**MODUL 9**

**OBJECT PERSISTENCE II**

Pertemuan lalu kita sudah pelajari **Object Persistence I**. Di OP1, kita sudah pelajari dasar-dasar object persistence. Di pertemuan ini kita akan belajar cara membuat relasi antara satu kelas dan kelas lain di database, cara menggunakan prepared statement di DAO, cara menggunakan JTable untuk menampilkan data menggunakan tabel, dan styling interface.

**Modifikasi Database**

**Penambahan Model Departemen**

Buatlah sebuah table baru pada dengan nama **departemen** pada database ksp\_xxxx dengan ketentuan sebagai berikut

Graphical user interface, application

Description automatically generated

**Membuat DAO Departemen**

Kita akan membuat DAO yang lengkap (CRUD (Create, Read/Search, Update, dan Delete)) untuk model Departemen yang telah kita buatkan.

Yang beda dari DAO kali ini adalah: **kita akan mencoba menggunakan *prepared statement***. *Prepared statement* sebenarnya merupakan **cara paling ampuh untuk mencegah *SQL Injection*** (adanya injeksi (penambahan) berbagai *statement* (perintah) SQL yang bisa berbahaya, karena value yang dimasukkan tidak lagi disaring, tapi langsung saja dijalankan).

Perbedaan antara statement biasa dan *prepared statement*:

Text

Description automatically generated

Prepared statement **wajib digunakan** kalau kita mau membuat aplikasi besar yang akan digunakan banyak user. Lebih lengkap tentang *SQL injection* dan *prepared statement* bisa dibaca di:

* <https://www.dqlab.id/kenali-sql-injection-dan-pencegahannya-sebelum-terlambat>
* <https://medium.com/pujanggateknologi/prepared-statement-di-go-927b1a8863ec#36e0>

Prepared statement dan statement biasa memiliki beberapa perbedaan, antara lain:

|  |  |  |
| --- | --- | --- |
| **Pembeda** | **Statement Biasa** | **Prepared Statement** |
| Penulisan *query SQL* | Terlihat kurang rapi karena sedikit-sedikit musti “digabungin” sama nama, jam kerja, dsb. | Terlihat lebih rapi, kita beri tanda saja bagian mana yang mau diisi dengan tanda tanya (?) |
| Pembuatan statement (*create statement*) | Kita menggunakan class “Statement” dan “CON.createStatement()” | Kita menggunakan class “PreparedStatement” dan “CON.prepareStatement()”, kemudian masukkan query apa yang mau diformat di dalam “CON.prepareStatement()” |
| Pengisian parameter | Sudah kita lakukan langsung di “String sql” di atas, walau agak susah dibaca. | Kita menggunakan “statement.setString(…)” (dan sejenisnya) untuk mengisi tanda tanya di “String sql”.  Parameter 1 menandakan “*tanda tanya keberapa yang ingin diganti*” dan parameter 2 menandakan “*apa yang mau diisi di situ*” |
| Mendapatkan hasil | Untuk mendapatkan result, kita memanggil “statement.executeUpdate/executeQuery()”, kemudian masukkan query yang ingin dijalankan sebagai parameter. | Karena query sudah diformat, langsung saja kita jalankan “statement.executeUpdate/executeQuery()”. |

Sekarang kita akan membuatkan DAO untuk model Departemen dengan memanfaatkan *prepared statement* yang ada di JDBC.

Buatkan class **DepartemenDAO** di dalam **package dao**, tambahkan method makeConnection() dan method closeConnection() seperti di StaffDAO.

Text

Description automatically generated

Text

Description automatically generated

Kita akan mulai membuat berbagai method untuk mengakses database, antara lain **insert** (menambah departemen)**, get** (mendapatkan data departemen dari kodenya)**, search** (mencari departemen)**, update** (mengupdate departemen)**,** dan **delete** (menghapus departemen).

*Insert:*

Text

Description automatically generated

*Get:*

Text

Description automatically generated

*Search:*

*Text

Description automatically generated*

*Update:*

Text

Description automatically generated

*Delete:*

*Text

Description automatically generated*

Kemudian jangan lupa tutup class dengan kurung kurawal :D



Perhatikanlah apa saja perbedaan antara DAO yang menggunakan prepared statement dan DAO yang tidak menggunakannya!

***Catatan:*** *metode DAO yang menggunakan prepared statement* ***hanya diajarkan di KSP Java****. Di kelas praktikum sesungguhnya, teman-teman masih akan menggunakan DAO dengan statement biasa (seperti yang diajarkan di OP1).*

**Membuat Table Departemen**

Di guided OP2, kita akan menggunakan table untuk menampilkan data departemen ke user. Karena itu kita musti membuat Class TableDepartemen.

Buatlah sebuah **package baru** dan beri nama **table**.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Setelah itu, buatkan **class baru** di dalam **package table** bernama **TableDepartemen**, isikan class tersebut dengan kode berikut untuk constructor:

Text

Description automatically generated

Kemudian, karena table yang kita buatkan merupakan **extends AbstractTableModel**, berarti ada beberapa method yang harus kita override.

Shape

Description automatically generated

Kedua method di atas merupakan method yang digunakan untuk mendapatkan **jumlah baris** (jumlah item di list) **dan jumlah kolom** (jumlah atribut di model Departemen).

Text

Description automatically generated

Method di atas merupakan method yang digunakan untuk mendapat “data apa yang akan ditampilkan di kolom ke-***x***”. Jadi di sini kita ingin menampilkan kode departemen di index ke-0 (berarti kolom pertama), nama di index 1 (kolom kedua), dan seterusnya.

Graphical user interface, text

Description automatically generated

Method di atas merupakan method yang digunakan untuk menampilkan judul dari setiap kolom.



Kemudian tutup class TableDepartemen dengan kurung kurawal.

