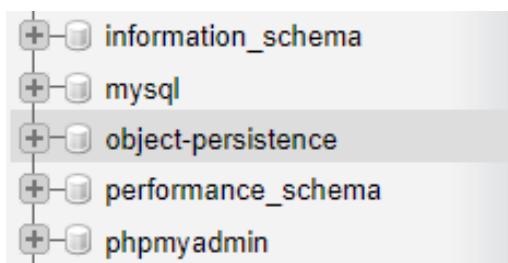
Pada modul kali ini, anda akan melanjutkan dari modul sebelumnya dengan menambahkan satu kelas lagi yaitu mata kuliah. Data mata kuliah yang ingin dikelola adalah nama, deskripsi, dosen pengampu, metode pembelajaran dan ruang kelas. Dari data-data tersebut, buatlah sistem yang dapat memenuhi fitur-fitur seperti berikut:

- Menambahkan data mata kuliah baru ke dalam sistem
- Menampilkan seluruh data-data mata kuliah yang ada di dalam sistem
- Mencari data mata kuliah yang ada di dalam sistem.
- Mengubah data mata kuliah.
- Menghapus data mata kuliah.
- Dapat memilih dosen yang telah terdaftar di dalam sistem melalui dropdown.
- Ketika metode pembelajaran dipilih **Online dan Offline**, maka metode pembelajaran yang disimpan adalah **Hybrid**.
- Ketika metode pembelajaran **Offline** dipilih, maka pengguna dapat memilih ruang kelas. Apabila metode pembelajaran **Online** maka ruang kelas tidak dapat dipilih.

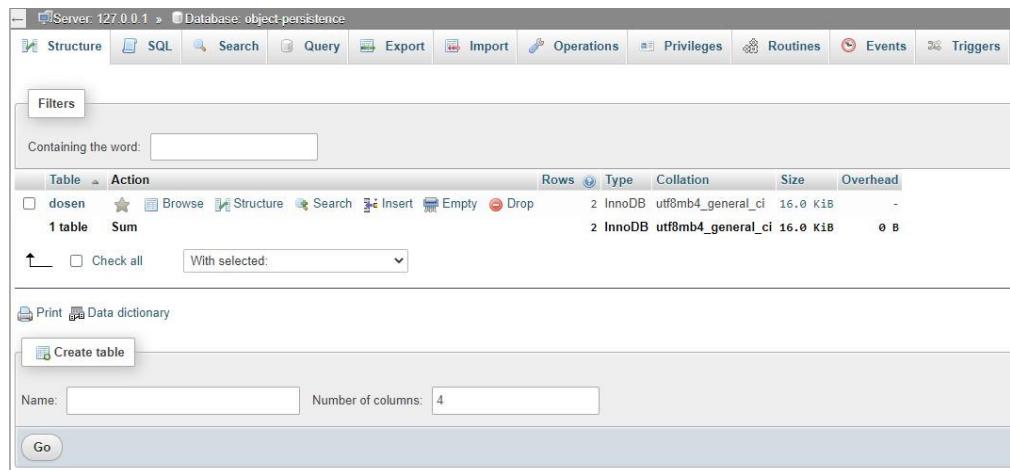
Seluruh data akan terhubung dengan basis data MySQL yang sudah dibuat pada modul sebelumnya. Pihak universitas juga meminta agar dipastikan setiap inputan tidak boleh kosong agar data dapat digunakan dengan baik. Berikut adalah langkah-langkahnya:

1. Membuat tabel mata kuliah

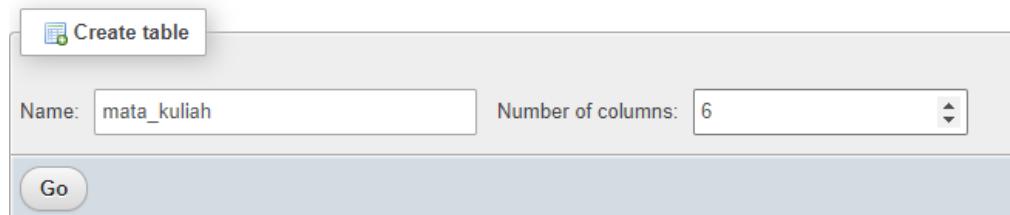
- Buka XAMPP control panel dan hidupkan service Apache dan MySQL dengan menekan tombol “Start” untuk masing-masing module Apache maupun MySQL.
- Klik tombol “Admin” pada module MySQL untuk membuka phpMyAdmin.
- Pilih database object-persistence yang sudah dibuat pada modul sebelumnya.



- Lalu akan muncul tampilan seperti berikut ini.



- Pada bagian “Create table” masukkan nama tabel mata\_kuliah dan jumlah kolom 6 lalu klik “Go”.



- Lalu masukkan nama kolom beserta tipe sesuai pada diagram yang sudah diberikan sebelumnya. Setelah itu untuk mengatur id menjadi auto increment, ubah terlebih dahulu nilai pada kolom attributes menjadi primary dan klik pada checkbox A\_I (Auto Increment). Lalu klik tombol “Save” pada bagian bawah maka tabel akan terbentuk.

Name	Type	Length/Values	Default	Collation	Attributes	Null	Index	A_I	Co
id	INT		None			PRIMARY		<input checked="" type="checkbox"/>	
nomor_ikund_dosen	VARCHAR	5	None					<input type="checkbox"/>	
nama	VARCHAR	255	None					<input type="checkbox"/>	
deskripsi	VARCHAR	255	None					<input type="checkbox"/>	
metode_pembelajaran	VARCHAR	7	None					<input type="checkbox"/>	
kelas	VARCHAR	6	None					<input type="checkbox"/>	

- Setelah tabel terbentuk seperti pada gambar berikut, klik “Relation view” untuk mengatur relasi dengan tabel dosen.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
1	id	int(11)			No	None		AUTO_INCREMENT	Change  Drop  More
2	nomor_induk_dosen	varchar(5)	utf8mb4_general_ci		No	None			Change  Drop  More
3	nama	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
4	deskripsi	varchar(255)	utf8mb4_general_ci		No	None			Change  Drop  More
5	metode_pembelajaran	varchar(7)	utf8mb4_general_ci		No	None			Change  Drop  More
6	kelas	varchar(6)	utf8mb4_general_ci		No	None			Change  Drop  More

- Setelah itu klik “Internal relationships” sehingga muncul seperti gambar berikut.

The screenshot shows the MySQL Workbench interface for the 'mata\_kuliah' table. The 'Internal relationships' tab is active. A grid lists columns and their corresponding internal relations:

Column	Internal relation
id	object-persistent
nomor_induk_dosen	object-persistent
nama	object-persistent
deskripsi	object-persistent
metode_pembelajaran	object-persistent
kelas	object-persistent

Below the grid, there is a dropdown for 'Choose column to display' set to 'nomor\_induk\_dosen'. At the bottom are 'Preview SQL' and 'Save' buttons.

- Pada kolom nomor\_induk\_dosen, isikan relasi dengan tabel dosen dan kolom nomor\_induk\_dosen lalu klik “Save”.

Internal relationships

Column	Internal relation		
id	object-persistent	▼	▼
nomor_induk_dosen	object-persistent	dosen	nomor_induk_dc
nama	object-persistent	▼	▼
deskripsi	object-persistent	▼	▼
metode_pembelajaran	object-persistent	▼	▼
kelas	object-persistent	▼	▼

Choose column to display: nomor\_induk\_dosen ▼

Preview SQL Save

## 2. Membuat entity class

Buka project dari modul sebelumnya dan buatlah kelas baru pada package model yaitu MataKuliah. Berikut codenya:

```

11  public class MataKuliah {
12      private int id;
13      private String nama;
14      private String deskripsi;
15      private String metode_pembelajaran;
16      private String kelas;
17      private Dosen dosen;
18
19      public MataKuliah(int id, String nama, String deskripsi, String metode_pembelajaran, String kelas, Dosen dosen) {
20          this.id = id;
21          this.nama = nama;
22          this.deskripsi = deskripsi;
23          this.metode_pembelajaran = metode_pembelajaran;
24          this.kelas = kelas;
25          this.dosen = dosen;
26      }
27
28      public MataKuliah(String nama, String deskripsi, String metode_pembelajaran, String kelas, Dosen dosen) {
29          this.nama = nama;
30          this.deskripsi = deskripsi;
31          this.metode_pembelajaran = metode_pembelajaran;
32          this.kelas = kelas;
33          this.dosen = dosen;
34      }
35
36      public int getId() {
37          return id;
38      }
39

```

```

40 |     public void setId(int id) {
41 |         this.id = id;
42 |     }
43 |
44 |     public String getNama() {
45 |         return nama;
46 |     }
47 |
48 |     public void setNama(String nama) {
49 |         this.nama = nama;
50 |     }
51 |
52 |     public String getDeskripsi() {
53 |         return deskripsi;
54 |     }
55 |
56 |     public void setDeskripsi(String deskripsi) {
57 |         this.deskripsi = deskripsi;
58 |     }
59 |
60 |     public String getMetode_pembelajaran() {
61 |         return metode_pembelajaran;
62 |     }
63 |
64 |     public void setMetode_pembelajaran(String metode_pembelajaran) {
65 |         this.metode_pembelajaran = metode_pembelajaran;
66 |     }
67 |
68 |     public String getKelas() {
69 |         return kelas;
70 |     }
71 |
72 |     public void setKelas(String kelas) {
73 |         this.kelas = kelas;
74 |     }
75 |
76 |     public Dosen getDosen() {
77 |         return dosen;
78 |     }
79 |
80 |     public void setDosen(Dosen dosen) {
81 |         this.dosen = dosen;
82 |     }
83 |
84 |

```

### 3. Membuat dao class

Langkah selanjutnya adalah membuat class baru pada package dao yaitu MataKuliahDAO.

Berikut adalah langkah-langkahnya:

- Import beberapa library yang akan digunakan.

```

5     package dao;
6
7     import connection.DbConnection;
8     import java.sql.Connection;
9     import java.sql.ResultSet;
10    import java.sql.Statement;
11    import java.util.ArrayList;
12    import java.util.List;
13    import model.Dosen;
14    import model.MataKuliah;

```

- Kemudian buatlah variable untuk menampung objek new DbConnection serta Connection agar dapat terkoneksi ke database.

```

20  public class MataKuliahDAO {
21      ...
22      ...
23      private DbConnection dbcon = new DbConnection();
24      private Connection con;

```

- Langkah selanjutnya adalah membuat method untuk menambahkan data mata kuliah ke dalam tabel mata\_kuliah yang ada pada database.

```

25  // method untuk menyimpan data mata kuliah
26  public void insertMataKuliah(MataKuliah mk) {
27
28      ...
29
30      String sql = "INSERT INTO mata_kuliah(nomor_induk_dosen, nama, deskripsi, metode_pembelajaran, kelas) VALUES ('"
31          + mk.getDosen().getNomorIndukDosen() + "','" + mk.getNama() + "','" + mk.getDeskripsi() + "','" +
32          + mk.getMetode_pembelajaran() + "','" +
33          + mk.getKelas() + "')";
34
35      System.out.println("Adding Mata Kuliah...");
36
37      try {
38          Statement statement = con.createStatement();
39          int result = statement.executeUpdate(sql);
40          System.out.println("Added " + result + " Mata Kuliah");
41          statement.close();
42      } catch (Exception e) {
43          System.out.println("Error adding Mata Kuliah...");
44          System.out.println(e);
45      }
46
47      dbcon.closeConnection();
48  }

```

- Lanjutkan dengan membuat method untuk menampilkan data berdasarkan kata kunci pencarian.

```

50  // method untuk mengambil data mata kuliah
51  public List<MataKuliah> showMataKuliah(String query) {
52
53      ...
54
55      String sql = "SELECT mk.*, d.* FROM mata_kuliah AS mk JOIN dosen AS d ON d.nomor_induk_dosen = mk.nomor_induk_dosen WHERE (mk.nama LIKE "
56          + "%" + query + "%"
57          + "OR mk.deskripsi LIKE '%" + query + "%'"
58          + "OR mk.metode_pembelajaran LIKE '%" + query + "%'"
59          + "OR mk.kelas LIKE '%" + query + "%'"
60          + "OR d.nama LIKE '%" + query + "%')";
61      System.out.println("Mengambil data Mata Kuliah...");
62      List<MataKuliah> list = new ArrayList();
63
64      try {
65          Statement statement = con.createStatement();
66          ResultSet rs = statement.executeQuery(sql);
67
68          if (rs != null) {
69              while (rs.next()) {
70                  Dosen d = new Dosen(rs.getString("d.nomor_induk_dosen"),
71                      rs.getString("d.nama"), rs.getString("d.email"),
72                      rs.getString("d.no_handphone"));
73
74                  MataKuliah mk = new MataKuliah(Integer.parseInt(rs.getString("mk.id")),
75                      rs.getString("mk.nama"), rs.getString("mk.deskripsi"),
76                      rs.getString("mk.metode_pembelajaran"), rs.getString("mk.kelas"),
77                      d);
78                  list.add(mk);
79              }
80          }
81          rs.close();
82          statement.close();
83      } catch (Exception e) {
84          System.out.println("Error reading database...");
85          System.out.println(e);
86      }
87
88      dbcon.closeConnection();
89
90      return list;
91  }

```

- Berikutnya buatlah method untuk mengubah data mata kuliah. Data pada tabel mata\_kuliah masing-masing nilainya akan diubah berdasarkan inputan pengguna.

```

93 // method untuk mengubah data mata kuliah
94 public void updateMataKuliah(MataKuliah mk) {
95
96     con = dbcon.makeConnection();
97
98     String sql = "UPDATE mata_kuliah SET nomor_induk_dosen = '" + mk.getDosen().getNomorIndukDosen()
99             + "', nama = '" + mk.getNama()
100            + "', deskripsi = '" + mk.getDeskripsi()
101            + "', metode_pembelajaran = '" + mk.getMetode_pembelajaran()
102            + "', kelas = '" + mk.getKelas()
103            + "' WHERE id = '" + mk.getId() + "'";
104
105     System.out.println("Editing Mata Kuliah...");
106
107     try {
108         Statement statement = con.createStatement();
109         int result = statement.executeUpdate(sql);
110         System.out.println("Edited " + result + " Mata Kuliah " + mk.getId());
111         statement.close();
112     } catch (Exception e) {
113         System.out.println("Error editing mata kuliah...");
114         System.out.println(e);
115     }
116
117     dbcon.closeConnection();
}

```

- Method terakhir yang perlu dibuat adalah method untuk menghapus data mata kuliah berdasarkan id.

```

119 // method untuk menghapus data mata kuliah
120 public void deleteMataKuliah(int id) {
121
122     con = dbcon.makeConnection();
123
124     String sql = "DELETE FROM mata_kuliah WHERE id = " + id + "";
125     System.out.println("Deleting Mata Kuliah...");
126
127     try {
128         Statement statement = con.createStatement();
129         int result = statement.executeUpdate(sql);
130         System.out.println("Delete " + result + " Mata Kuliah " + id);
131         statement.close();
132     } catch (Exception e) {
133         System.out.println("Error deleting mata kuliah...");
134         System.out.println(e);
135     }
136
137     dbcon.closeConnection();
138 }
139
140

```

#### 4. Membuat table class

Pada modul ini, kita akan menggunakan table untuk menampilkan dan berinteraksi dengan data mata kuliah. Buatlah sebuah package baru bernama table dan buat class baru bernama TableMataKuliah. Jangan lupa untuk menambahkan extends AbstractTableModel yang merupakan base dari table yang akan digunakan di view nanti. Dengan kelas yang dibuat ini, maka akan mempermudah kita dalam menampilkan data atau mengambil data dari table. Lanjutkan dengan menambahkan code seperti berikut:



```
5     package table;
6
7     import java.util.List;
8     import javax.swing.table.AbstractTableModel;
9     import model.MataKuliah;
10
11    /**
12     *
13     *  asistenPBO2022
14     */
15    public class TableMataKuliah extends AbstractTableModel{
16
17        private List<MataKuliah> list;
18
19        public TableMataKuliah(List<MataKuliah> list){
20            this.list = list;
21        }
22
23        public int getRowCount() {
24            return list.size();
25        }
26
27        public int getColumnCount() {
28            return 5;
29        }
30
31        public Object getValueAt(int rowIndex, int columnIndex) {
32            switch (columnIndex) {
33                case 0:
34                    return list.get(rowIndex).getNama();
35                case 1:
36                    return list.get(rowIndex).getDeskripsi();
37                case 2:
38                    return list.get(rowIndex).getDosen().getNama();
39                case 3:
40                    return list.get(rowIndex).getMetode_pembelajaran();
41                case 4:
42                    return list.get(rowIndex).getKelas();
43                case 5:
44                    return list.get(rowIndex).getId();
45                case 6:
46                    return list.get(rowIndex).getDosen().getNomorIndukDosen();
47                default:
48                    return null;
49            }
50        }
51
52        public String getColumnName(int column) {
53            switch (column) {
54                case 0:
55                    return "Nama Mata Kuliah";
56                case 1:
57                    return "Deskripsi";
58                case 2:
59                    return "Dosen";
60                case 3:
61                    return "Metode Pembelajaran";
62                case 4:
63                    return "Kelas";
64                default:
65                    return null;
66            }
67        }
68    }
69}
```

Pada class ini, dibuat pertama kali yaitu method untuk mendapatkan jumlah baris dan kolom dari data yang akan ditampilkan. Untuk mendapatkan jumlah baris, bisa langsung menggunakan `list.size()` sedangkan untuk jumlah kolom bisa langsung dituliskan angka 5 karena akan menampilkan 5 kolom pada tabel. Selanjutnya membuat method `getValueAt` yang fungsinya untuk mendapatkan data dari baris yang dipilih oleh pengguna. Langkah terakhir adalah membuat method `getColumnName` untuk menentukan nama kolom yang akan ditampilkan nantinya pada tampilan.

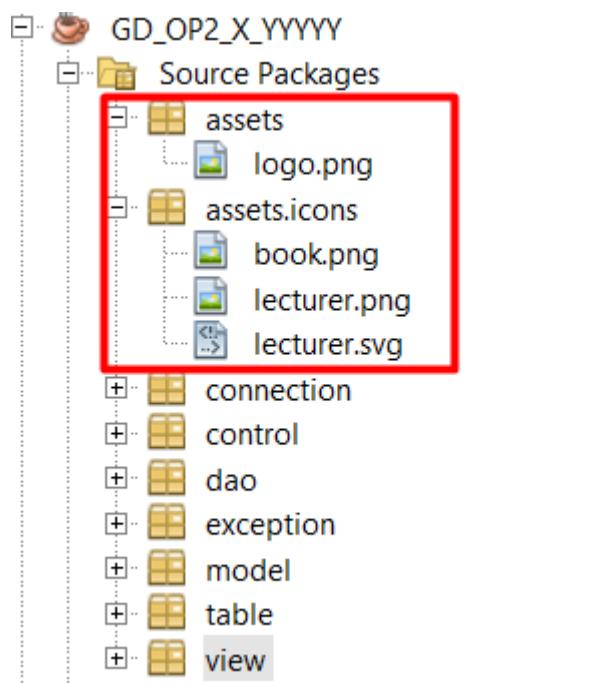
##### 5. Membuat control class

Selanjutnya buatlah class baru pada package control bernama MataKuliahControl dengan code seperti berikut:

```
5      package control;
6
7      import dao.MataKuliahDAO;
8      import java.util.List;
9      import model.MataKuliah;
10     import table.TableMataKuliah;
11
12     /**
13      *
14      * asistenPBO2022
15      */
16
17     public class MataKuliahControl {
18         private MataKuliahDAO mkDao = new MataKuliahDAO();
19
20         public void insertDataMataKuliah(MataKuliah mk) {
21             mkDao.insertMataKuliah(mk);
22         }
23
24         public TableMataKuliah showMataKuliah(String query)
25         {
26             List<MataKuliah> dataMataKuliah = mkDao.showMataKuliah(query);
27             TableMataKuliah tableMataKuliah = new TableMataKuliah(dataMataKuliah);
28
29             return tableMataKuliah;
30         }
31
32         public void updateDataMataKuliah(MataKuliah mk) {
33             mkDao.updateMataKuliah(mk);
34         }
35
36         public void deleteDataMataKuliah(int id) {
37             mkDao.deleteMataKuliah(id);
38         }
39     }
```

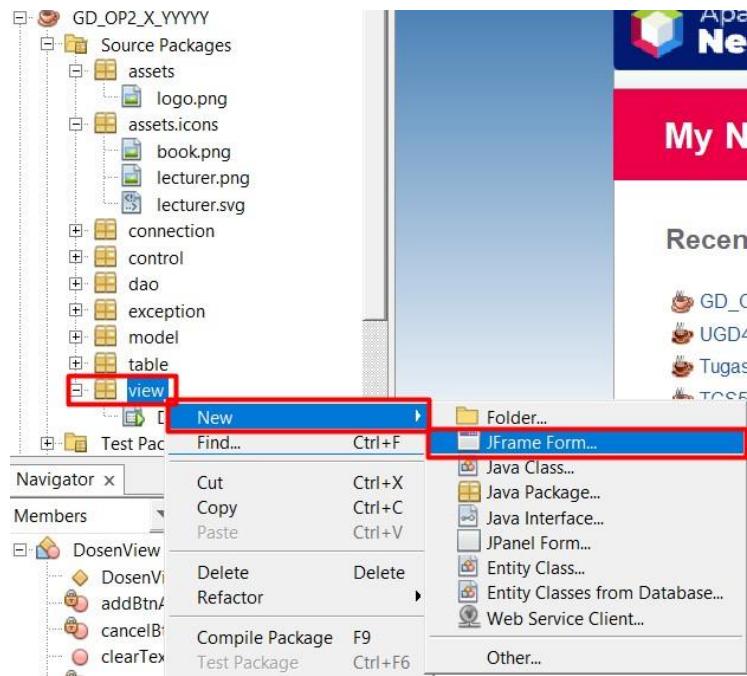
## 6. Membuat user interface class

User interface merupakan bagian presentation logic sebuah aplikasi yang digunakan sebagai antarmuka kepada user. Sebelum masuk ke pembuatan UI silahkan **download assets.zip** yang ada **di situs kuliah**, lalu ekstrak dan masukkan ke dalam folder src pada proyek netbeans kalian. Sehingga struktur folder akan menjadi seperti ini.