**Membuat Controller Departemen**

Setelah kita membuatkan DAO, kita akan membuatkan controllernya. Di controller ini kita akan membuatkan beberapa method, yakni **insertDataDepartemen** (tambah departemen)**, getByKode** (mencari data departemen dari kodenya)**, searchDataDepartemen** (searching departemen, return List)**, showDataDepartemen** (searching departemen, return TableDepartemen)**, updateDataDepartemen** (update departemen)**,** dan **deleteDataDepartemen** (menghapus departemen).

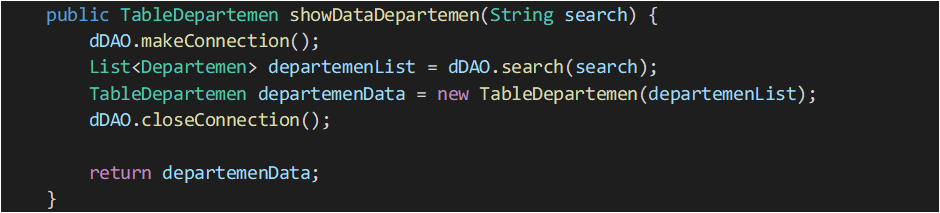
Buatlah sebuah **class baru** di dalam **package control** dengan nama **DepartemenControl**. Isikan class tersebut dengan kode:

Text

Description automatically generated

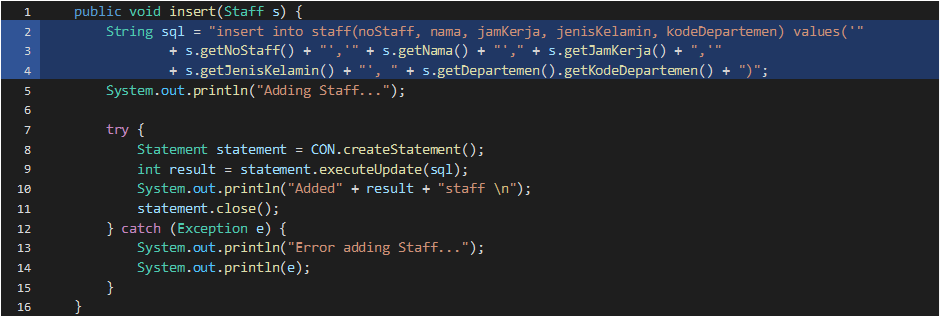
Dari class DepartemenControl di atas, ada beberapa method yang cukup berbeda dengan method-method yang dipakai di OP1. Ada **method getByKode** yang sebenarnya tidak kita gunakan di dalam guided ini, tetapi kan teman-teman sudah terlanjur copas ^\_^. Method ini sebenarnya bisa kita gunakan untuk **mendapat data lengkap departemen seorang staff**.

Ada juga method **searchDataDepartemen** dan **showDataDepartemen**. Sekilas terdengar sama, tetapi **nilai balikan** dari kedua method ini berbeda. Method searchDataDepartemen me-return List, sedangkan showDataDepartemen me-return TableDepartemen. *Mengapa perlu ada searchDataDepartemen?* Karena kita akan menggunakan List yang di-return searchDataDepartemen untuk menampilkan data di TableDepartemen yang ada di method showDataDepartemen. Sebenarnya bisa saja kita langsung “gabungkan” kedua method ini seperti ss di bawah, tetapi bisa kelihatan kurang elegan.



**Modifikasi StaffDAO**

Karena kita sudah menambah relasi antara Departemen dan Staff (1 staff memiliki 1 departemen), sehingga perlu kita update juga class StaffDAO pada beberapa bagian. Perubahan di class StaffDAO diberi warna background biru tua pada method-method di bawah ini.

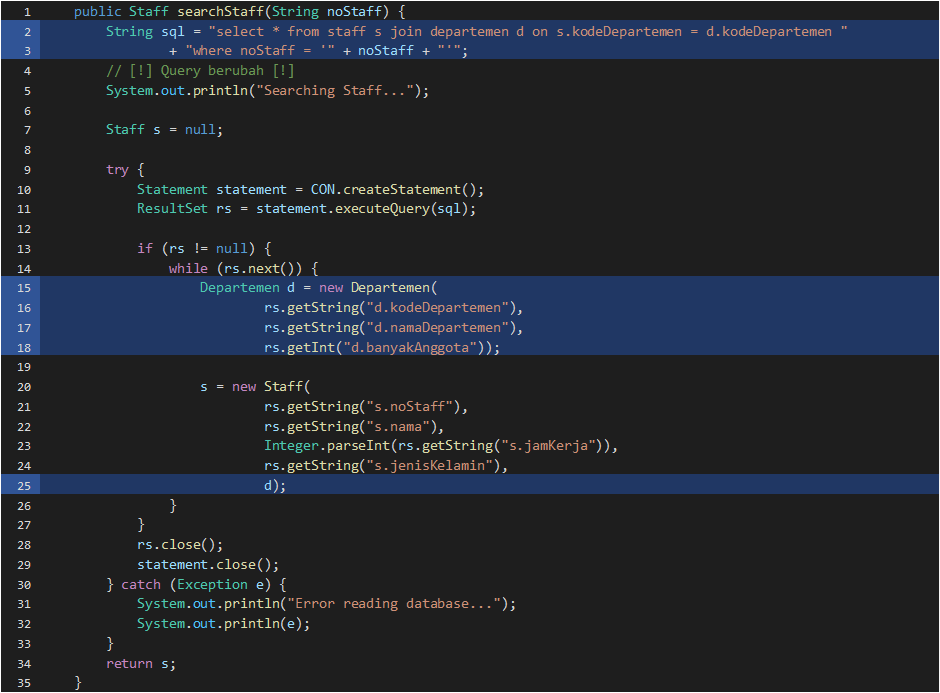


Pada method insert di atas, kita juga harus meng-insert kode departemen staff tersebut, jadi query SQL kita modifikasi untuk menambahkannya (lihat line 2-4).

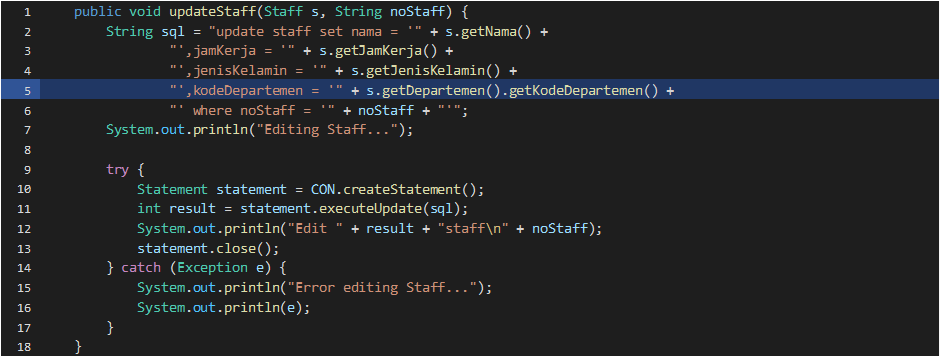
Text

Description automatically generated

Pada method show di atas, kita ingin **menampilkan staff dan departemennya**, sehingga pada query SQL (line 2), kita tambahkan keyword “JOIN” *(ingat-ingat kembali yang dipelajari di Basis Data)* untuk menggabungkan table staff dan departemen. Karena kita sudah mendapat juga data departemen, perlu kita tambahkan juga departemen tersebut ke dalam objek Staff yang akan dibuat. Caranya bisa dilihat di line 13-16. Kemudian karena constructor Staff perlu diisi dengan departemen, kita juga mengisinya dengan cara yang ada di line 23.

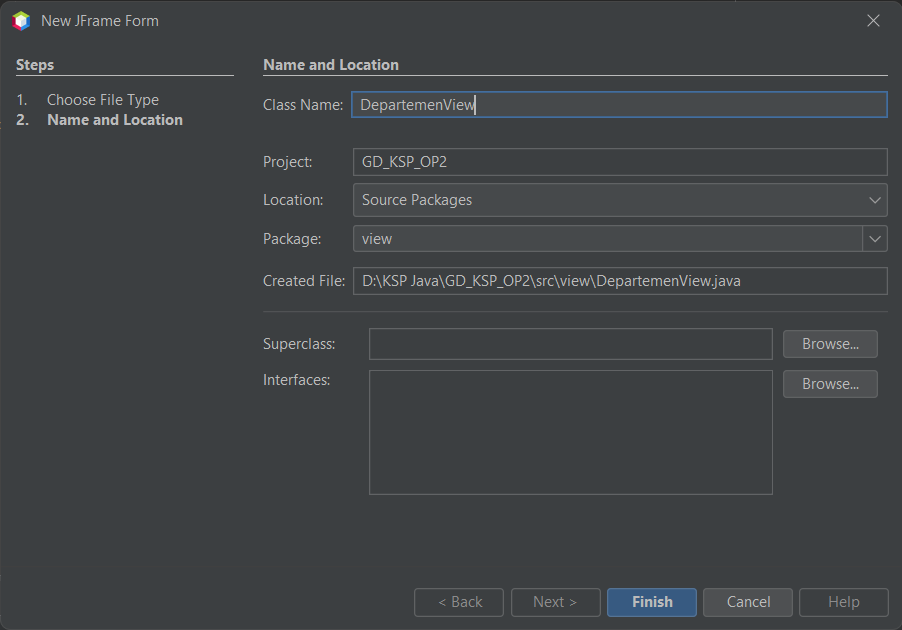


Sama seperti method show yang sudah kita modifikasi, method searchStaff juga perlu kita bah untuk bisa menampung data departemen dari staff yang dicari.



Di method updateStaff, kita juga perlu mengupdate kode departemen dari staff tersebut, jadi query SQL kita ubah (lihat line 2-6 (khususnya 5)).

Method deleteStaff tidak ada perubahan karena tugasnya hanya menghapus staff dan tidak melibatkan departemen.