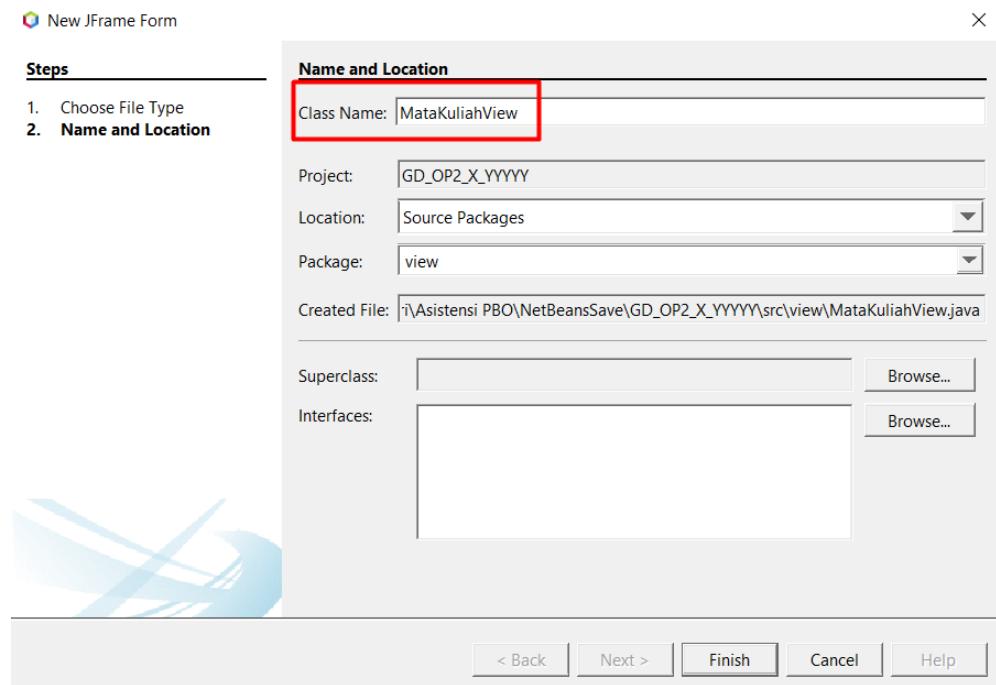


Selanjutnya, kita akan masuk ke pembuatan UI. Berikut adalah langkah-langkahnya :

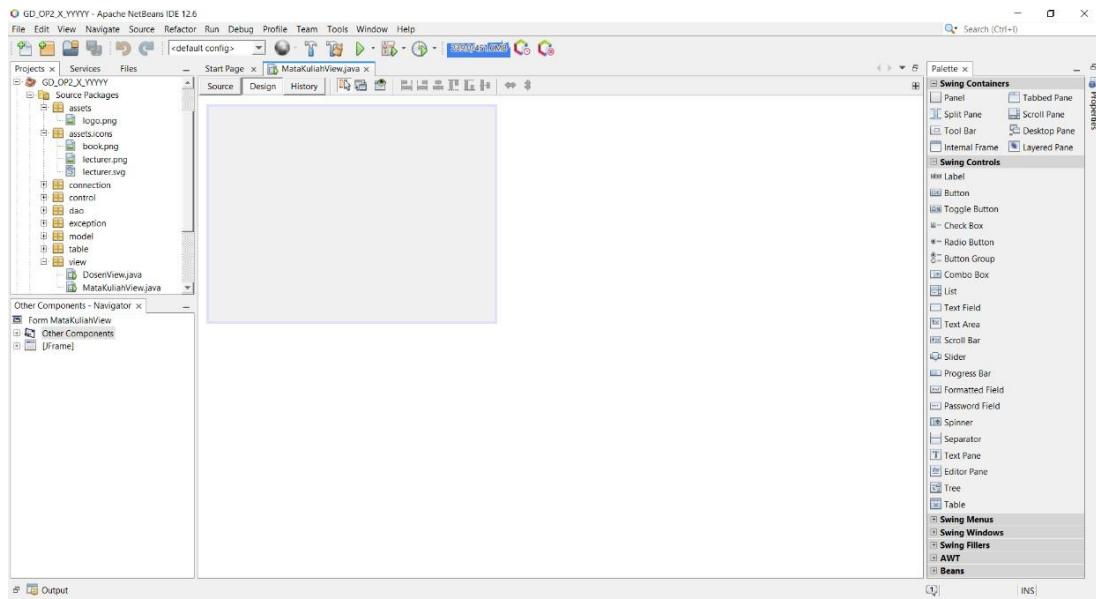
- Buatlah class baru dengan tipe JFrameForm pada package view seperti terlihat pada gambar di bawah ini



- Pada modal yang muncul berilah nama class tersebut dengan nama MataKuliahView, lalu klik Finish



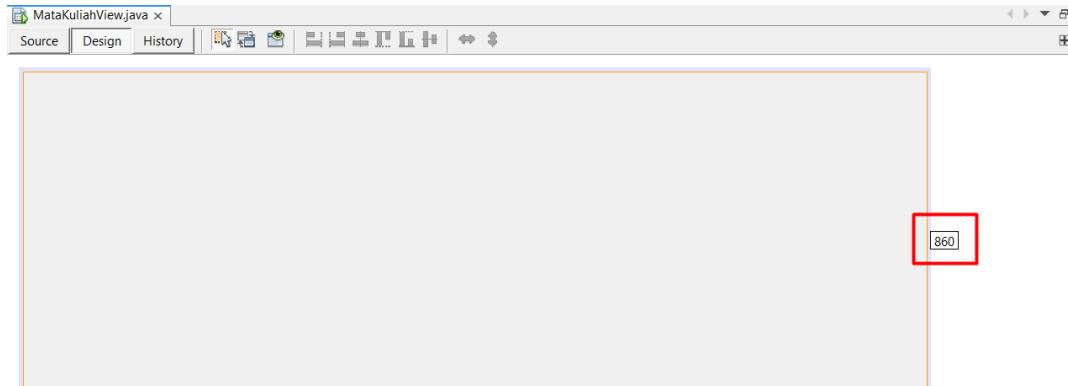
- Setelah itu maka akan muncul halaman baru seperti gambar di bawah ini

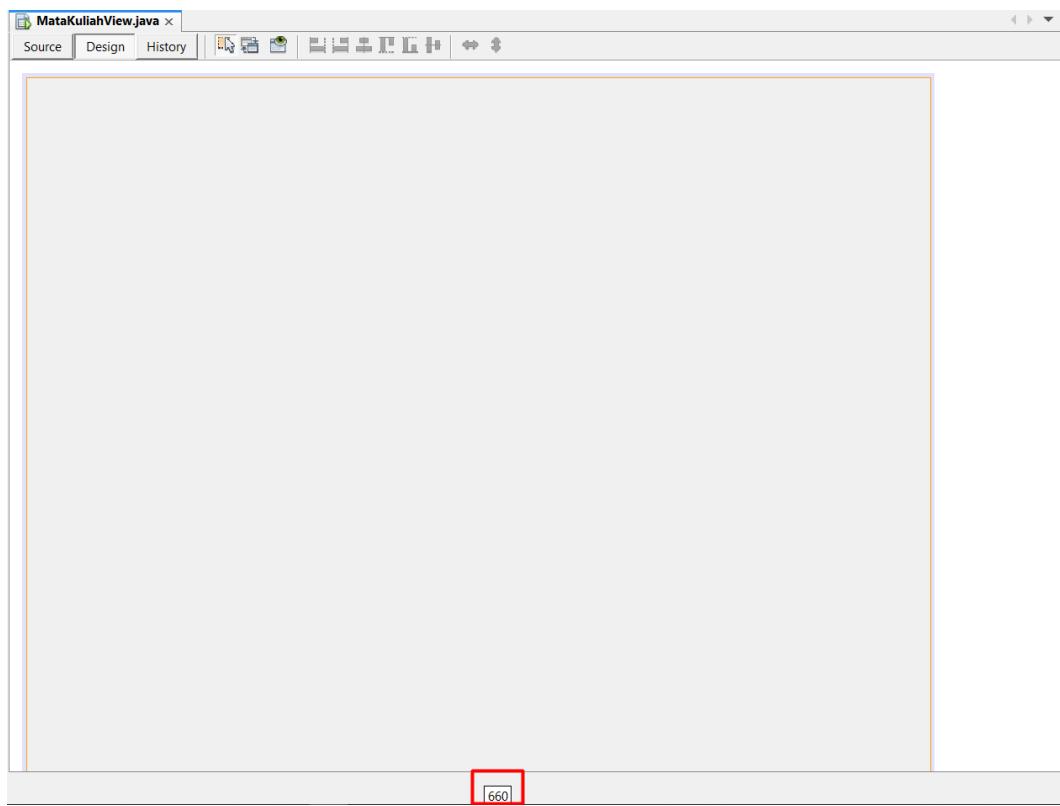


Proses pembuatan UI dibagi menjadi 2 bagian, yaitu styling untuk mendesain UI dan mengintegrasikan logic dengan UI yang telah dibuat.

### I. Styling desain komponen UI

Sebelum masuk lebih dalam silahkan atur ukuran frame menjadi 860 x 660 dengan cara menarik pada bagian sisi atas dan bawahnya, atau double click pada bordernya.

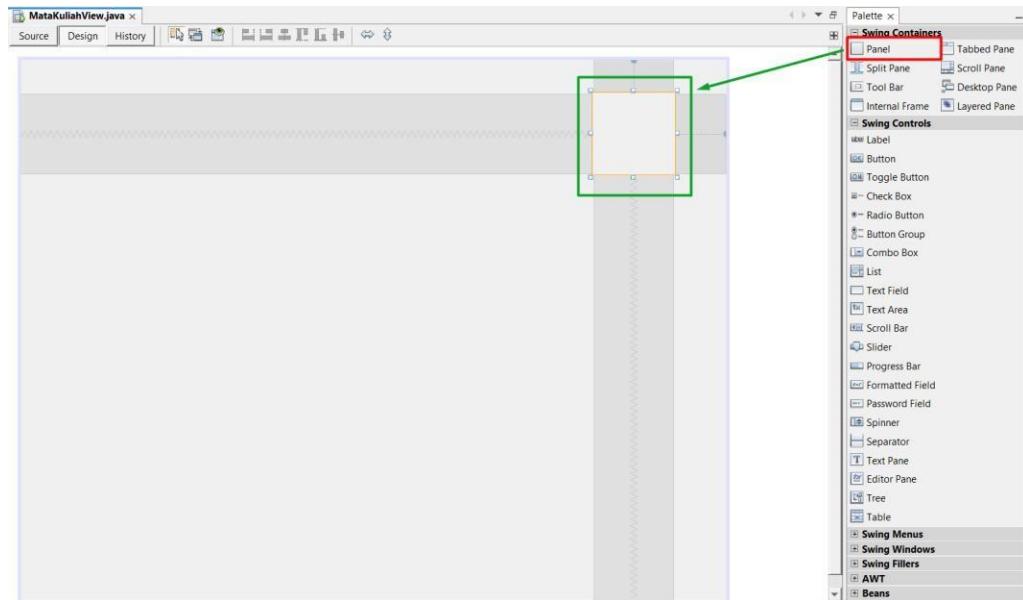




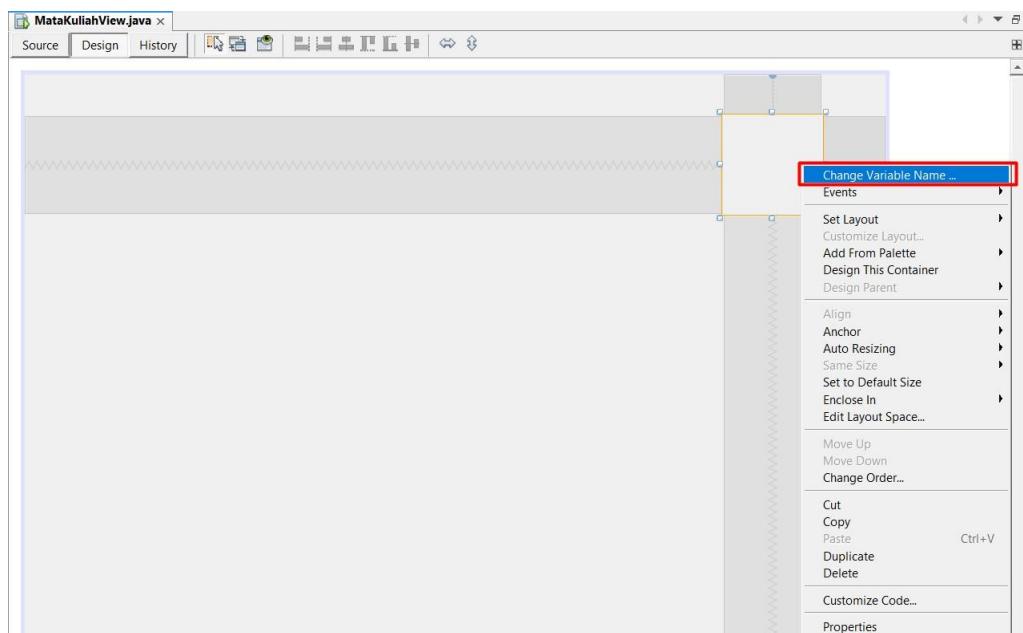
Ikuti langkah-langkah berikut ini secara teliti untuk mendesain tampilan UI.

## PEMBUATAN SIDEBAR

- a. Pada bagian pertama kita akan membuat komponen sidebar. Pada bagian Pallete klik komponen Panel (perhatikan kotak merah) dan jPanel otomatis akan masuk ke dalam jFrame kalian (perhatikan kotak hijau). Selain itu, kalian dapat melakukan *drag and drop* komponen yang ingin ditambahkan.

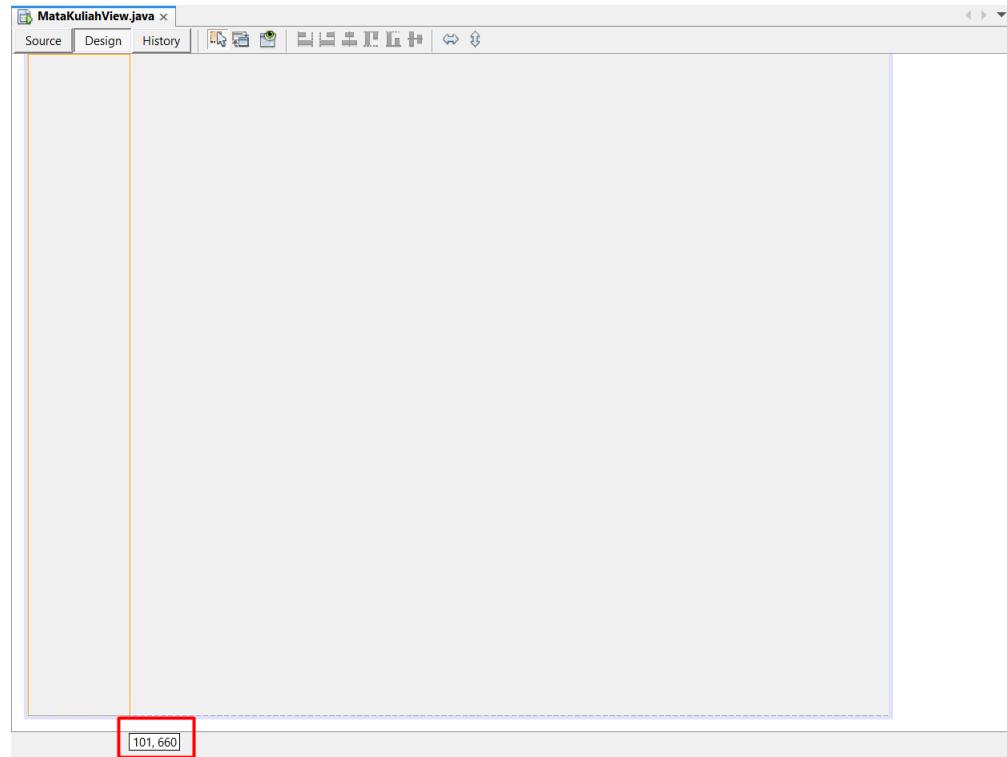


- b. Pada jPanel tersebut, silahkan ganti nama variabel dengan cara klik kanan pada jPanel pilih Change variable name...

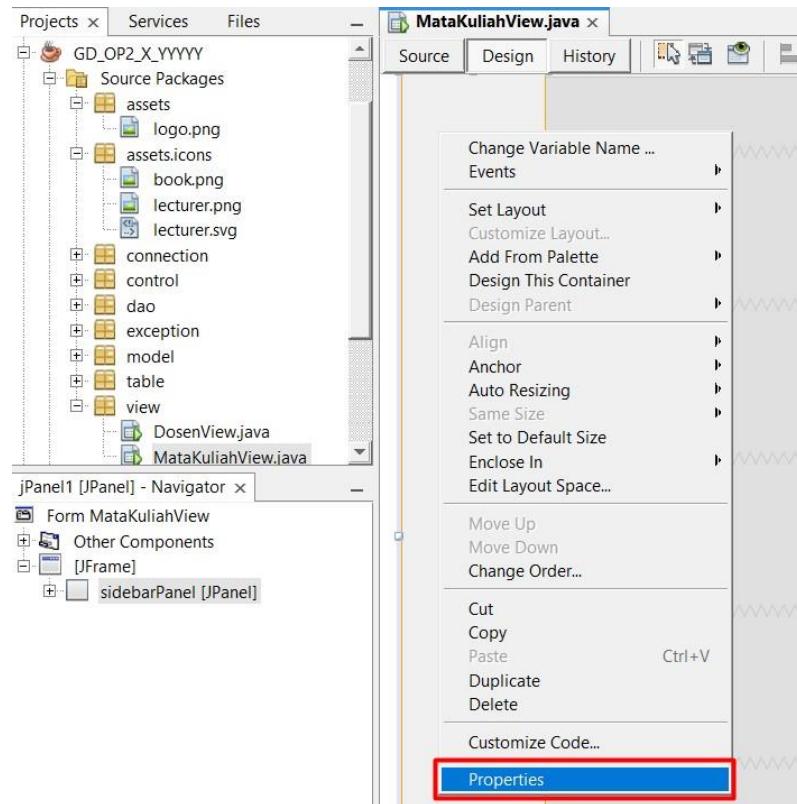


- c. Setelah itu akan muncul sebuah modal dan ganti nama variabel dengan nama sidebarPanel. Lakukan penggantian variabel setiap anda menambahkan komponen baru dengan tujuan memudahkan mengenali komponen yang kalian buat ketika membuat logic dari UI.