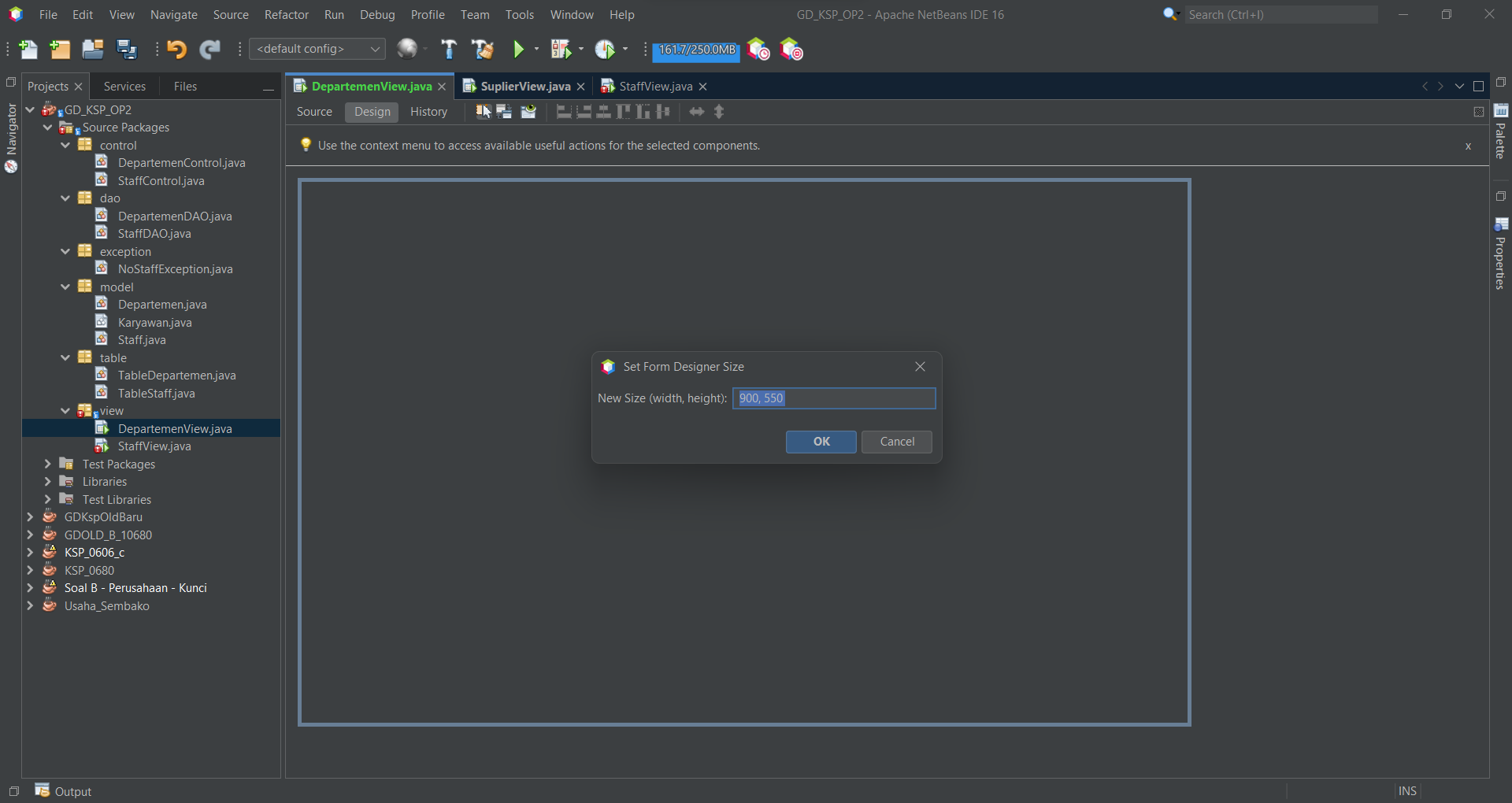
**Membuat Departemen View**

Buatlah sebuah kelas baru menggunakan **JFrame Form** pada kelas view dengan nama DepartemenView

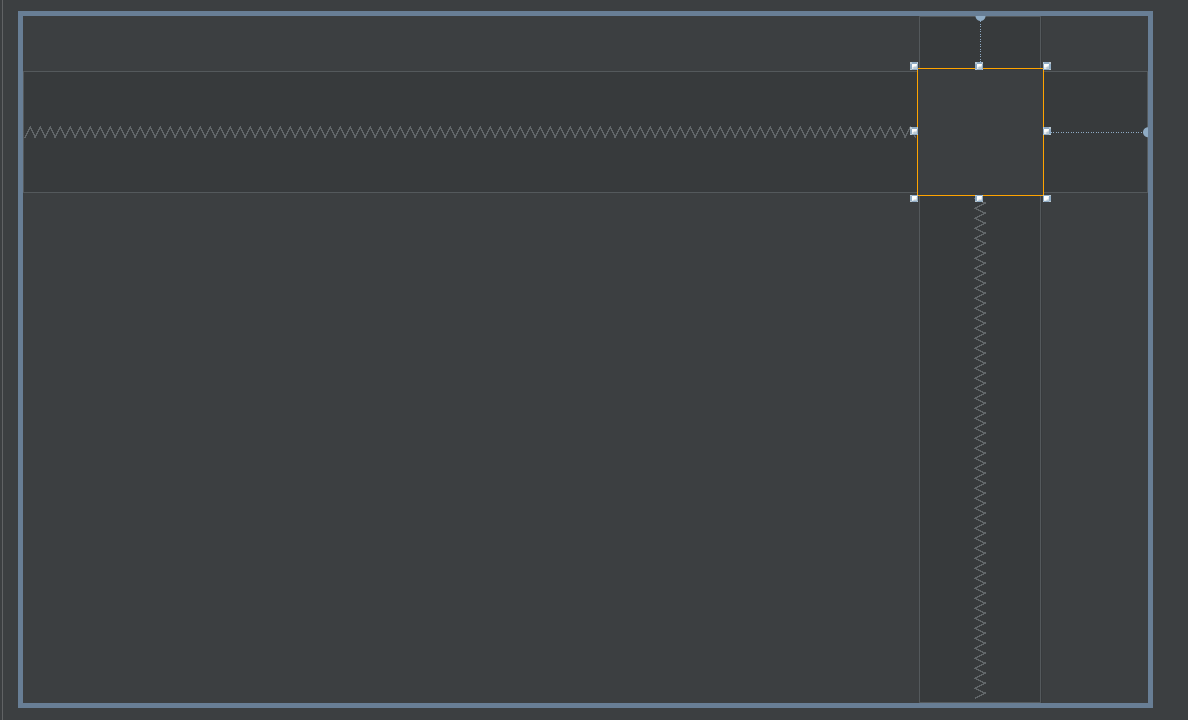
Graphical user interface, application

Description automatically generated

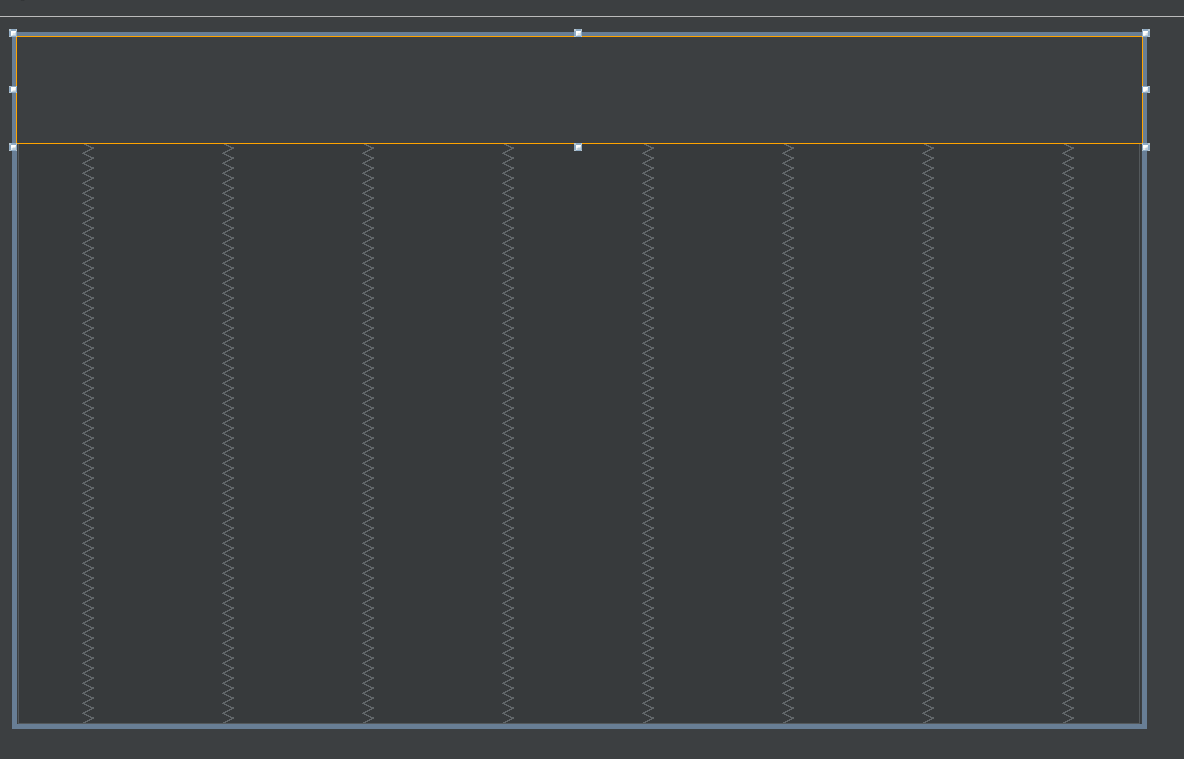
Buatlah ukurannya menjadi 900 x 550 dengan mengeklik 2 kali pada bordernya sehingga akan muncul seperti dibawah ini



Drag Panel dari pallete, kemudian ubah ukurannya menjadi



kemudian ubah ukurannya menjadi 900 x 87 dan taruh dibagian atas jFrame seperti ini



Lalu klik kanan pada panel untuk merename panel tersebut

Graphical user interface

Description automatically generated

Ubah Namanya menjadi navigationPanel.

Graphical user interface

Description automatically generated

Graphical user interface

Description automatically generated

Lalu klik kanan lagi pada panel untuk masuk ke menu properties pada panel

Ubah background menjadi 20,54,29

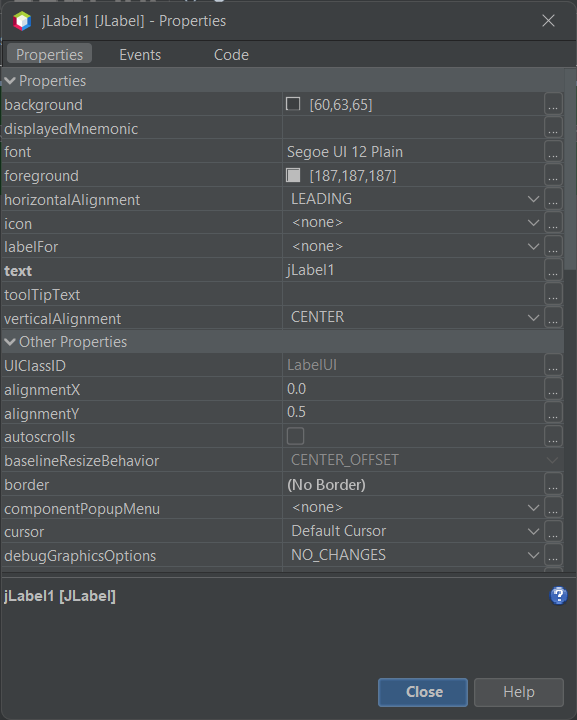
Graphical user interface

Description automatically generated

Taruh jLabel seperti bada gambar dibawah ini, lalu kemudian klik kanan untuk masuk ke menu properties   
Graphical user interface

Description automatically generated with low confidence

Disini kita akan menambahkan logo pada jLabel dengan cara mengeklik titik 3 yang ada pada bagian kanan icon



Lalu sesuaikan seperti berikut, jadi untuk icon diambil dari asset yang sudah kita tambakan diawal tadi.  
Graphical user interface, application

Description automatically generated

Setelah itu sesuaikan gambar atau icon menjadi seperti pada gambar dibawah ini

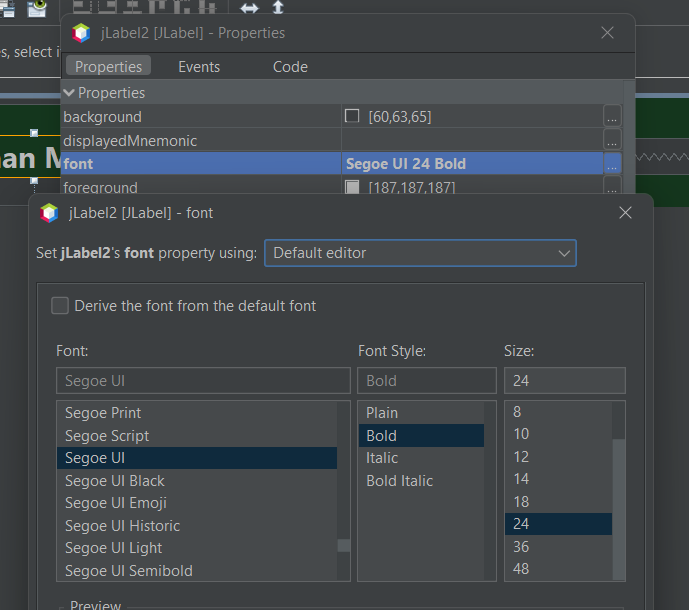
Graphical user interface

Description automatically generated

Lalu tambah jLabel disampin icon yang sudah ditambahkan tadi, dan edit text menjadi seperti dbawah ini  
Graphical user interface, text, application

Description automatically generated

Agar judul dapat dibaca dengan jelas, ubah ukuran font menjadi 24 dan font style menjadi Bold



Sehingga menjadi seperti ini

Graphical user interface, text

Description automatically generated

Selanjutnya kita akan membuat 2 buah button menggunakan panel dan label tanpa menggunakan jButton.