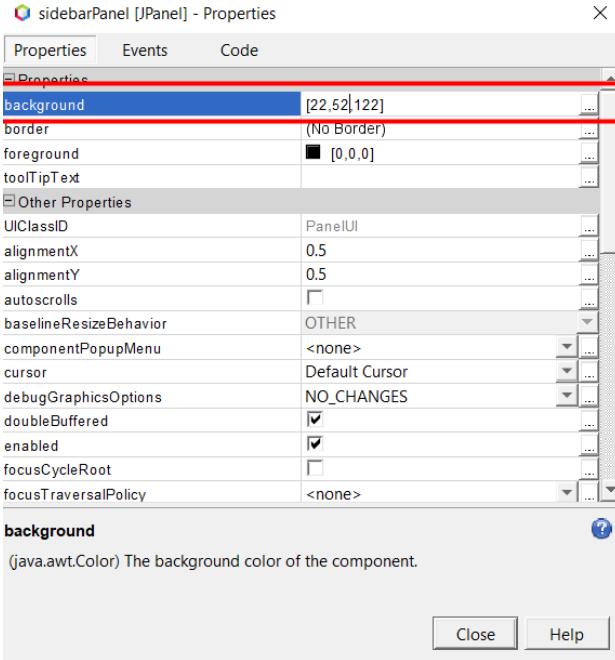
d. Setelah itu letakan sidebarPanel di sebelah kiri jFrame dan set ukuran menjadi 101 x 660.



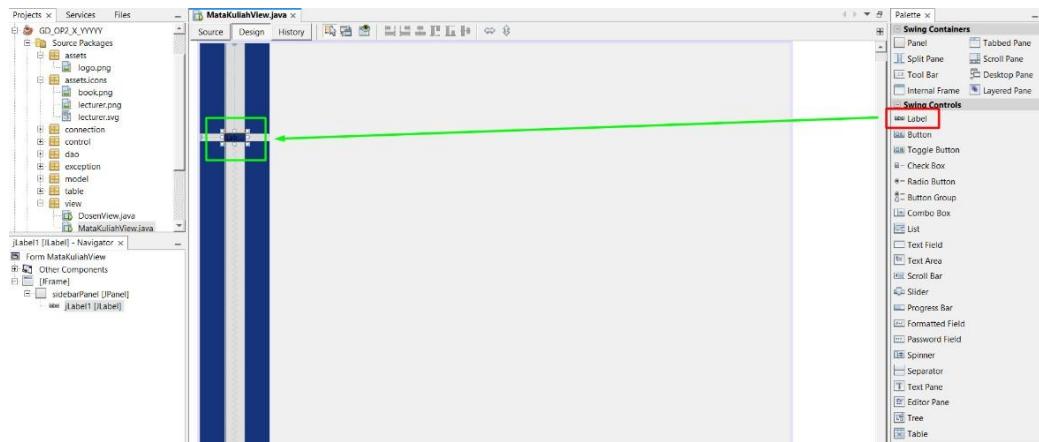
e. Ubah background dari sidebarPanel dengan cara klik kanan pada sidebarPanel di navigator lalu pilih properties atau klik kanan pada field sidebarPanel lalu pilih properties.



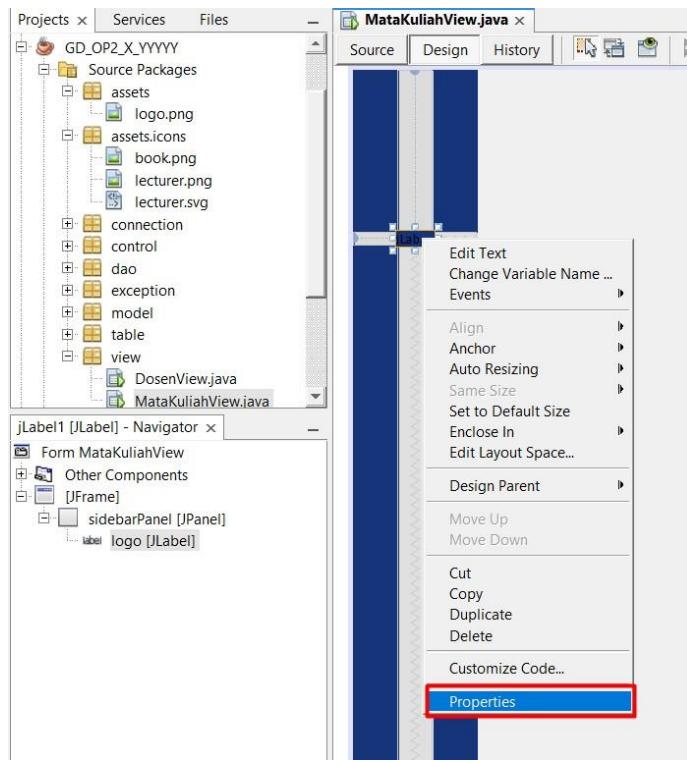
- f. Pada modal yang muncul ganti value pada field background dengan kode warna 22, 52, 122 seperti pada gambar di bawah ini, lalu klik close



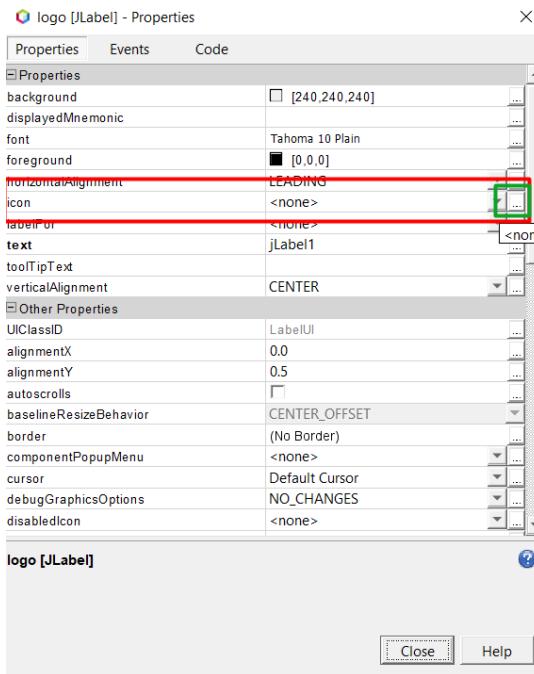
- g. Setelah itu, kita akan mengimport logo dari folder assets yang sudah dimasukkan ke dalam folder projek kalian. Pertama, *drag* komponen Label pada bagian Pallette dan *drop* di sidebarPanel. Setelah itu ubah nama variabel menjadi logo.



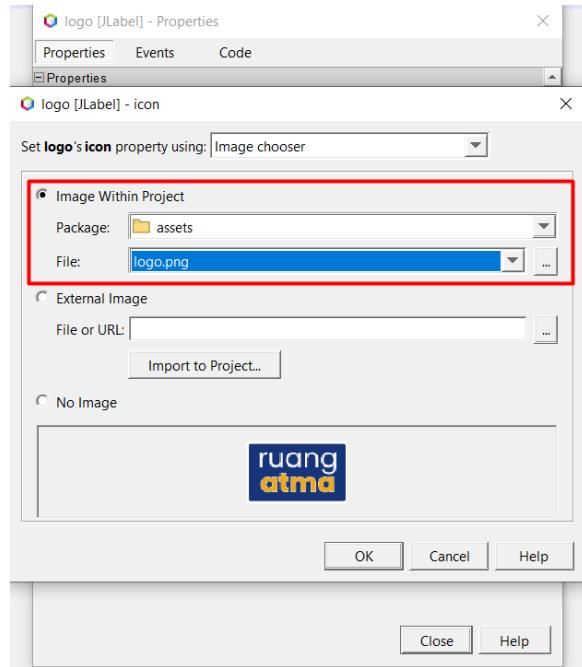
- h. Untuk mengimport gambar silahkan klik kanan pada jLabel logo dan pilih properties.



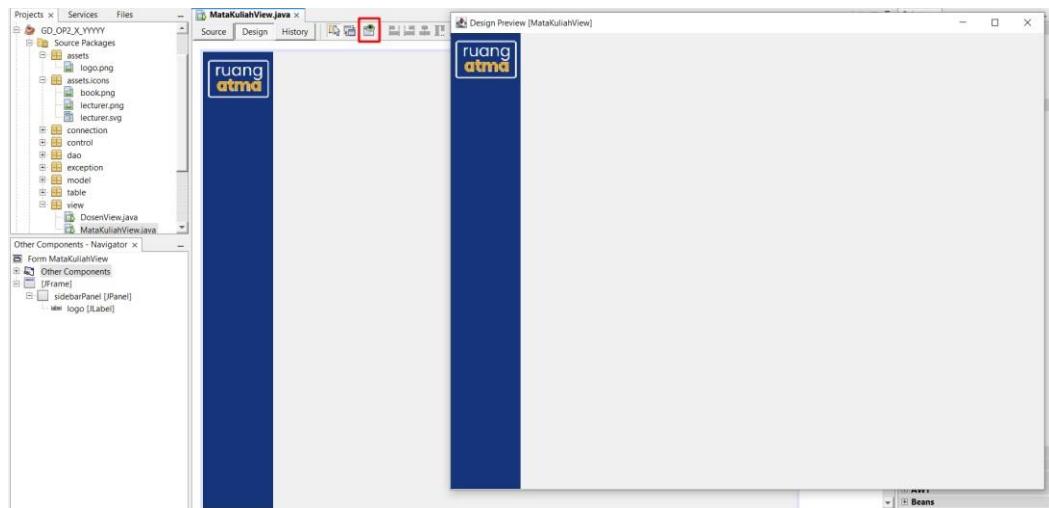
- i. Pada modal yang muncul silahkan pilih pada bagian icon (kotak merah) dan klik button yang ditandai dengan warna hijau



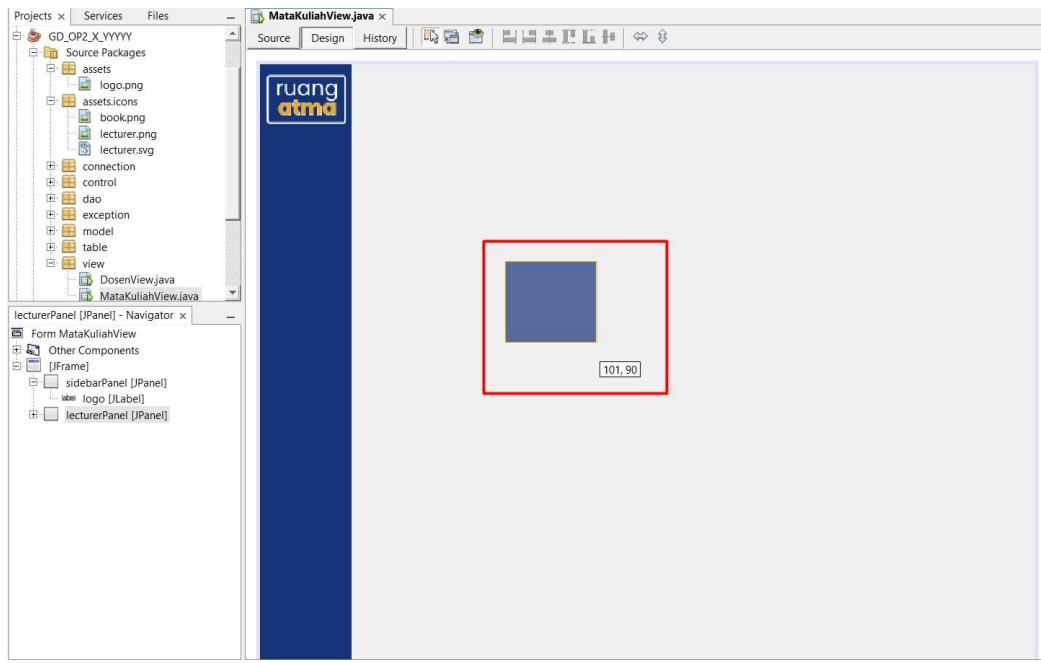
- j. Silahkan pilih gambar logo.png sesuai dengan *screenshot* di bawah ini. Lalu klik OK dan klik Close. Anda dapat menghapus tulisan yang ada di label tersebut dengan *double click* pada label tersebut dan hapus. Jika panelSidebar berubah ukuran, anda dapat mengatur ulang ukuran sesuai ketentuan di awal.



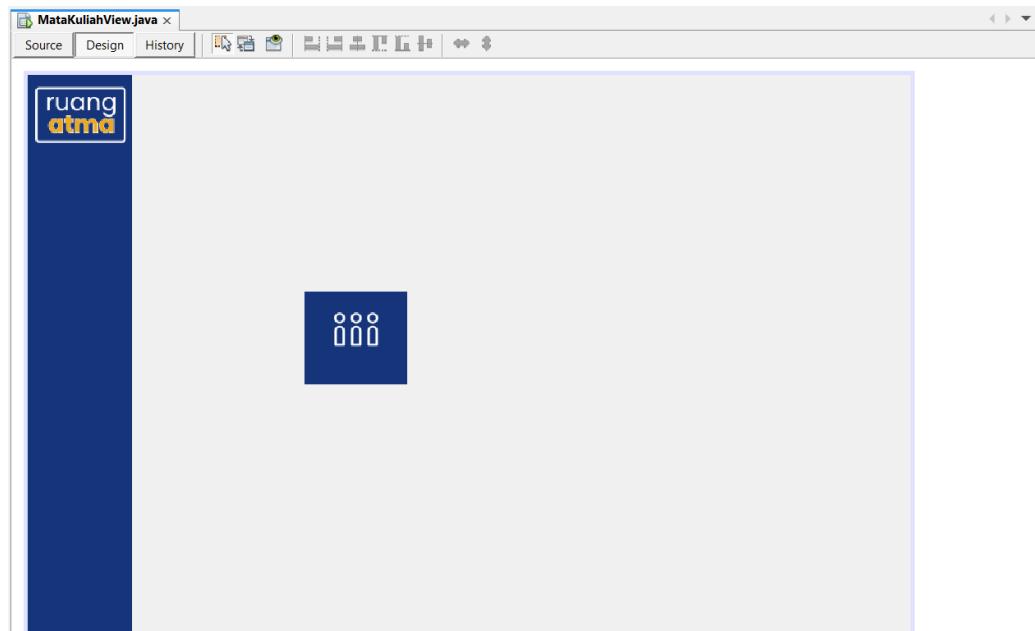
- k. Setelah itu posisikan logo ke bagian atas `sidebarPanel` seperti gambar di bawah ini dengan *drag and drop* komponen. Untuk mengecek hasil *design* tanpa harus melakukan scroll dan bisa tampil secara full screen klik kotak warna merah untuk melihat *design preview*.



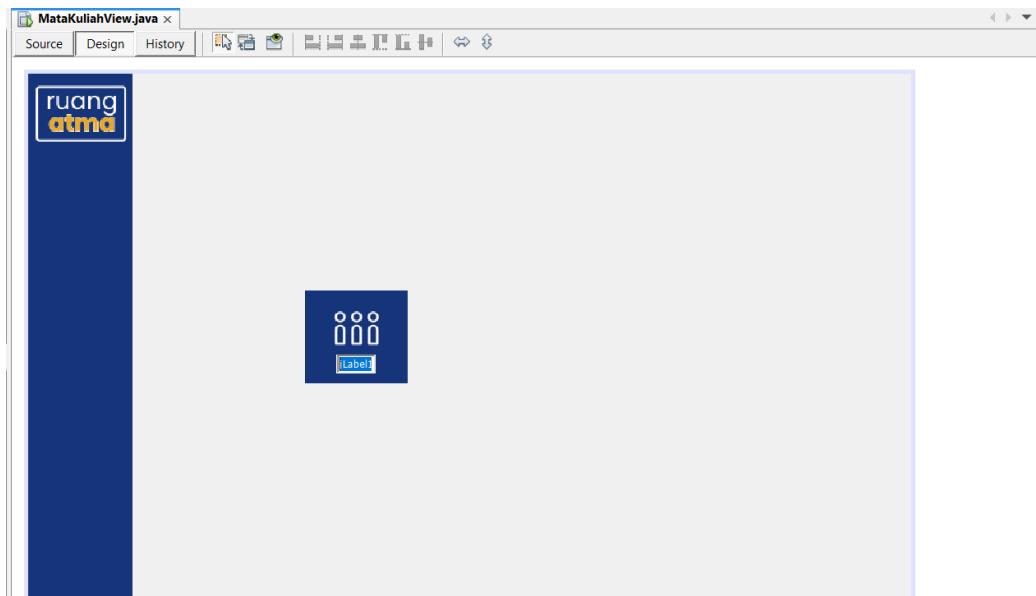
- l. Selanjutnya kita akan membuat menu yang ada pada `sidebarPanel`. Tambah komponen panel dan beri nama variabel `lecturerPanel`. Setelah itu beri background warna dengan kode `22, 52, 122` dan set ukurannya menjadi `101 x 90` hingga tertampil seperti ini.



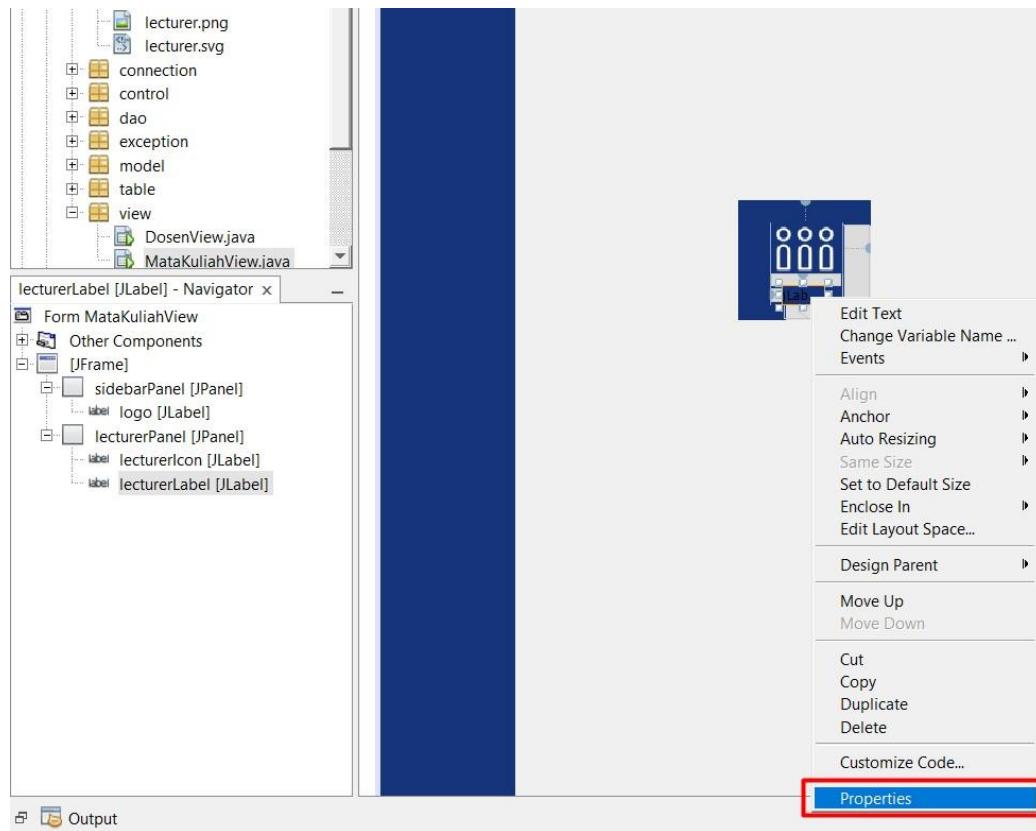
- m. Setelah itu tambahkan komponen Label ke dalam komponen lecturerPanel. Ubah variabel label tersebut menjadi lecturerIcon. Silahkan import gambar pada folder assets.icons/lecturer.png pada label lecturerIcon dan atur posisi sedemikian rupa hingga seperti gambar di bawah ini.