Pertama buat sebuah panel lalu set ukurannya menjadi 140 x 37 dan beri warna background [205,154,73], dan tata pada navigation panel seperti pada gambar dibawah ini

Timeline

Description automatically generated

Setelah itu buat jLabel dengan text Karyawan dengan setingan seperti berikut

Graphical user interface, text

Description automatically generated

Lalu letakkan ditengah tengah panel yang sudah dibuat sebelumnya, menjadi seperti ini.

Graphical user interface, website

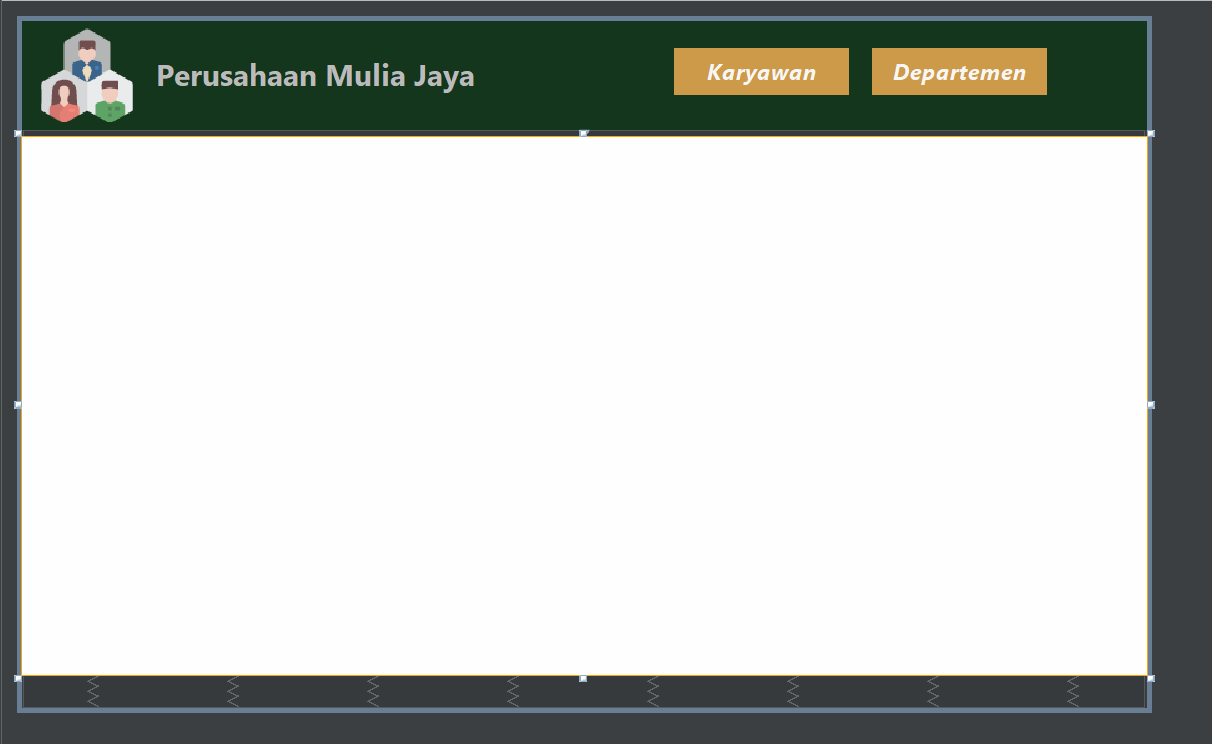
Description automatically generated

Lakukan hal yang sama untuk membuat button Departemen, sehingga menjadi seperti gambar dibawah ini