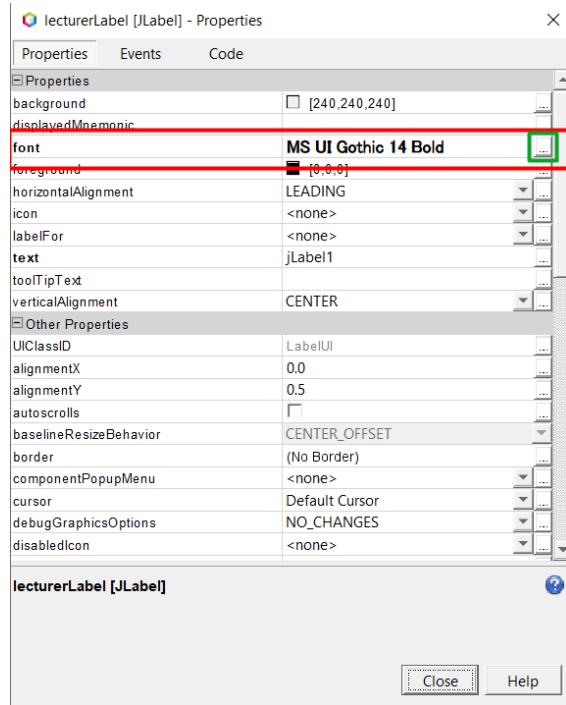
- n. Tambahkan komponen Label ke dalam komponen lecturerPanel dan ubah nama variabel label menjadi lecturerLabel. Atur panjang lecturerLabel menjadi sama dengan lecturerPanel. Pastikan saat ini anda memiliki tampilan yang sama seperti gambar di bawah ini.



- o. Selanjutnya kita akan memberikan *styling* sederhana pada tulisan tersebut dengan cara klik kanan pada komponen `lecturerLabel` dan pilih properties. Jangan lupa juga bisa klik kanan pada field `lecturerLabel` dan pilih properties.

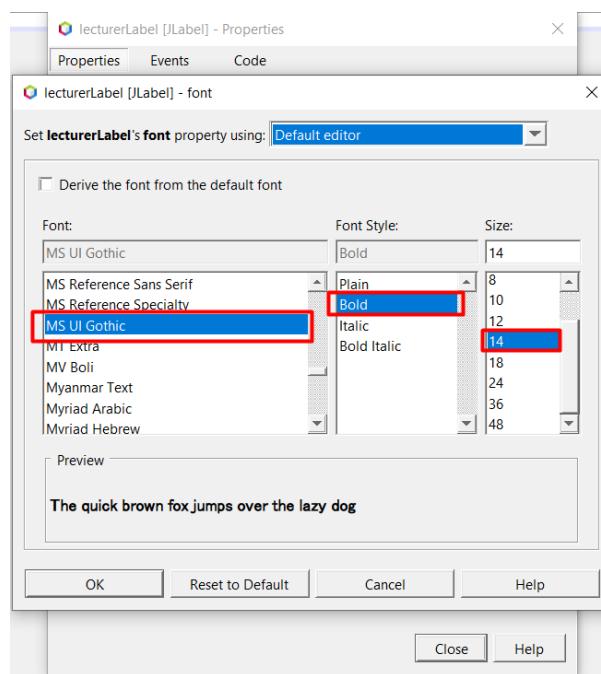


- p. Untuk mengubah font, klik button pada bagian font yang diberi kotak hijau seperti di bawah ini.

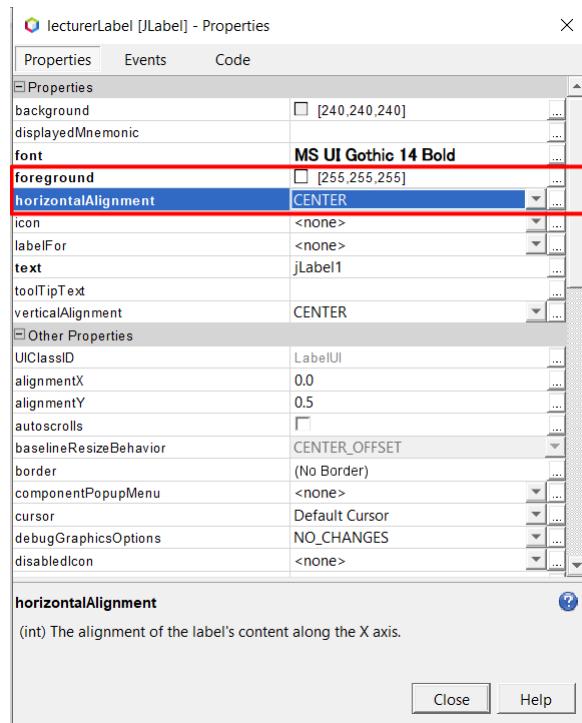


q. Silahkan atur *styling* -nya sama seperti gambar di bawah ini.

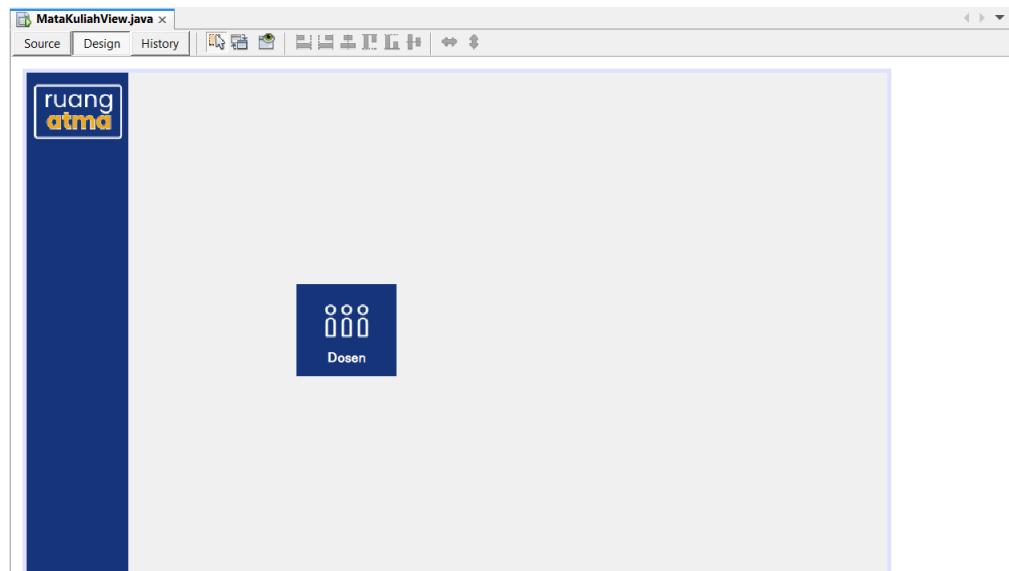
- Font : MS UI Gothic (optional)
- Font style: bold
- Font size : 14



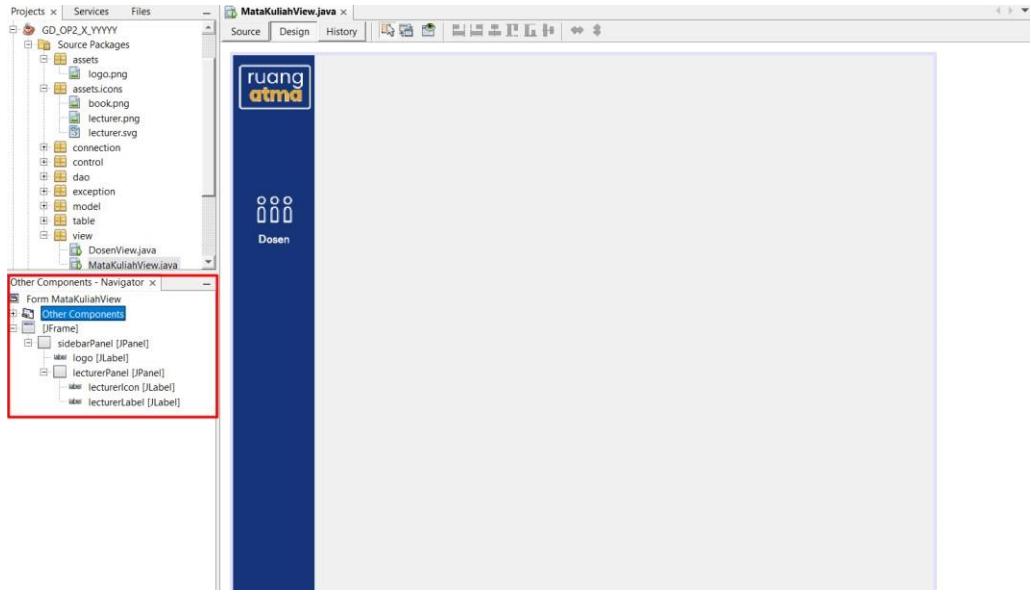
r. Untuk mengubah warna font, pada bagian foreground silahkan masukan kode warna 255,255,255. Untuk mengubah alignment tulisan menjadi tengah, silahkan pilih CENTER pada bagian horizontalAlignment.



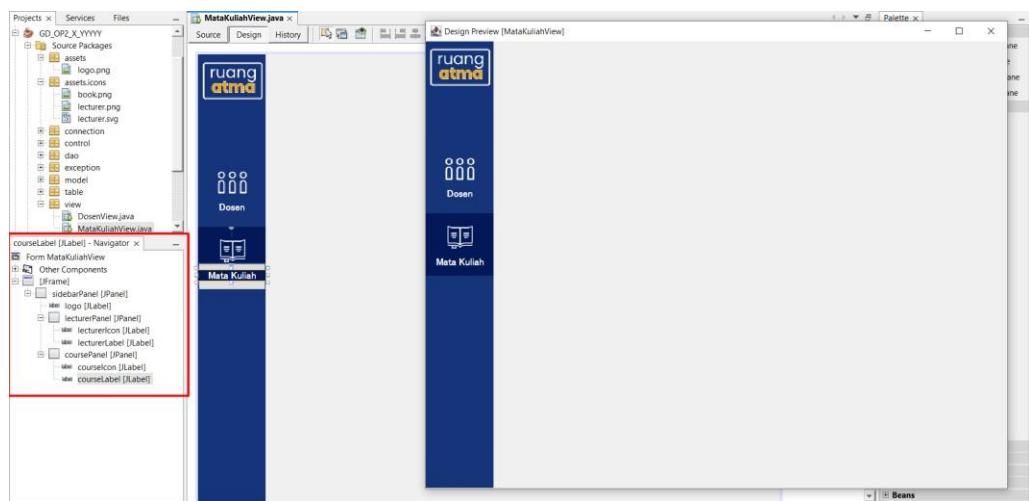
- s. Berikutnya *double click* pada komponen lecturerLabel dan ubah text label menjadi Dosen. Sehingga tampilannya akan menjadi seperti gambar di bawah ini.



- t. Masukan komponen lecturerPanel ke dalam komponen sidebarPanel dan posisikan sedemikian rupa seperti pada gambar di bawah ini. Lihat kotak merah dan pastikan susunan hirarki pada navigator sudah benar.



u. Setelah itu, buatlah menu kedua dengan nama panel `coursePanel` dengan cara yang sama seperti membuat `lecturerPanel`. Jangan lupa untuk mengubah nama variabel setiap kali menambahkan komponen baru. Jika sudah selesai maka tampilan akan sama seperti gambar di bawah ini. Masukan kode warna 7, 24, 89 pada background `coursePanel`. Pastikan navigator(kotak merah besar) sudah urut dan benar. Selalu ingat untuk melakukan klik *design preview*(kotak merah kecil) guna membantu melihat *design* secara windowed/full screen agar sesuai dengan yang sudah dibuat.



v. Anda juga dapat mengubah struktur urutan setiap komponen (hirarki) dengan melakukan *drag and drop*. Lihat gambar diatas, pada bagian navigator(kotak merah besar) untuk melakukan *drag and drop*.

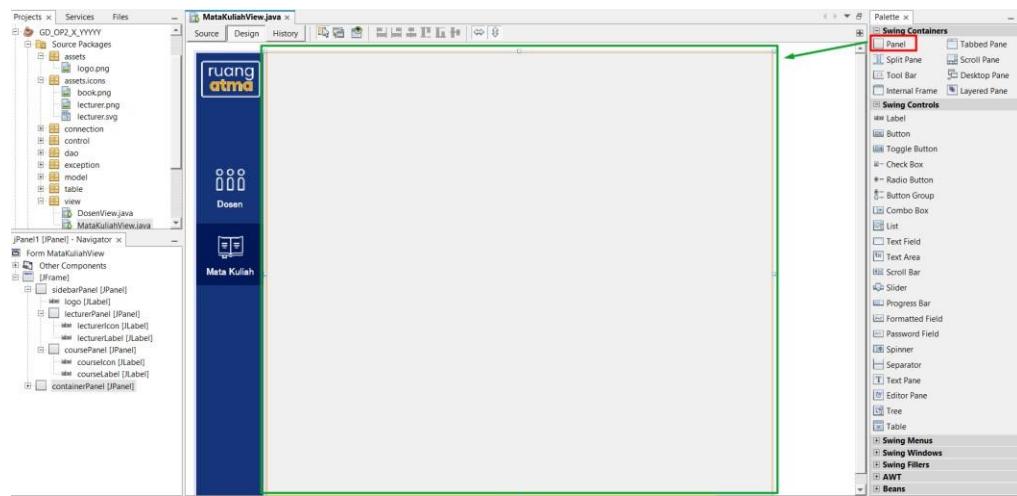
## PEMBUATAN KONTEN

- Pada bagian kedua kita akan membuat komponen untuk konten. Pada bagian Pallete pilih komponen Panel dan setting dengan aturan berikut ini.

Nama variabel : containerPanel

Background : 243,243,243

Size : memenuhi sisa field JFrame



- Pada containerPanel tambahkan 2 panel.

Panel 1:

Nama variabel : headerPanel

Background : 255,255,255

Border : Etched Border (shadow color : light gray)

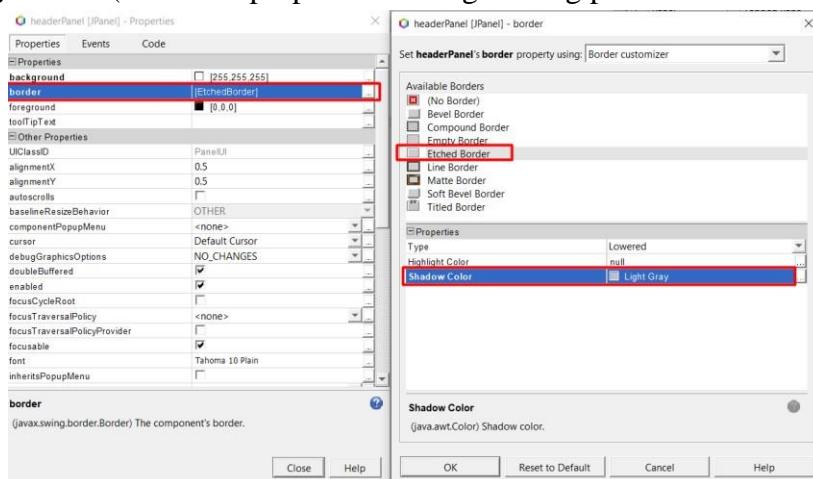
Panel 2 :

Nama variabel : contentPanel

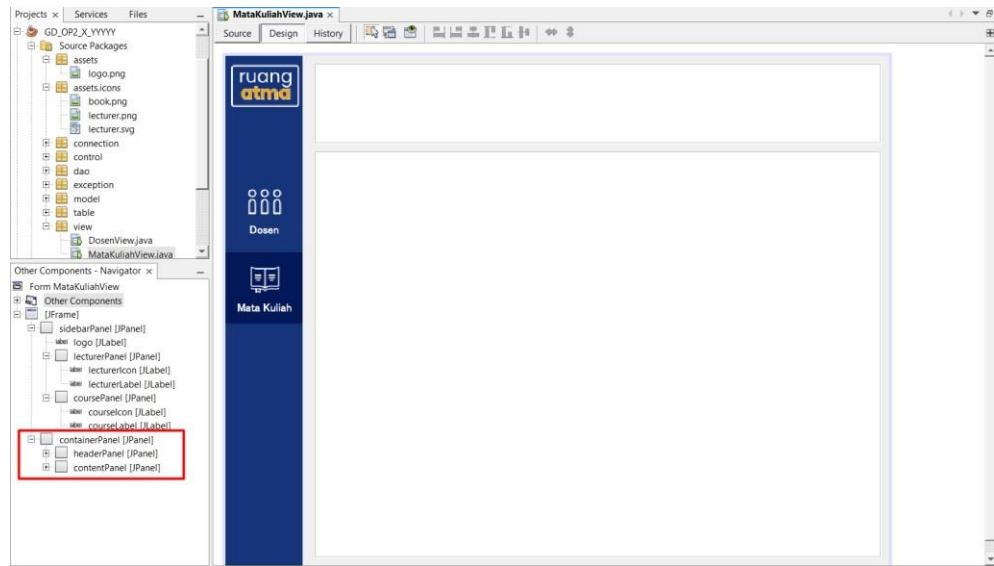
Background : 255,255,255

Border : Etched Border (shadow color : light gray)

**Setting border** (masuk ke properties masing-masing panel terlebih dahulu) :



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

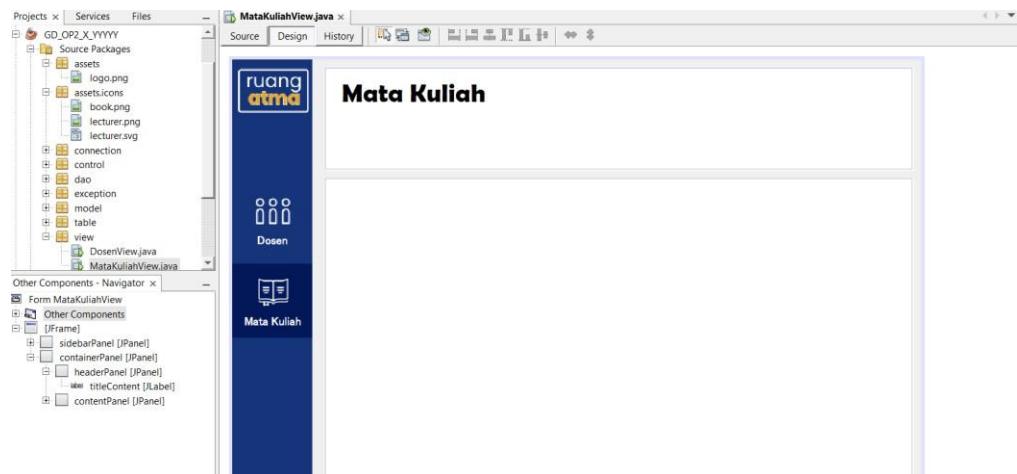


- c. Tambahkan Label ke dalam panel `headerPanel` sebagai judul menu dan dengan aturan di bawah ini.

Nama variabel : `titleContent`

Text : Mata Kuliah

Font : **Berlin Sans FB Demi** (optional) - Bold - 32



- d. Tambahkan 3 komponen button ke dalam `headerPanel` dengan aturan di bawah ini.

Button-1

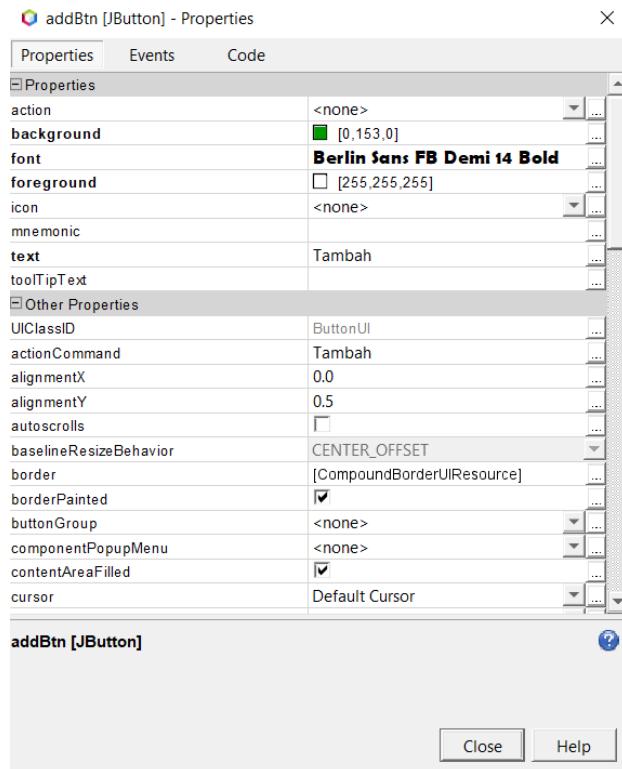
Nama variabel : `addBtn`

Text : Tambah

Background : 0,153,0

Foreground : 255,255,255

Font : **Berlin Sans FB Demi** (optional) - Bold - 14



Button-2

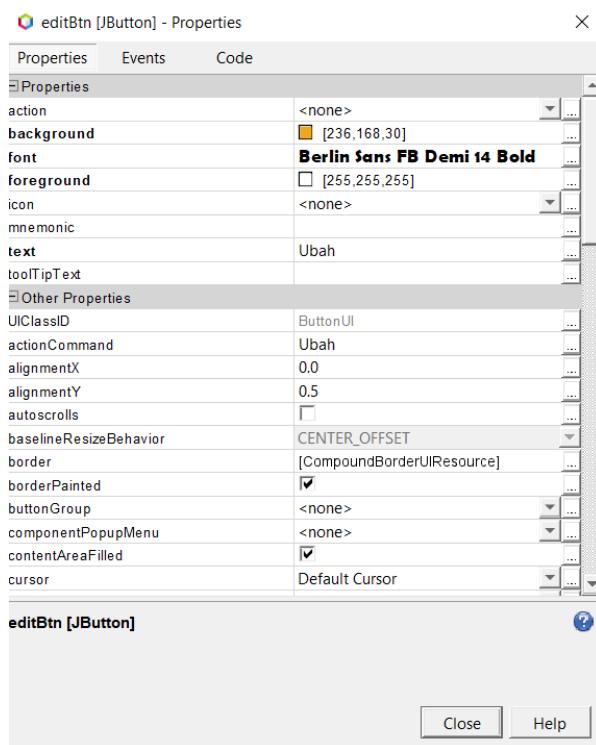
Nama variabel : editBtn

Text : Ubah

Background : 236,168,30

Foreground : 255,255,255

Font : **Berlin Sans FB Demi** (optional) - Bold - 14



Button-3

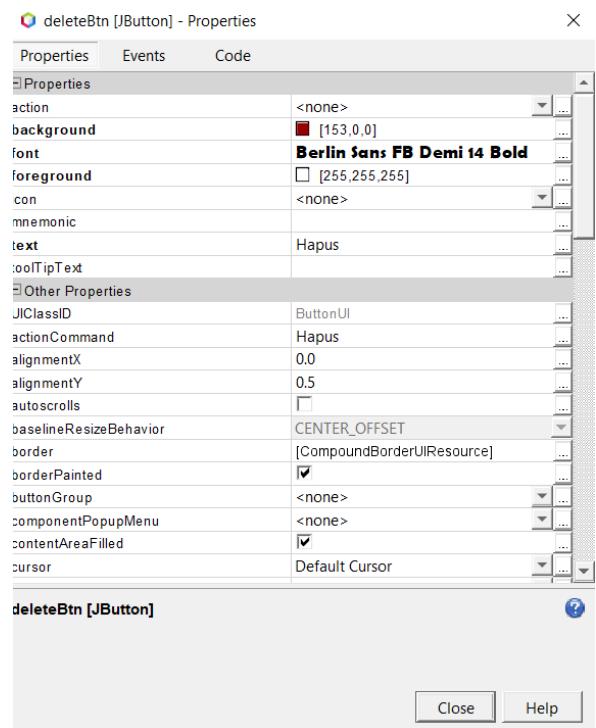
Nama variabel : deleteBtn

Text : Hapus

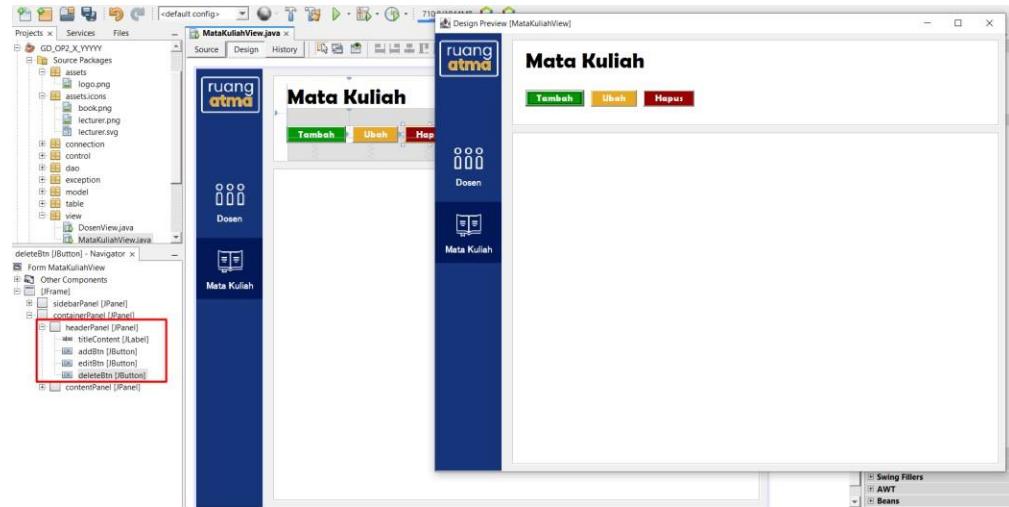
Background : 153,0,0

Foreground : 255,255,255

Font : **Berlin Sans FB Demi** (optional) - Bold - 14



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



- e. Tambahkan komponen button dan komponen Text Field ke dalam headerPanel dengan aturan di bawah ini.

#### Button

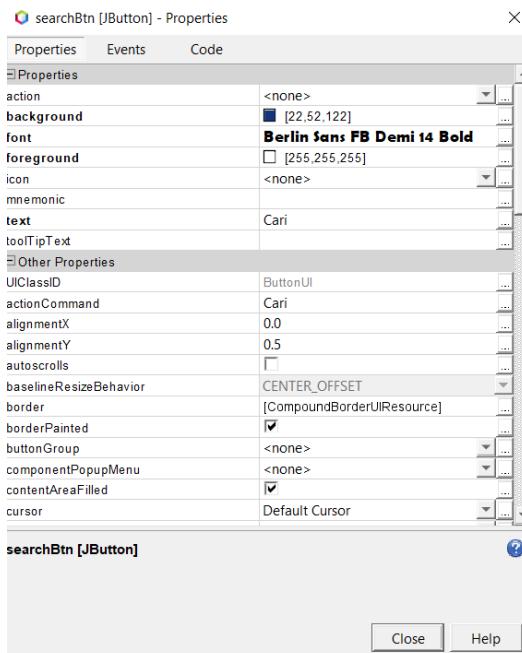
Nama variabel : searchBtn

Text : Cari

Background : 22,52,122

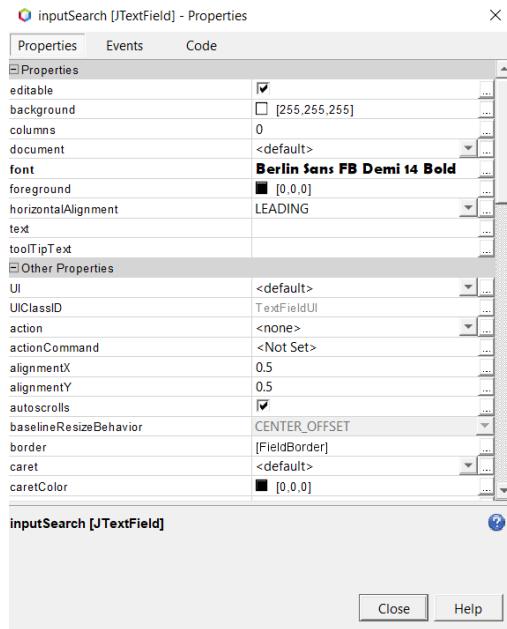
Foreground : 255,255,255

Font : **Berlin Sans FB Demi** (optional) - Bold - 14

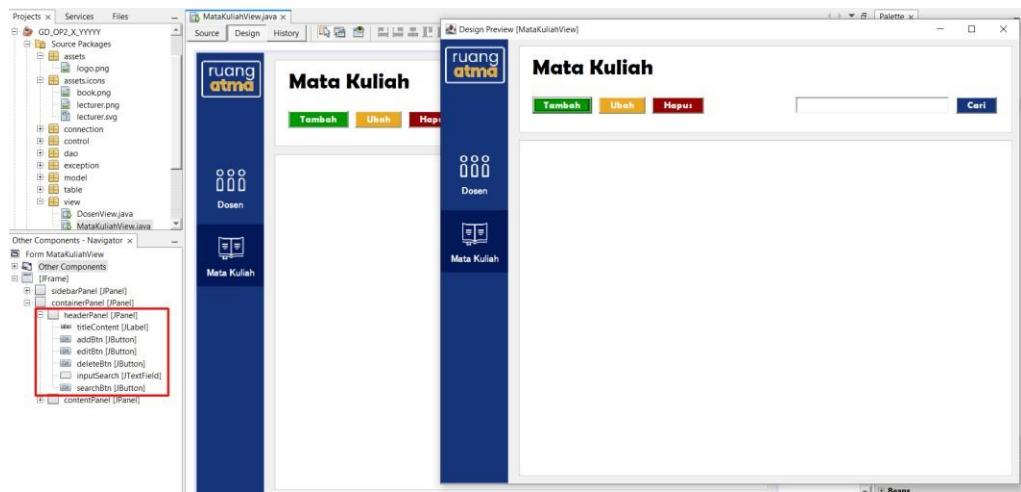


Nama variabel : inputSearch

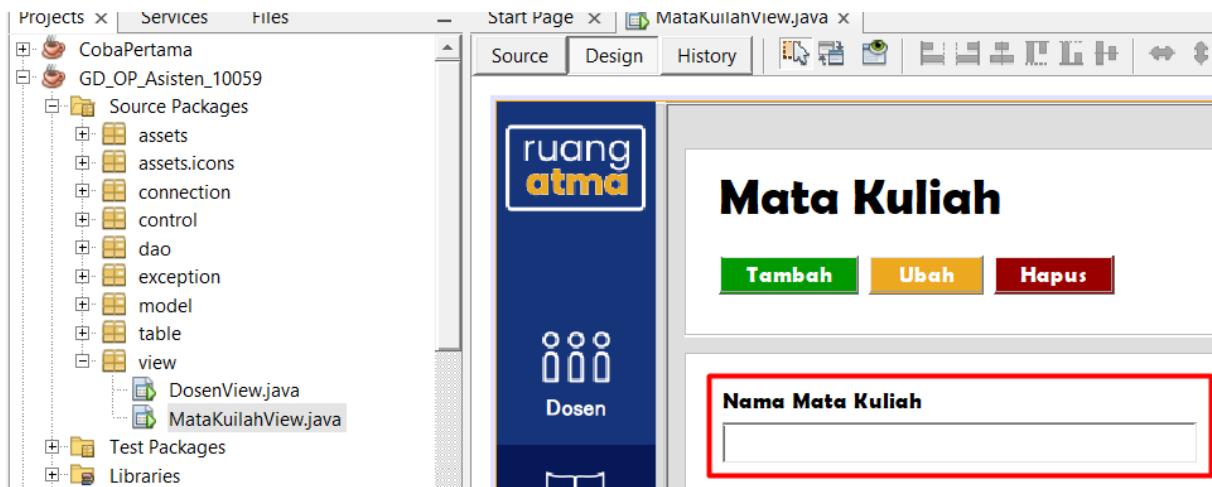
Font : **Berlin Sans FB Demi** (optional) - Plain - 14



Pastikan tampilan dan struktur komponen sudah sesuai dengan gambar di bawah ini. **Jika netbeans teman-teman bertema gelap, bisa mengubah Background berwarna putih(255,255,255) dan Foreground berwarna hitam(0,0,0) pada text field (optional).**

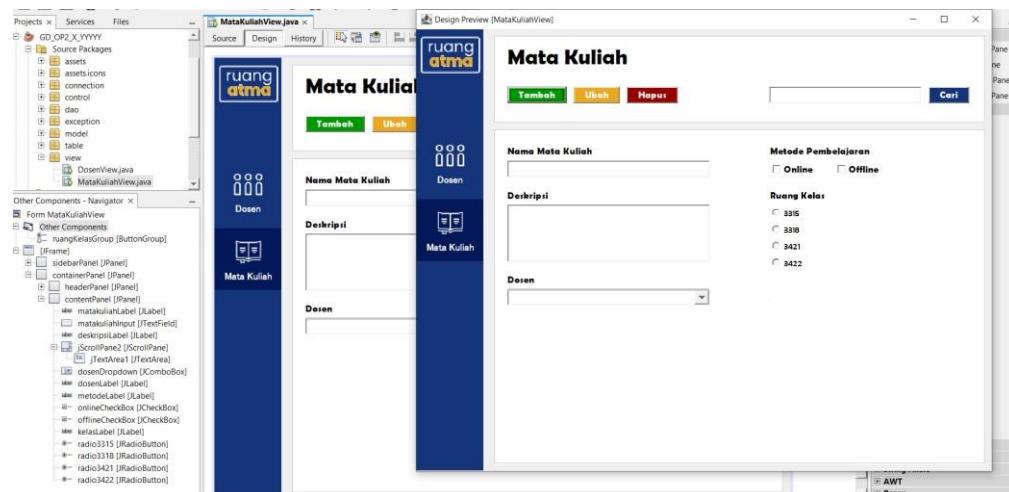


- f. Tambahkan komponen Label dan komponen Text Field ke dalam komponen `contentPanel` dan posisikan seperti gambar berikut ini



Silahkan buat komponen input dan label dengan aturan berikut ini.

Pastikan hasil akhir dan struktur hirarki sesuai dengan gambar dibawah ini.



- Komponen 1 (Nama Mata Kuliah)

Komponen : Text Field dan Label

Nama variabel Label : matakuliahLabel

Font Label & Field: **Berlin Sans FB Demi** (optional) - Plain - 14

Nama variabel Input : matakuliahInput

- Komponen 2 (Deskripsi)

Komponen : Text Area & Label

Nama variabel Label : deskripsiLabel

Font Label & Text Are: **Berlin Sans FB Demi** (optional) - Plain - 14

Nama variabel Text Area: deskripsiInput

- Komponen 3 (Dosen)

Komponen : Combo Box & Label

Nama variabel Label : dosenLabel

Font Label & Combo Box: **Berlin Sans FB Demi** (optional) - Plain - 14

Nama variabel Combo Box : dosenDropdown

- Komponen 4 (Metode Pembelajaran)

Komponen : Check Box & Label

Nama variabel Label : metodeLabel

Font Label & Check Box : **Berlin Sans FB Demi** (optional) - Plain - 14

Nama variabel Check Box 1 : onlineCheckBox

Nama variabel Check Box 2 : offlineCheckBox

Background : 255, 255, 255

- Komponen 5 (Ruang Kelas)

Komponen : Radio Button, Button Group, & Label

Nama variabel Label : kelasLabel

Font Label : **Berlin Sans FB Demi** (optional) - Plain – 14

Font Radio : **Berlin Sans FB** (optional) - Plain – 14

Background : 255, 255, 255

Nama variabel Button Group : ruangKelasGroup **(Baca keterangan dibawah)**

Nama variabel Radio Button 1 : radio3315

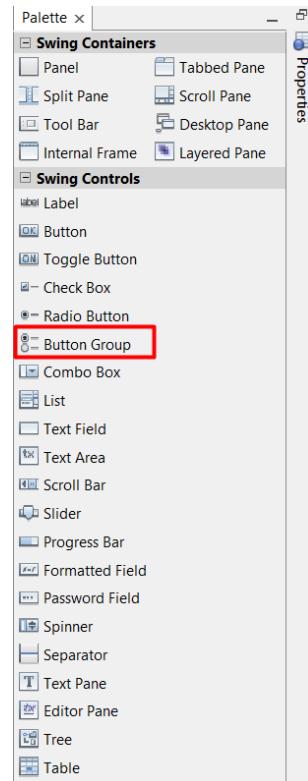
Nama variabel Radio Button 2 : radio3318

Nama variabel Radio Button 3 : radio3421

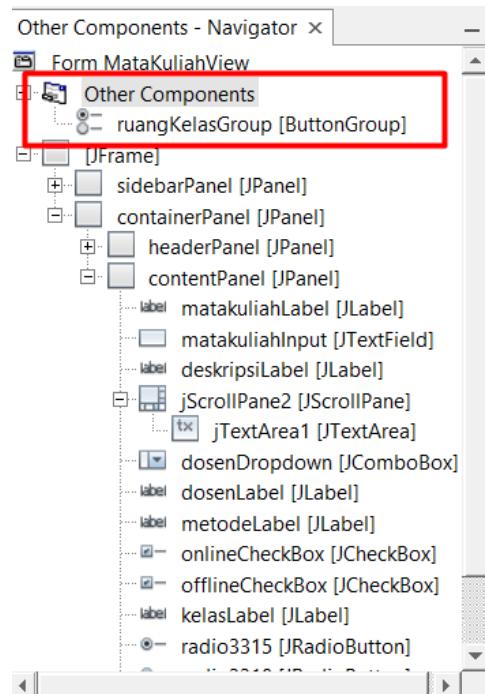
Nama variabel Radio Button 4 : radio3422

## Keterangan Button Group :

Button Group dapat ditemukan di palette seperti gambar dibawah ini.

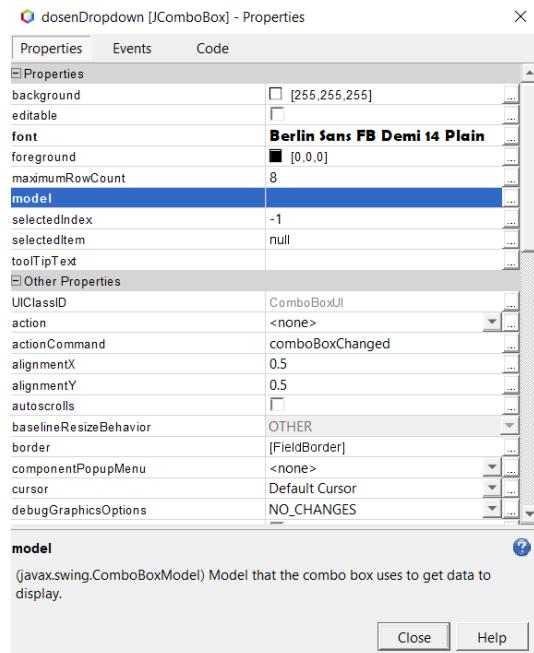


Kemudian Lakukan *drag and drop* Button Group kearah field. Maka button group akan muncul pada navigator di dalam Other Components. Lakukan rename pada button group sesuai ketentuan diatas.



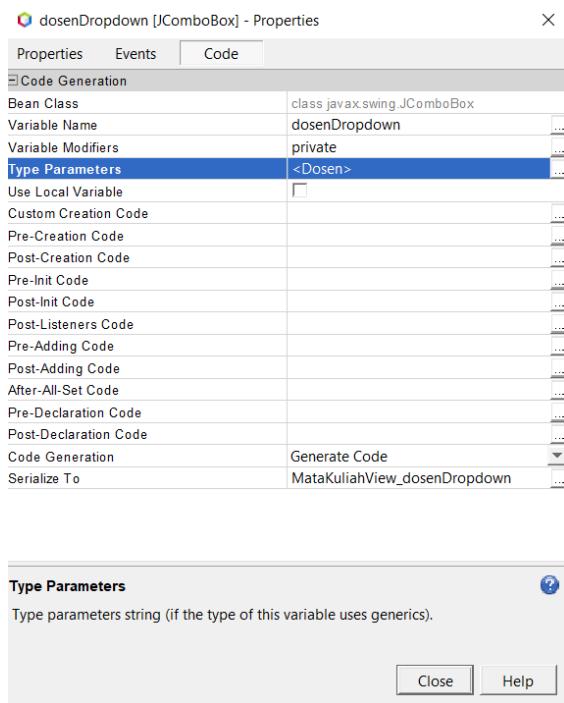
## PENTING :

- Pada komponen Combo Box `dosenDropdown` pada Properties, ubah **model** menjadi kosong.