Graphical user interface

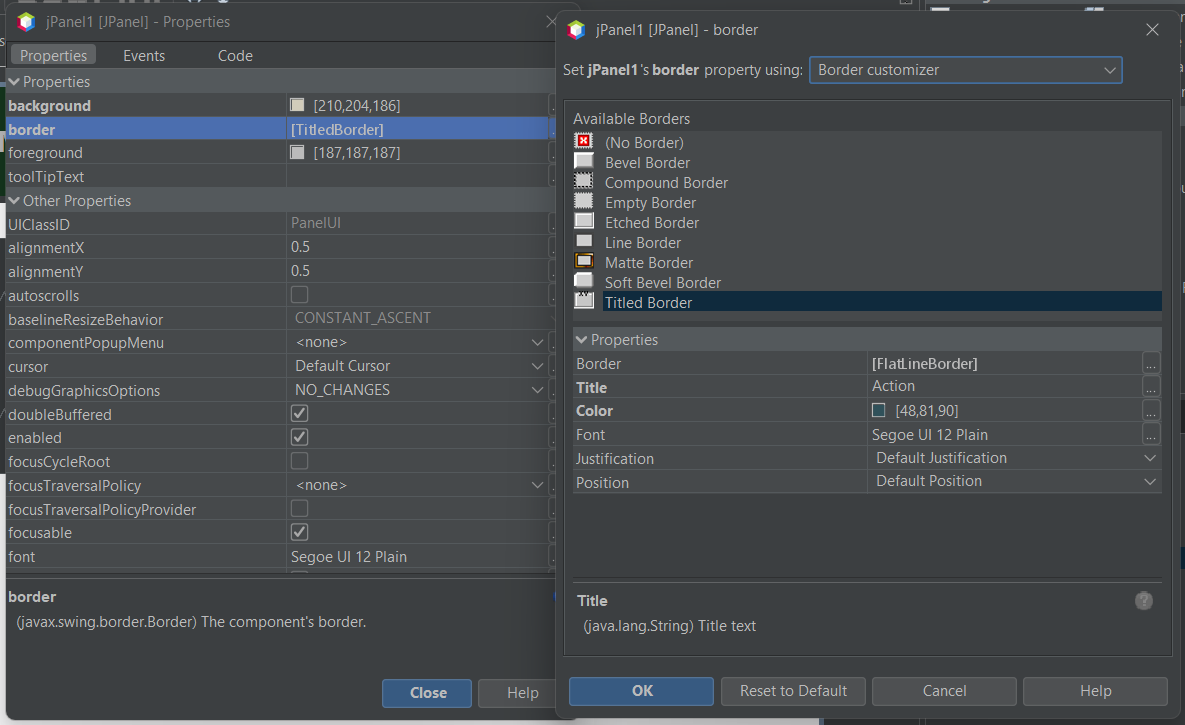
Description automatically generated

Selanjutnya, buatlah sebuah panel untuk menempatkan fungsi-fungsi Utama pada departemen view dengan ukuran 900 x 425, dengan background color [254,254,255] dan ditaruh di bawah navigation bar seperti berikut, beri nama panel tersebut contentPanel

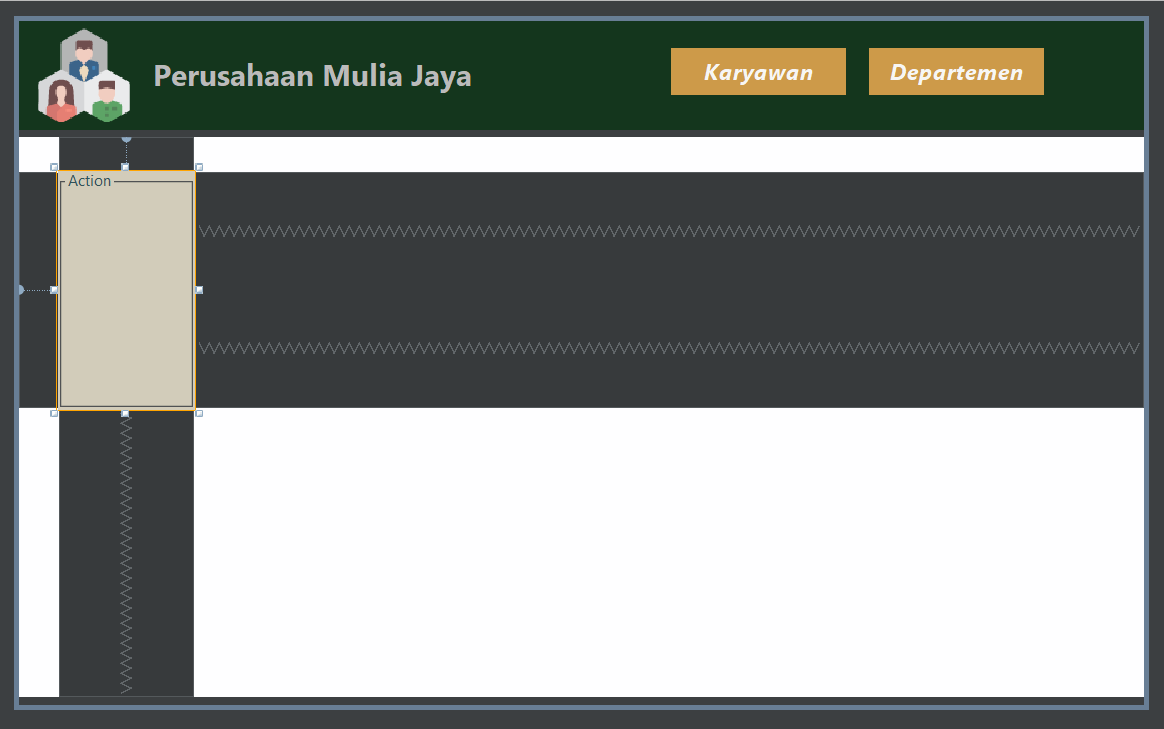


Tambah Sebuah panel pada contentPanel dengan ketentuan seperti berikut:

* [210,204,186]
* Nama variabel : actionPanel
* Background : [210,204,186]
* Border: TitledBorder
* Title: Action
* Title Color: [48,81,90]



Lalu tata panel aksi seperti gambar dibawah ini.