- Pada komponen Combo Box `dosenDropdown` ubah Type Parameternya menjadi `<Dosen>` dengan cara klik properties dan masuk ke tab code seperti di bawah ini. Hal ini dilakukan karena secara default, item pada combobox hanya bisa menerima tipe String. Dalam kasus ini, kita ingin pengguna agar dapat memilih berdasarkan nama dosen dan ketika objek mata kuliah dibuat maka akan disimpan objek dosen yang dipilih. Sehingga tidak perlu melakukan pencarian kembali pada basis data

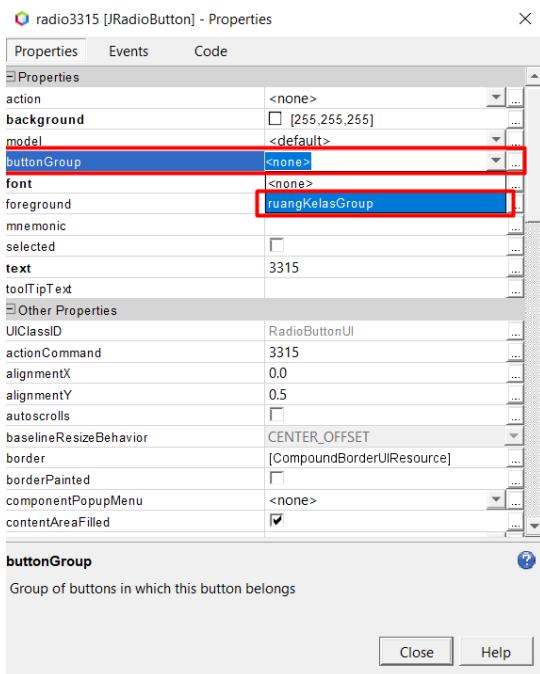
berdasarkan nama dosen yang dipilih. (Jika terjadi eror tidak masalah, lanjutkan saja dulu.)



- Lalu tambahkan method berikut pada kelas Dosen.java. Hal ini dilakukan untuk mengatur value apa yang akan ditampilkan di dropdown. Seperti yang sudah disampaikan sebelumnya, item pada dropdown hanya bisa menampung String. Maka disini kita override fungsi tersebut untuk mereturnkan nilai nama sehingga value yang muncul pada item dropdown adalah nama dosen.

```
@Override  
public String toString() {  
    return nama;  
}
```

- Pada setiap komponen radio button, set parameter buttonGroup menjadi ruangKelasGroup dengan cara masuk ke properties pada setiap komponen radio button. Hal ini dilakukan untuk melakukan grouping radio button yang membuat dalam satu group radio button hanya ada satu pilihan. Mudahnya ketika salah satu radio dipilih (bernilai true) maka pilihan lain akan otomatis mati (bernilai false)



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

#### Button-1

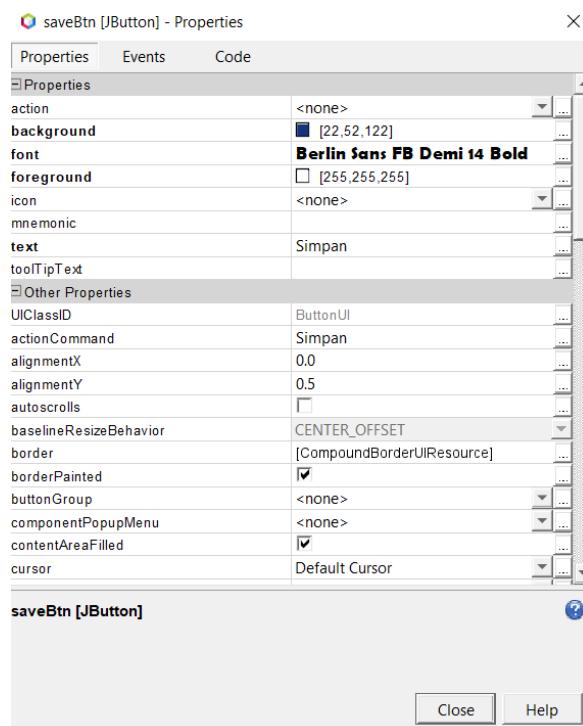
Nama variabel : saveBtn

Background : 22,52,122

Font : **Berlin Sans FB Demi** (optional) - Bold - 14

Foreground : 255,255,255

Text : Simpan



#### Button-2

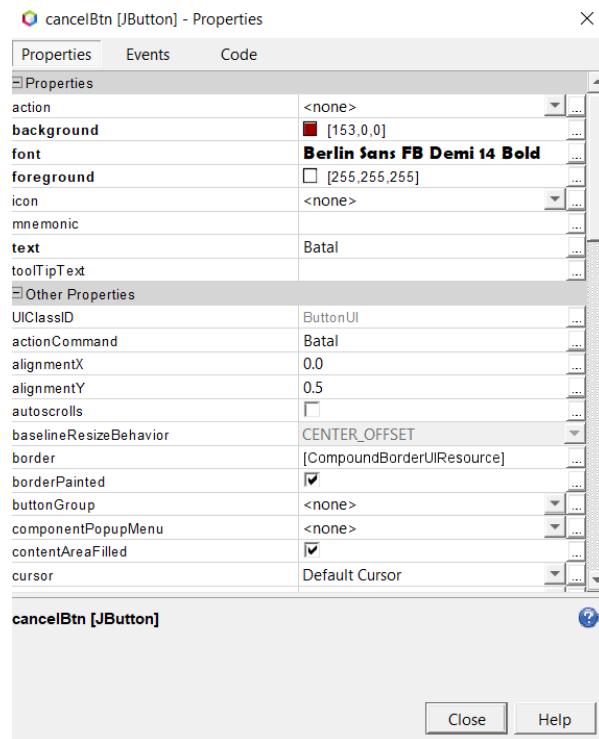
Nama variabel : cancelBtn

Background : 153,0,0

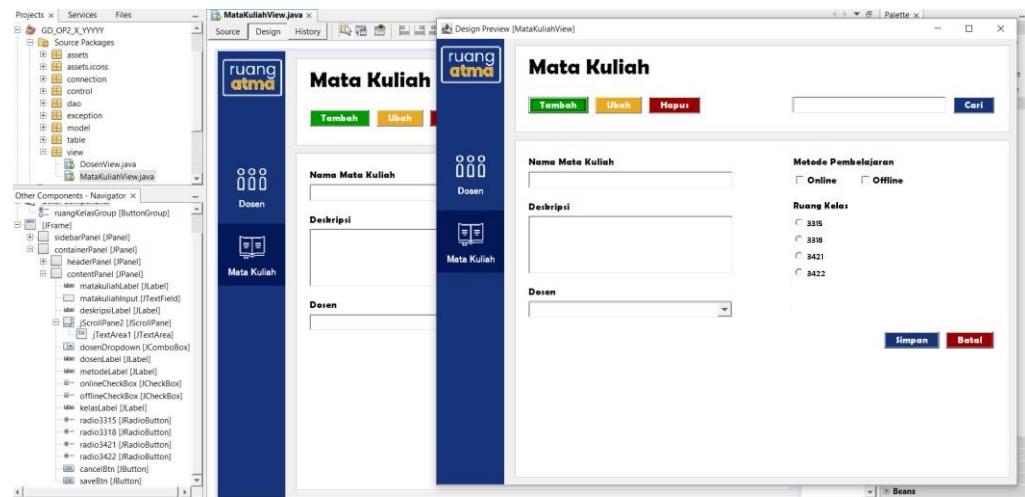
Font : **Berlin Sans FB Demi** (optional) - Bold - 14

Foreground : 255,255,255

Text : Batal



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

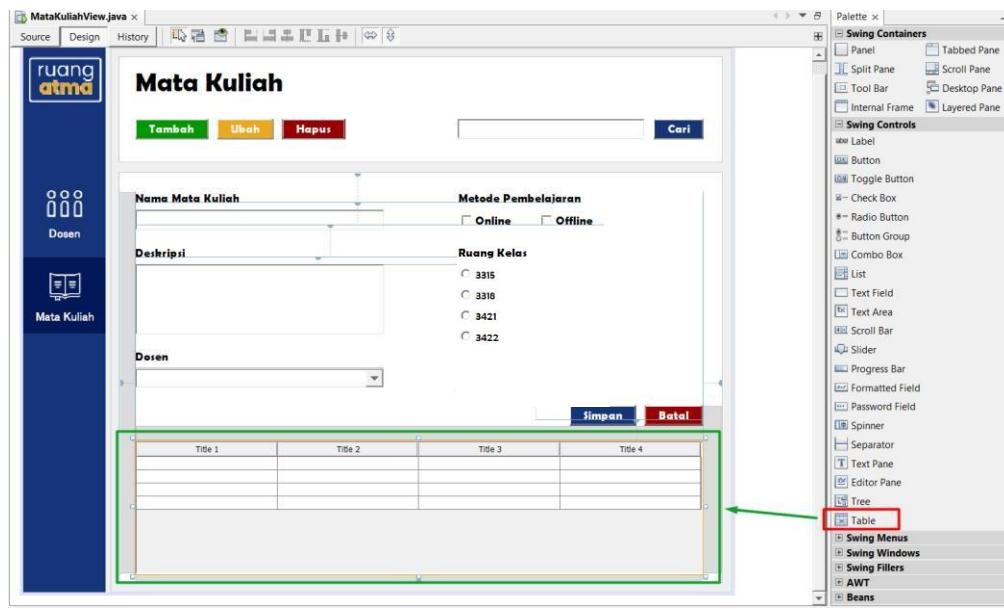


- Tambahkan komponen Table ke dalam `contentPanel` dengan aturan berikut ini.

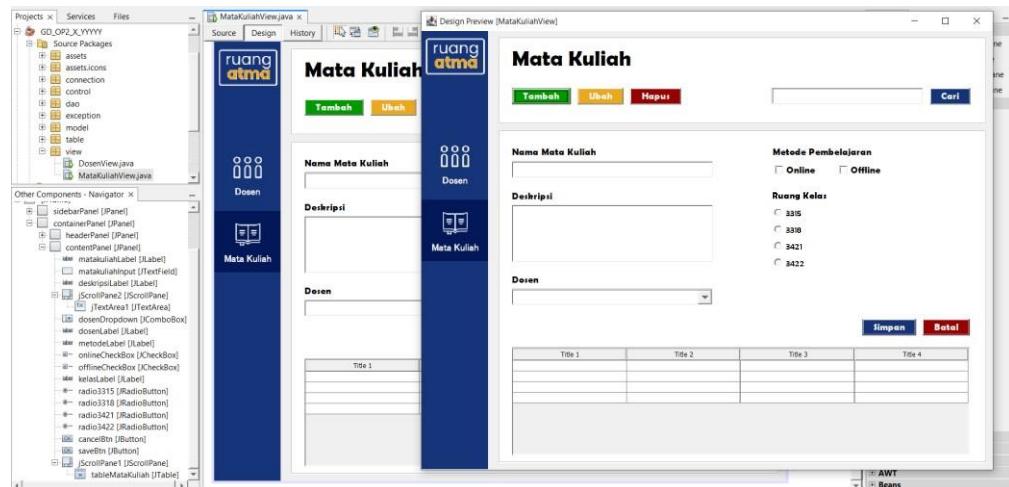
Nama variabel : `tableMataKuliah`

Width : Sesuaikan dengan sisa field yang ada di `contentPanel`

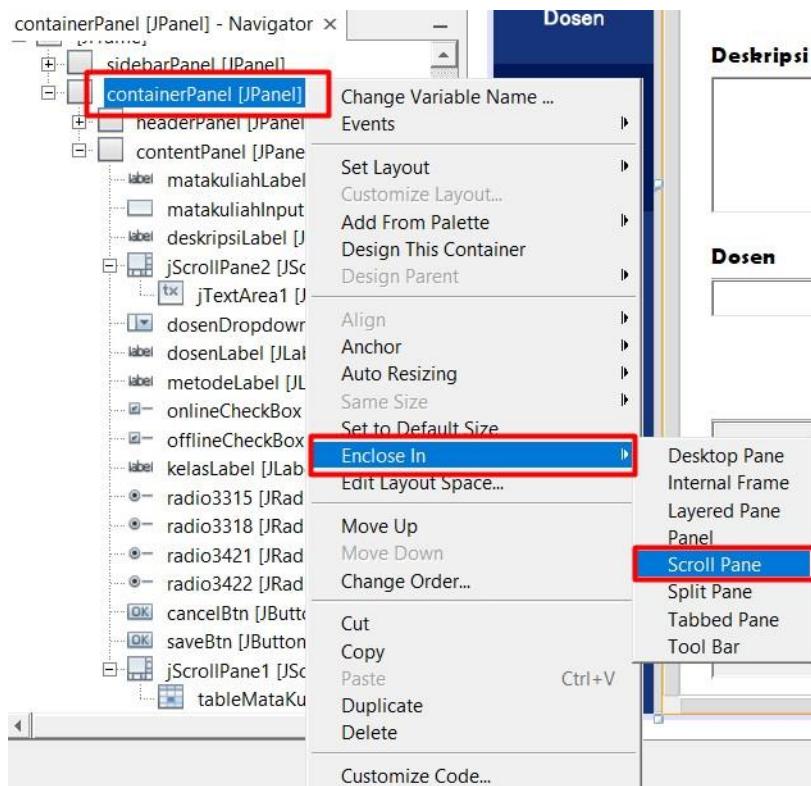
Height : Sesuaikan dengan sisa field yang ada di `contentPanel`



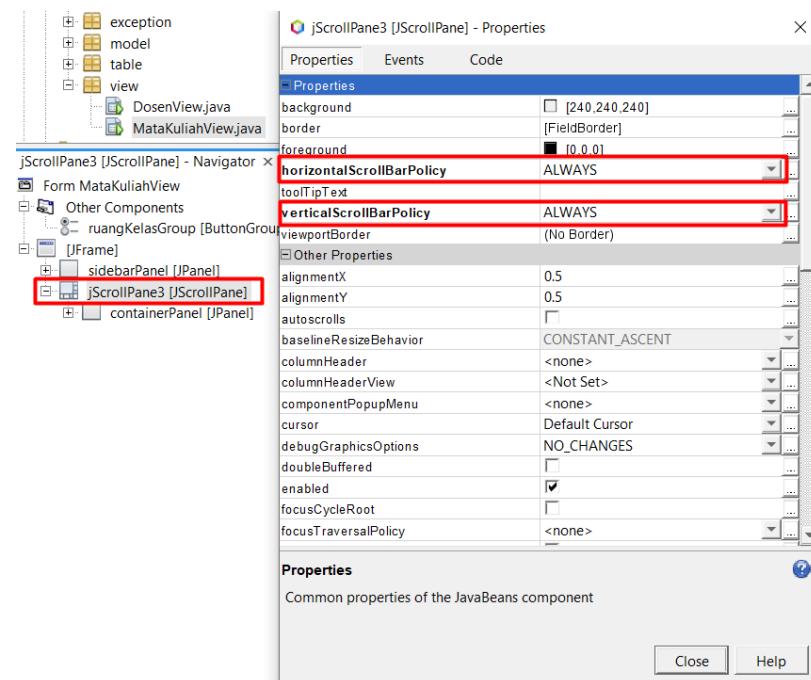
Pastikan tampilan dan struktur komponen sudah sesuai dengan gambar di bawah ini



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



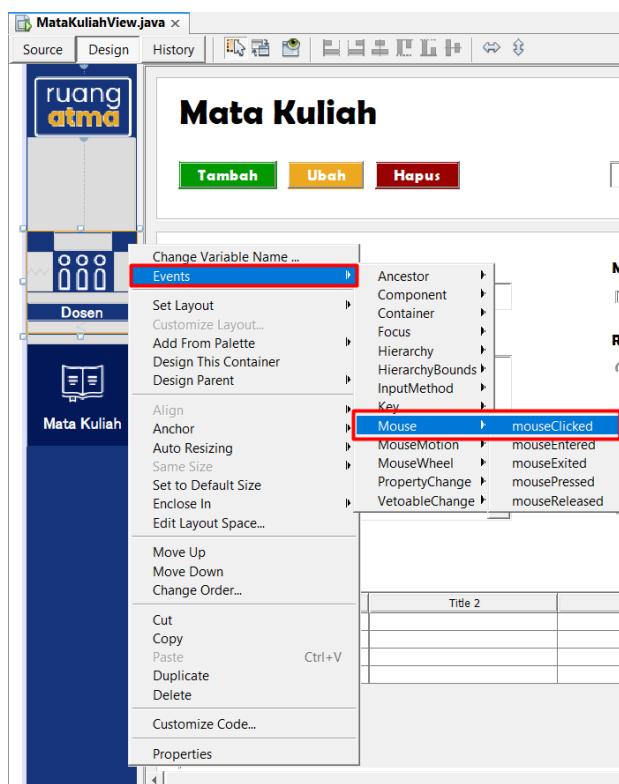
Lalu pada bagian kanan layar akan muncul bagian properties. Ubah nilai dari **horizontalScrollBarPolicy** dan **verticalScrollBarPolicy** menjadi **ALWAYS**.



## INTEGRASI LOGIC DENGAN UI

Setelah selesai membuat tampilan UI, maka langkah selanjutnya adalah melakukan “coding” untuk menghubungkan *business logic* dan *presentation logic*.

- Klik kanan pada panel lecture lalu pilih opsi berikut.



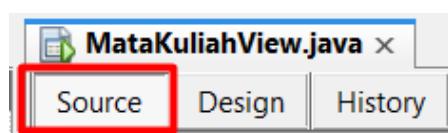
Perintah tersebut akan mengarahkan ke bagian dimana ketika `lecturePanel` mendapatkan *event* klik dari *mouse*

- Setelah itu akan muncul tampilan seperti ini dan ketikan code berikut :

```
private void lecturerPanelMouseClicked(java.awt.event.MouseEvent evt) {  
    // code untuk pindah ke halaman DosenView.java  
    DosenView dv = new DosenView();  
    this.dispose();  
    dv.setVisible(true);  
}
```

Kode di atas berfungsi untuk mengarahkan ke halaman DosenView lalu menghilangkan halaman saat ini dengan perintah `dispose()`.

- Klik tab “Source” pada bagian kiri atas MataKuliahView.



*Import library* dan kelas yang dibutuhkan:

```
package view;

import control.DosenControl;
import control.MataKuliahControl;
import exception.InputKosongException;
import java.util.List;
import javax.swing.JOptionPane;
import javax.swing.table.TableModel;
import model.Dosen;
import model.MataKuliah;
import table.TableMataKuliah;
```

- d. Lakukan deklarasi variabel dan pembuatan object DosenControl serta MataKuliahControl pada constructor :

```
private DosenControl dosenControl;
private MataKuliahControl mataKuliahControl;
String action = null;
List<Dosen> listDosen;
int selectedId = 0;

public MataKuliahView() {
    initComponents();
    setComponent(false);
    setRadioKelas(false);
    setEditDeleteBtn(false);
    dosenControl = new DosenControl();
    mataKuliahControl = new MataKuliahControl();
    showMataKuliah();
    setActionCommandRadio();
    setDosenToDropdown();
}
```

- e. Tambahkan method seperti berikut pada **DosenControl** untuk menyiapkan data dosen untuk dropdown :

```
public List<Dosen> showListDosen() {
    List<Dosen> dataDosen = dDao.showDosen();
    return dataDosen;
}
```

- f. Tambahkan method seperti berikut pada MataKuliahView untuk menampilkan data dari DosenControl pada pilihan dropdown :

```
public void setDosenToDropdown() {  
    listDosen = dosenControl.showListDosen();  
    for (int i = 0; i < listDosen.size(); i++) {  
        dosenDropdown.addItem(listDosen.get(i));  
    }  
}
```

Fungsi `setDosenToDropdown` akan mengeset pilihan yang ada pada `dosenDropdown`. Pertama dilakukan pengambilan data dengan memanggil method `showListDosen()` yang berada pada `dosenControl` yang ditampung pada variabel `listDosen` bertipe `List`. Selanjutnya, dari data yang sudah diambil akan dilakukan perulangan sebanyak data yang ada di `listDosen`. Setiap iterasi akan dilakukan pengesetan data ke `dropdown` dengan memanggil fungsi `addItem` pada `dosenDropdown` dengan parameter data dosen pada index tersebut (`listDosen.get(i)`).

- g. 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) {
    matakuliahInput.setEnabled(value);
    deskripsiInput.setEnabled(value);
    dosenDropdown.setEnabled(value);
    onlineCheckBox.setEnabled(value);
    offlineCheckBox.setEnabled(value);

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

public void setRadioKelas(Boolean value) {
    radio3315.setEnabled(value);
    radio3318.setEnabled(value);
    radio3421.setEnabled(value);
    radio3422.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() {
    matakuliahInput.setText("");
    deskripsiInput.setText("");
    dosenDropdown.setSelectedItem(ABORT);
    onlineCheckBox.setSelected(false);
    offlineCheckBox.setSelected(false);

    ruangKelasGroup.clearSelection();
    searchInput.setText("");
}

```

- h. Tambahkan method `setActionCommandRadio` yang berfungsi untuk mengatur value index pada setiap radio button.

```

public void setActionCommandRadio() {
    radio3315.setActionCommand("3315");
    radio3318.setActionCommand("3318");
    radio3421.setActionCommand("3421");
    radio3422.setActionCommand("3422");
}

```

- i. Tambahkan method `showMataKuliah` untuk menampilkan data mata kuliah pada tabel. Pada DAO untuk menampilkan mata kuliah, dibutuhkan parameter

input berupa String untuk query pencarian. Maka dari itu berikan String kosong pada method ini agar dapat mengambil semua data pada database.

```
// Method untuk menampilkan data mata kuliah pada tabel
public void showMataKuliah() {
    tableMataKuliah.setModel(mataKuliahControl.showMataKuliah(""));
}
```

- j. Tambahkan juga method yang bernama `inputKosongException` 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. Berikut adalah potongan kodennya :

```
// Method untuk memeriksa input kosong
public void inputKosongException() throws InputKosongException {
    if (matakuliahInput.getText().isEmpty() || deskripsiInput.getText().isEmpty()
        || dosenDropdown.getSelectedIndex() == -1) {
        throw new InputKosongException();
    }

    if (!onlineCheckBox.isSelected() && !offlineCheckBox.isSelected()) {
        throw new InputKosongException();
    }

    if (offlineCheckBox.isSelected() && ruangKelasGroup.getSelection() == null) {
        throw new InputKosongException();
    }
}
```

Pada exception ini terdapat beberapa pengecekan. Pertama, adalah jika mata kuliah, deskripsi, atau dropdown tidak diisi maka akan dilakukan `throw InputKosongException()`. Kedua, jika kedua checkBox tidak ada yang diisi akan dilakukan `throw InputKosongException()`. Ketiga, jika offlineCheckBox dipilih dan ruang kelas tidak ada yang dipilih maka akan dilakukan `throw InputKosongException()`.

- k. 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);
    setRadioKelas(false);
    ruangKelasGroup.clearSelection();

    clearText();
    searchInput.setText("");
    action = "Tambah";
    dosenDropdown.setSelectedIndex(0);
}
```

Ketika tombol add ditekan maka seluruh komponen input akan diaktifkan dengan memanggil fungsi `setComponent(true)`. Untuk komponen radio button akan aktif ketika pengguna memilih metode pengajaran offline sehingga pada bagian ini radio button akan dinonaktifkan. Selanjutnya, setiap inputan yang ada akan dibersihkan dengan memanggil fungsi `clearSelection()` untuk radio button, `clearText()` untuk input yang lain, `setSelectedIndex` untuk mengeset `dosenDropdown` pada pilihan index pertama. Lalu kita juga akan mengeset variabel action menjadi “Tambah” yang berguna untuk mengetahui kondisi aksi yang harus dilakukan saat menyimpan data.

1. 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 kan mengaktifkan component Input
    setComponent(true);
    action = "Ubah";
    Boolean value = false;
    if (offlineCheckBox.isSelected()) {
        value = true;
    }
    setRadioKelas(value);
}
```

Ketika tombol edit ditekan maka seluruh komponen input akan diaktifkan dengan memanggil fungsi `setComponent(true)`. Setelah itu dilakukan pengeset variabel action menjadi “Ubah”. Ketika data yang diubah memiliki metode pengajaran offline maka komponen radio button akan diaktifkan.

- m. 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) {
    // TODO add your handling code here;
    // Ketika button delete diklik maka kan menampilkan modal
    int getAnswer = JOptionPane.showConfirmDialog(rootPane, "Apakah yakin ingin menghapus data ?", "Konfirmasi", JOptionPane.YES_NO_OPTION);
    switch (getAnswer) {
        case 0:
            //jika jawaban user adalah Yes
            try {
                mataKuliahControl.deleteDataMataKuliah(selectedId);
                clearText();
                showMataKuliah();
                setComponent(false);
                setRadioKelas(false);
                setEditDeleteBtn(false);
                ruangKelasGroup.clearSelection();
            } catch (Exception e) {
                System.out.println("Error : " + e.getMessage());
            }
            break;
        case 1:
            //jika jawaban user adalah No
            break;
    }
}
```

Ketika tombol delete ditekan maka akan muncul dialog konfirmasi yang akan mengembalikan nilai berupa `integer` dengan nilai 0 berarti yes dan 1 berarti

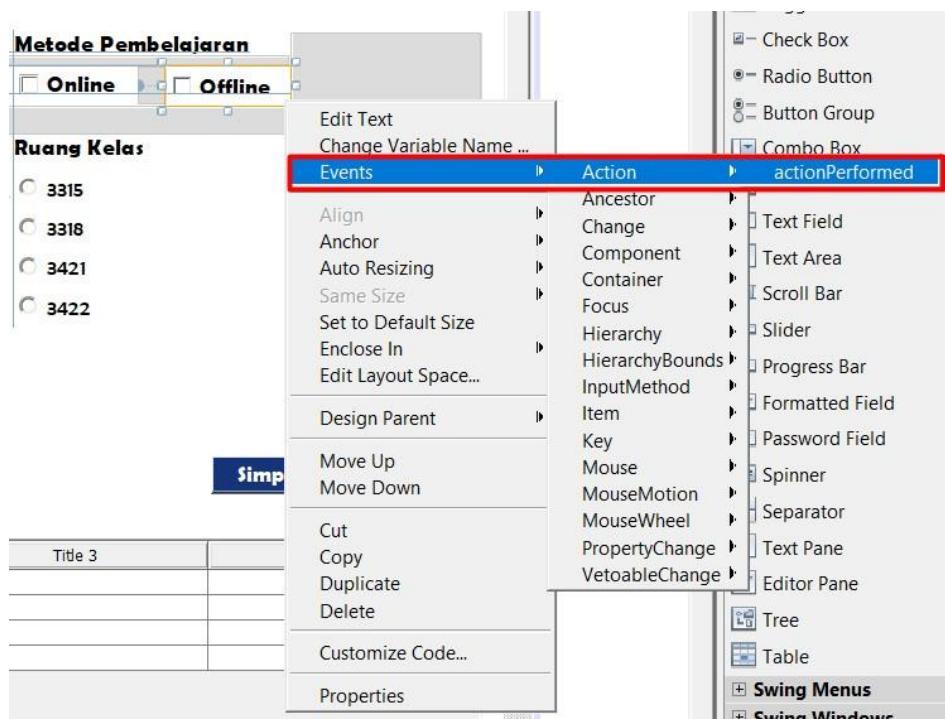
no. Ketika pengguna memilih yes maka langkah yang dilakukan adalah memanggil method `deleteDataMataKuliah(selectedId)` dari `mataKuliahControl` yang akan melakukan proses penghapusan data dari basis data. Selanjutnya, setiap inputan yang ada akan dibersihkan dengan memanggil fungsi `clearSelection()` untuk radio button group, `clearText()` untuk input yang lain, `setComponent(false)` untuk mematikan field input, `setRadioKelas(false)` untuk mematikan radio button serta `showMataKuliah()` untuk mengupdate data pada table.

- n. 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 kan nonaktifkan component Button edit dan Delete
    setEditDeleteBtn(false);
    // Ketika button cari diklik maka kan nonaktifkan component Input
    setComponent(false);

    try {
        TableMataKuliah matakuliah = mataKuliahControl.showMataKuliah(searchInput.getText());
        if (matakuliah.getRowCount() == 0) {
            //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
            tableMataKuliah.setModel(matakuliah);
        }
        // Ketika button cari diklik maka kan menghapus seluruh isi input
        clearText();
    } catch (Exception e) {
        System.out.println("Error: " + e.getMessage());
    }
}
```

- o. Klik kanan pada *offline* check Box dan pilih opsi berikut.



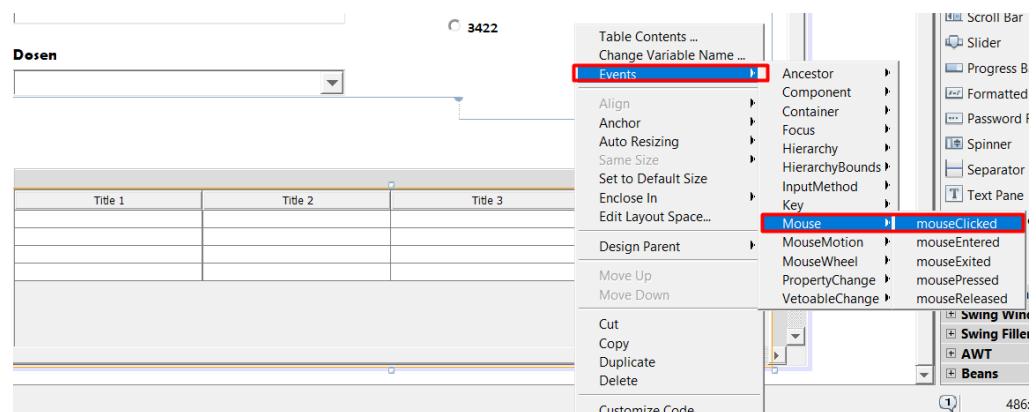
Perintah tersebut akan mengarahkan kita ke bagian dimana ketika offline check box mendapatkan aksi.

- p. Lalu akan muncul tampilan seperti di bawah ini dan ketikkan code berikut :

```
private void offlineCheckBoxActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    Boolean value = false;  
    if (offlineCheckBox.isSelected()) {  
        value = true;  
    }  
    setRadioKelas(value);  
    ruangKelasGroup.clearSelection();  
}
```

Pada fungsi tersebut dilakukan pembuatan variabel bertipe boolean dengan value *false*. Ketika *offline* check box dipilih maka variabel value akan diset menjadi *true*. Variabel tersebut digunakan sebagai value untuk mengeset komponen radio button yang ada. Untuk memastikan tidak ada radio yang dipilih maka dilakukan pemanggilan fungsi `clearSelection()`.

- q. Klik kanan pada tabel mata kuliah dan pilih opsi berikut.



Perintah tersebut akan mengarahkan kita ke bagian dimana ketika data pada tabel mendapatkan *event* klik dari *mouse*.

- r. Lalu akan muncul tampilan seperti di bawah ini dan ketikkan code berikut

```
private void tableMatakuliahMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int indexDosen = -1;
    setEditDeleteBtn(true);

    setComponent(false);
    setRadioKelas(false);
    ruangKelasGroup.clearSelection();

    int clickedRow = tableMatakuliah.getSelectedRow();
    TableModel tableModel = tableMatakuliah.getModel();

    selectedId = Integer.parseInt(tableModel.getValueAt(clickedRow, 5).toString());
    matakuliahInput.setText(tableModel.getValueAt(clickedRow, 0).toString());
    deskripsiInput.setText(tableModel.getValueAt(clickedRow, 1).toString());

    String metodePembelajaran = tableModel.getValueAt(clickedRow, 3).toString();
    switch (metodePembelajaran) {
        case "Hybrid":
            onlineCheckBox.setSelected(true);
            offlineCheckBox.setSelected(true);
            break;
        case "Online":
            onlineCheckBox.setSelected(true);
            offlineCheckBox.setSelected(false);
            break;
        case "Offline":
            onlineCheckBox.setSelected(false);
            offlineCheckBox.setSelected(true);
            break;
    }
}
```

```

        String ruangKelas = tableModel.getValueAt(clickedRow, 4).toString();
        switch (ruangKelas) {
            case "3315":
                radio3315.setSelected(true);
                break;
            case "3318":
                radio3318.setSelected(true);
                break;
            case "3421":
                radio3421.setSelected(true);
                break;
            case "3422":
                radio3422.setSelected(true);
                break;
        }

        deskripsiInput.setText(tableModel.getValueAt(clickedRow, 1).toString());

        String nomor_induk_dosen = tableModel.getValueAt(clickedRow, 6).toString();
        for(Dosen dosen : listDosen) {
            if (dosen.getNomorIndukDosen().equals(nomor_induk_dosen)) {
                indexDosen = listDosen.indexOf(dosen);
            }
        }

        dosenDropdown.setSelectedIndex(indexDosen);
    }
}

```

Pertama variabel int index dosen berfungsi untuk menampung id dosen yang dipilih dari array dosenDropdown. Ketika memilih data di tabel maka tombol edit dan delete akan diaktifkan dengan memanggil fungsi setEditDeleteBtn(true) untuk dapat dilakukan aksi selanjutnya dan menonaktifkan seluruh komponen input. Untuk mendapatkan data yang diklik pengguna dari tabel dilakukan penyimpanan id baris dengan memanggil fungsi getSelectedRow() pada tableMatakuliah. Berikutnya dilakukan pengambilan model data dari tabel tableMatakuliah dengan fungsi getModel(). Untuk menampung id yang akan edit dilakukan pengesetan pada variabel global selectedId dengan mengambil data pada tableModel kolom ke 5 sesuai switch case yang telah diset sebelumnya. Berikutnya dilakukan pengesetan value pada input nama mata kuliah dan deskripsi dengan fungsi setText. Data diambil dari tableModel dan di set sesuai index switch case yang telah diset sebelumnya. Lalu, untuk mengeset checkbox dan radio akan dilakukan pengecekan menggunakan switch case.

- s. 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();
        String metodePembelajaran = "...";
        String kelas = "...";
        int selectedIndex = dosenDropdown.getSelectedIndex();
        Dosen selectedDosen = listDosen.get(selectedIndex);

        if (onlineCheckBox.isSelected() && offlineCheckBox.isSelected()) {
            metodePembelajaran = "Hybrid";
            kelas = ruangKelasGroup.getSelection().getActionCommand();
        } else if (onlineCheckBox.isSelected()) {
            metodePembelajaran = "Online";
            kelas = "Online";
        } else {
            metodePembelajaran = "Offline";
            kelas = ruangKelasGroup.getSelection().getActionCommand();
        }

        if (action.equals("Tambah")) {
            //Ketika action berstatus tambah maka akan memanggil fungsi insert
            MataKuliah mk = new MataKuliah(matakuliahInput.getText(),
                deskripsiInput.getText(), metodePembelajaran, kelas, selectedDosen);
            matakuliahControl.insertDataMataKuliah(mk);
        } else {
            //Ketika action berstatus ubah maka akan memanggil fungsi update
            MataKuliah mk = new MataKuliah(selectedId, matakuliahInput.getText(),
                deskripsiInput.getText(), metodePembelajaran, kelas, selectedDosen);
            matakuliahControl.updateDataMataKuliah(mk);
        }
        clearText();
        showMataKuliah();
        setComponent(false);
        setRadioKelas(false);
        ruangKelasGroup.clearSelection();

        setEditDeleteBtn(false);
    } catch (InputKosongException e) {
        JOptionPane.showMessageDialog(this, e.message());
    }
}

```

Ketika tombol save ditekan maka pertama kali dilakukan pemanggilan `inputKosongException()` untuk memvalidasi masukkan pengguna. Setelah itu dilakukan pengambilan data dari dropdown dengan menggunakan `getSelectedIndex()` dan mengambil data dosen terpilih dengan mengambil data dari `listDosen` dengan index dari data yang dipilih dari dropdown. Selanjutnya mengambil nilai metode pembelajaran dan kelas dengan menggunakan kondisi sesuai dengan studi kasus yang sudah diterapkan.

Setelah mendapatkan semua data, maka akan dilakukan pengecekan apabila action yang terpilih adalah “Tambah” atau “Ubah”. Ketika action yang dipilih adalah “Tambah” maka akan dibentuk objek `MataKuliah` baru tanpa id sedangkan sebaliknya apabila action yang dipilih adalah “Ubah” maka akan dibentuk objek `MataKuliah` baru dengan id objek `MataKuliah` tersebut. Dilanjutkan dengan membersihkan inputan dan mengupdate data pada table.

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

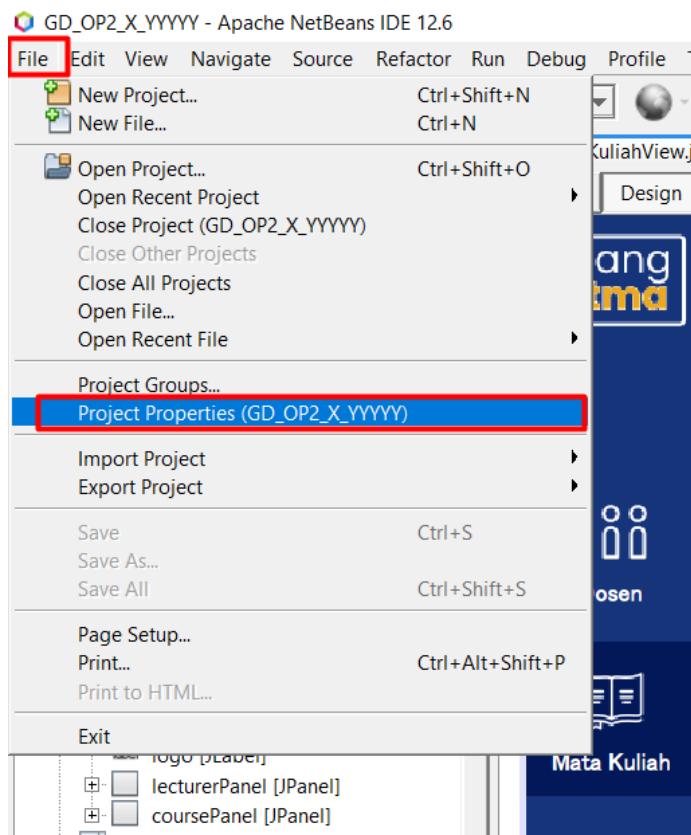
```
private void cancelBtnActionPerformed(java.awt.event.ActionEvent evt) {
    setComponent(false);
    setEditDeleteBtn(false);
    clearText();
}
```

Ketika *button cancel* di klik maka seluruh komponen input dan *button edit* dan *button delete* akan dinonaktifkan. Selain itu akan dipanggil method `clearText()` untuk membersihkan data yang ada di komponen input.

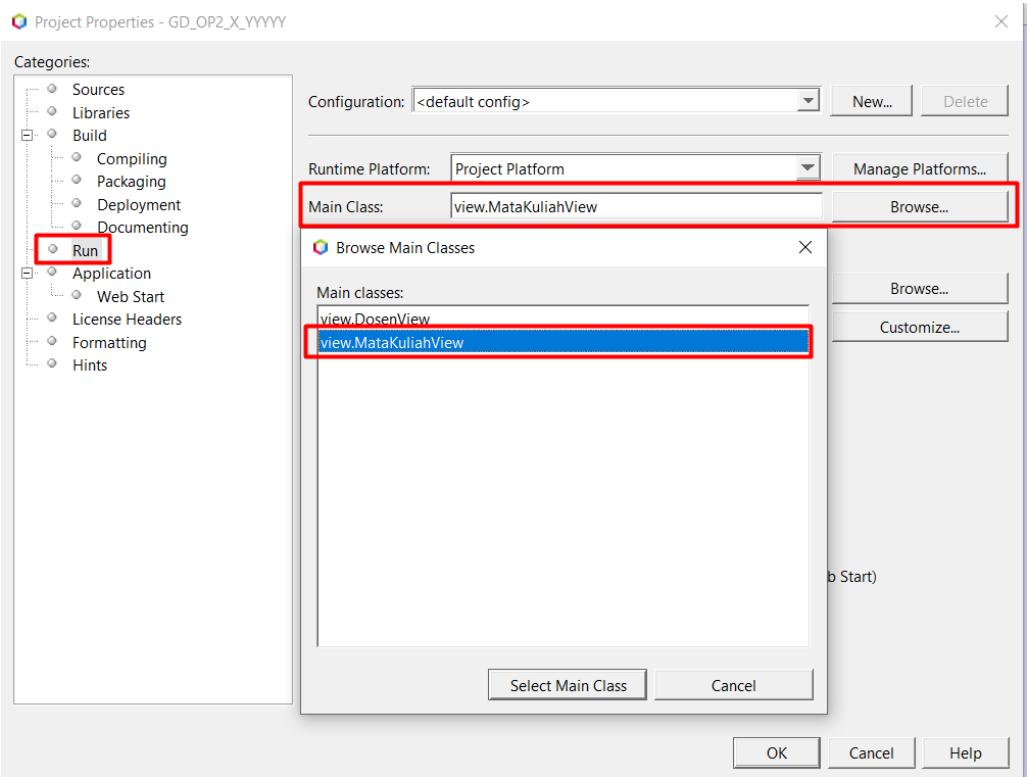
## MAIN VIEW

Pada awalnya main class tidak dibuat karena kita menggunakan view sebagai main kita. GD Object Persistence 2 merupakan lanjutan dari GD Object Persistence 1. Sehingga main view telah diset kepada view Dosen. Sehingga kita harus mengubah main view tersebut menjadi main view Mata Kuliah.

- a. Klik File di pojok kiri atas dan pilih Project Properties.

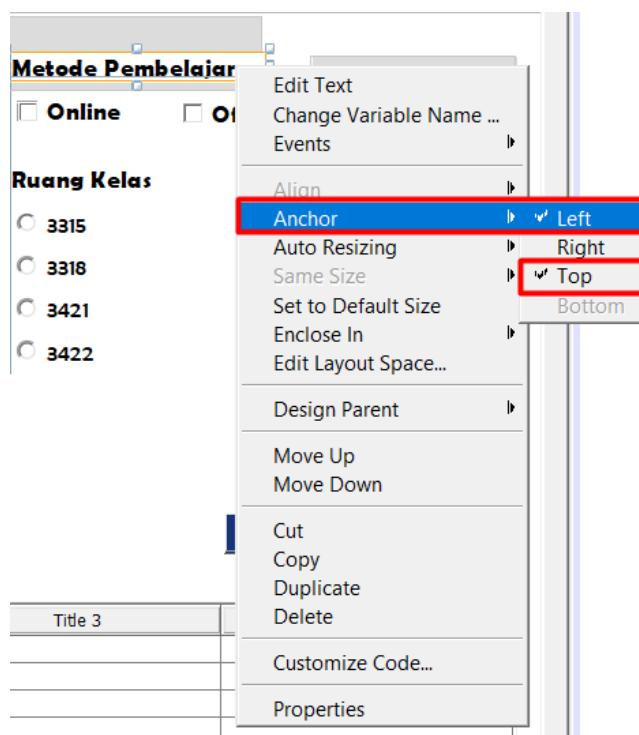


- b. Klik Run kemudian klik browse pada bagian main class dan pilih view MataKuliahView.

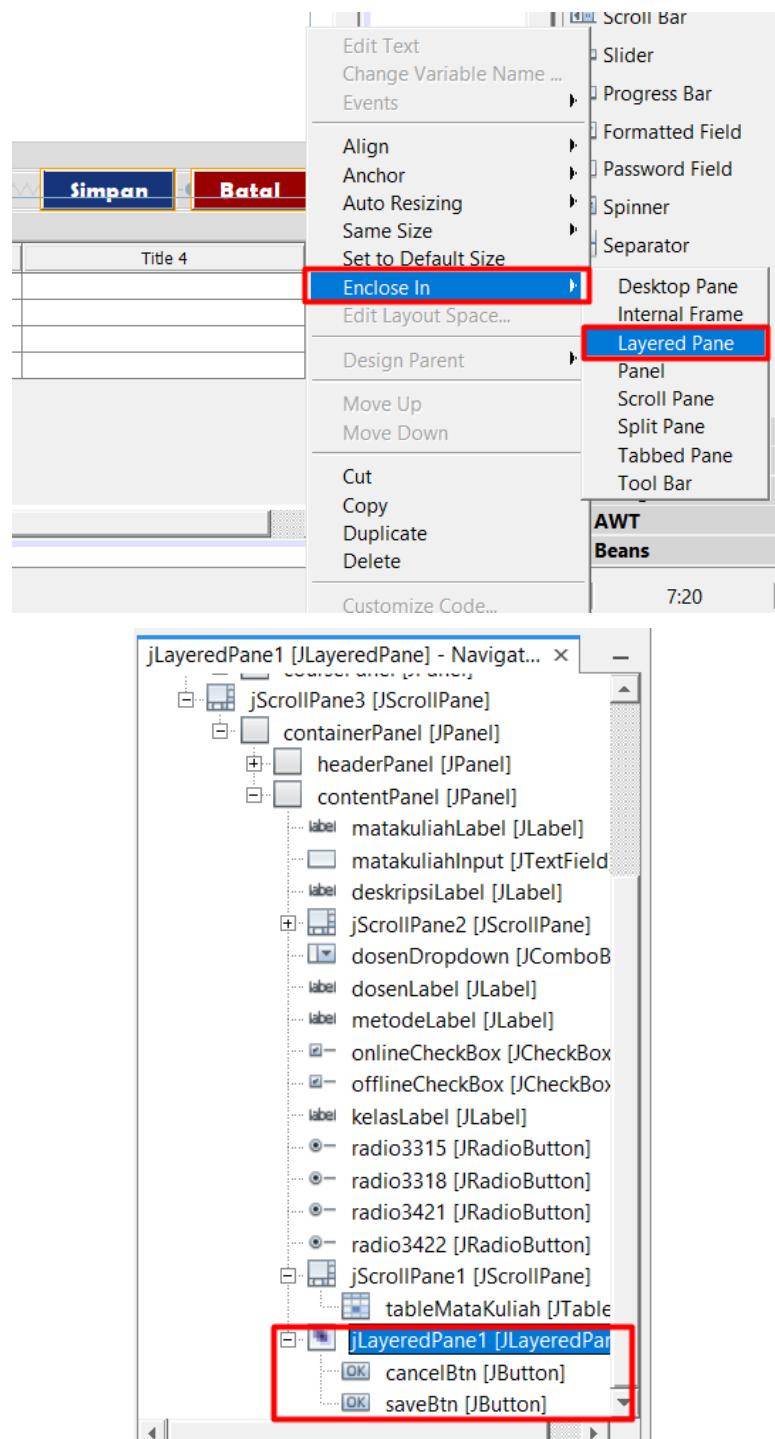


### Tambahan (Optional) :

Terkadang design yang kita buat tidak sesuai dengan arah atau peletakan titik berat yang kita mau. Untuk memperbaiki peletakan beberapa component, dapat memberikan arahan anchor kepada component dalam suatu field. Klik kanan pada suatu component(gambar dibawah ini meng-klik metodeLabel) kemudian pilih anchor, ubah menjadi left dan top. Jika kita ingin menitik beratkan suatu component agar mengikut ke kanan layer, maka anchor di set right. Perlu diingat bahwa anchor akan **berlaku untuk semua component yang ada di dalam field** tersebut.

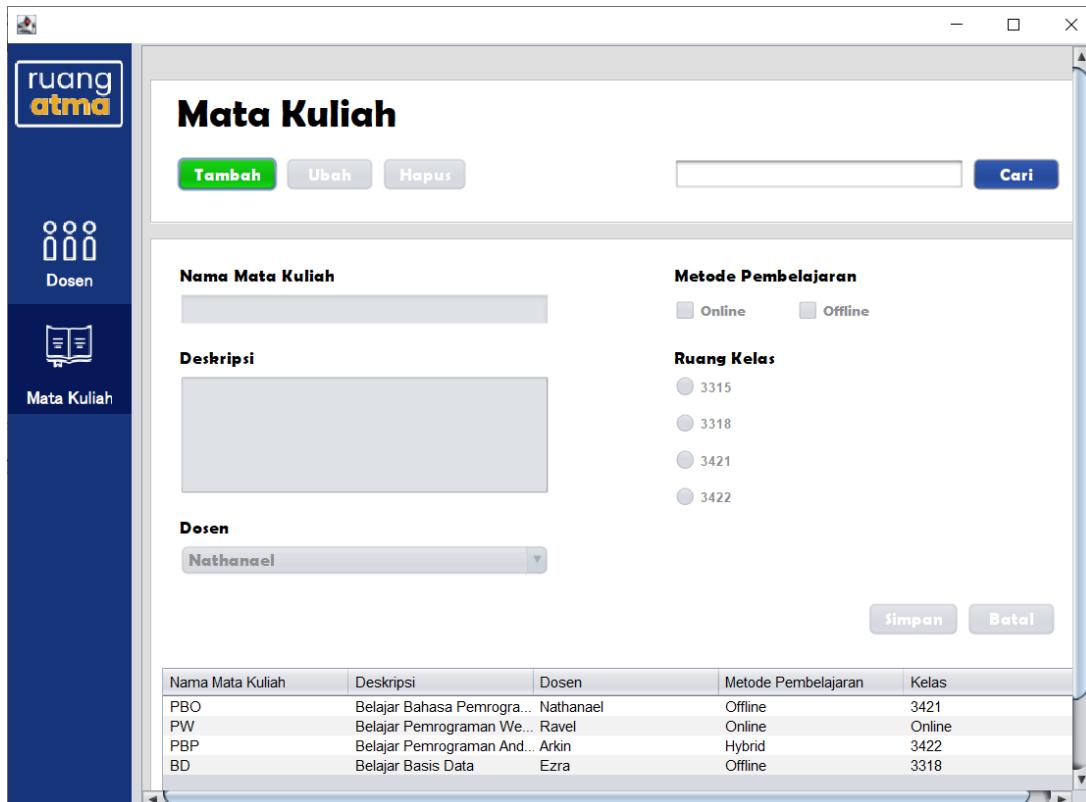


Agar beberapa component dapat berdiri sendiri terpisah dari parentnya(berdasarkan hirarki navigator) dan tidak terpengaruh dari anchor field-parentnya, gunakan Layered Pane. Select 2 component yakni `saveBtn` dan `cancelBtn` dengan cara tahan left shift pada keyboard dan klik kedua component tersebut. Lalu klik kanan dan pilih Enclose In => Layered Pane. Dengan begini, `saveBtn` dan `cancelBtn` akan berdiri sebagai layer sendiri di dalam `contentPanel`.

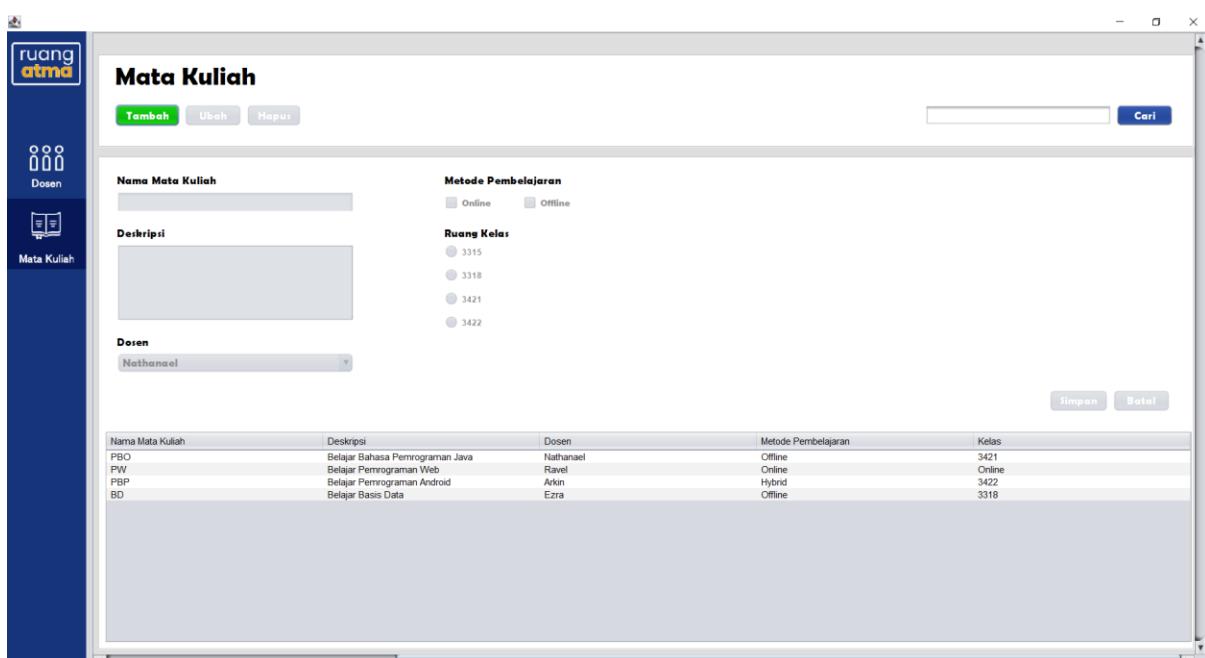


Masih ada banyak cara untuk memperindah GUI. Silakan teman-teman untuk eksplor dan mencoba sendiri. Selamat! Anda telah menyelesaikan modul kali ini. Berikut adalah hasil akhir tampilan dari aplikasi yang sudah dibuat.

Hasil Akhir (Windowed)



Hasil Akhir (Full Screen)



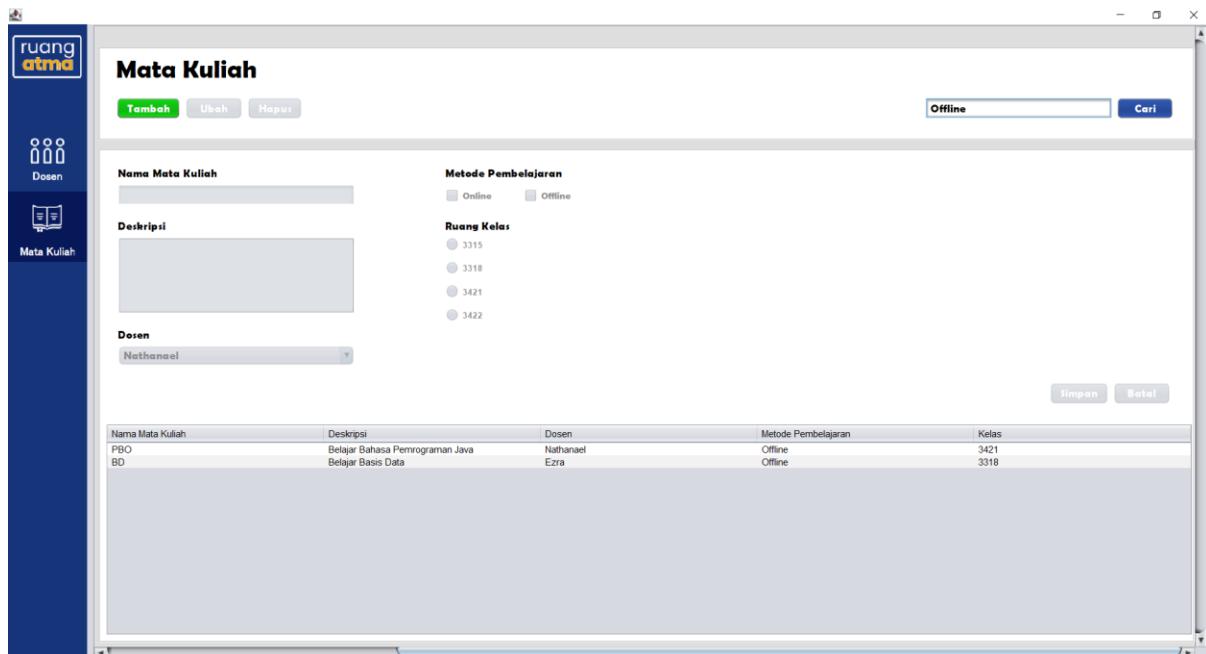
Pada modul kali ini cara untuk menambah, mengubah, menghapus dan mencari data masih mirip dengan modul sebelumnya. Hanya bedanya pada modul ini, pengguna dapat melakukan pencarian dengan kata kunci baik itu nama mata kuliah, deskripsi, nama dosen, metode pembelajaran atau kelas. Sedangkan untuk mengubah atau menghapus data pengguna dapat mengklik data pada tabel dan data yang dipilih akan terisi pada form inputan seperti pada modul sebelumnya.

Silakan mencoba Tambah, Ubah, Hapus, Cari, Exception, Klik Data Tabel, Simpan, dan Batal. Jangan lupa juga coba klik lecturerPanel untuk pindah ke Dosen view.

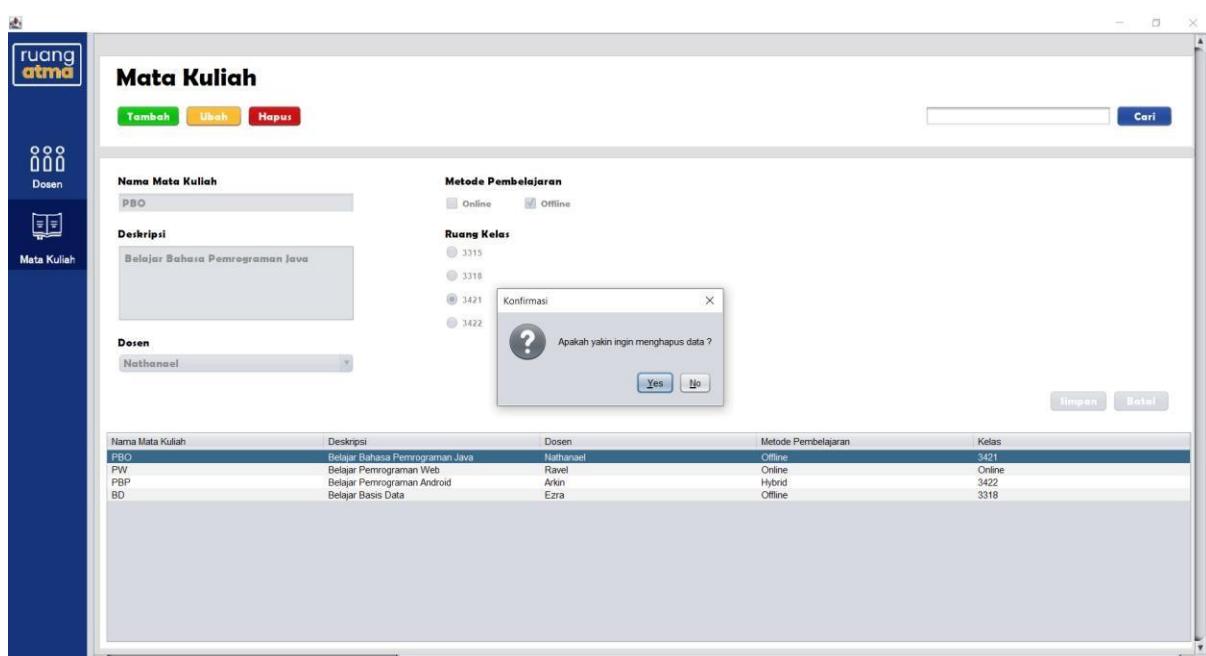
Tampilan saat Klik Data pada Table.

Nama Mata Kuliah	Deskripsi	Dosen	Metode Pembelajaran	Kelas
PBO	Belajar Bahasa Pemrograman Java	Nathanael	Offline	3421
PW	Belajar Pemrograman Web	Ravel	Online	3422
PBP	Belajar Pemrograman Android	Arkin	Hybrid	3318
BD	Belajar Basis Data	Ezra	Offline	

Tampilan saat Search suatu kata (kata yang dicari = “Offline”)



Tampilan saat muncul JOptionPane ketika akan melakukan Delete



Tampilan Exception ketika inputan kosong