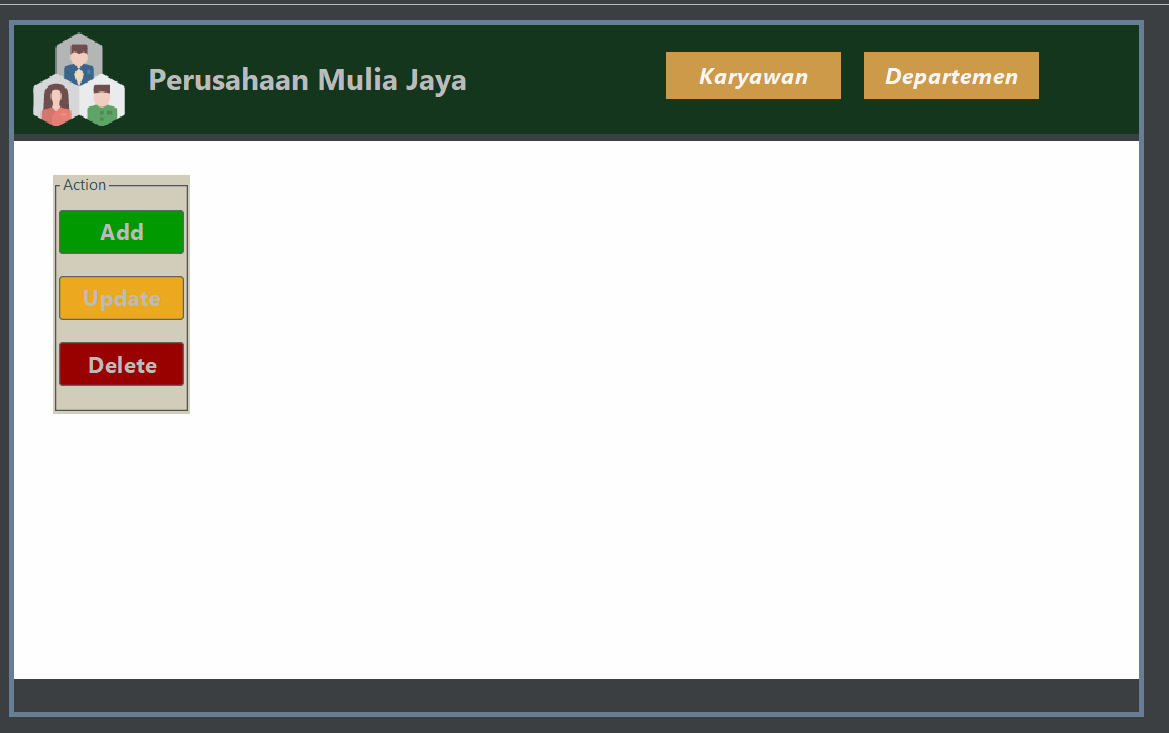


Selanjutnya kita akan menambahkan 3 button pada panel action, dengan drag button pada panel action, dengan pengaturan seperti berikut

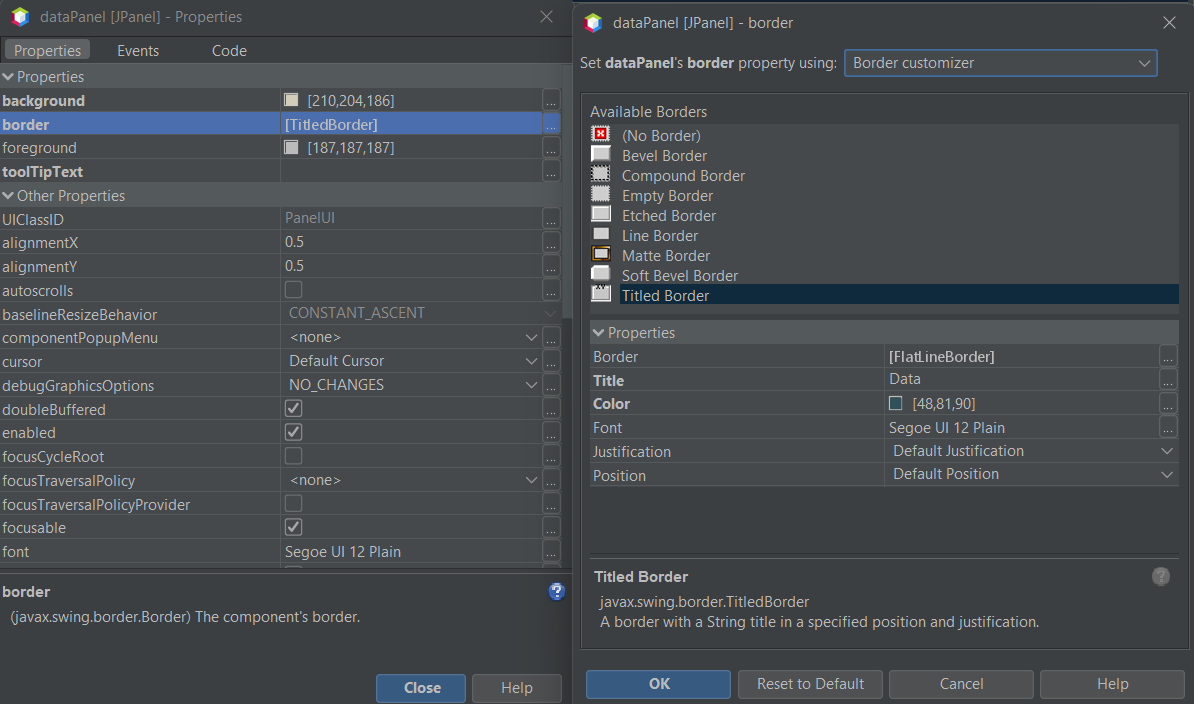
Ukuran Btn: 100 x 35 untuk setiap Button

1. Btn 1
   * Nama variabel : addBtn
   * Text : Add
   * Background : 0,153,0
   * Foreground: [247,246,244]
   * Font : - Bold – 18
2. Btn 2
   * Nama variabel : updateBtn
   * Text : Update
   * Background : 236,168,30
   * Foreground: [247,246,244]
   * Font : - Bold – 18
3. Btn 3
   * Nama variabel : deleteBtn
   * Text : Delete
   * Background : 153,0,0
   * Foreground: [247,246,244]
   * Font : - Bold – 18

Lalu tata Button yang sudah dibuat tadi dengan tatanan seperti berikut



Selanjutnya buat sebuah dataPanel dengan ukuran 265 x 270 dengan setingan sama denan actionPanel sebelumnya



Lalu taruh disebelah actionPanel seperti berikut



Selanjutnya akan dibuat untuk input data pada dataPanel, dengan beberapa bagian seperti berikut

1. Input Kode Departemen
   * Komponen : Text Field dan Label Nama variabel
   * Label : kodeDepartemenLabel
   * Font Label & Field: Bold- 14
   * Font foreground: [48,81,90]
   * Nama variabel Input : kodeDepartemen Input
2. Input nama Departemen
   * Komponen : Text Field dan Label Nama variabel
   * Label : namaDepartemenLabel
   * Font Label & Field: Bold- 14
   * Font foreground: [48,81,90]
   * Nama variabel Input : namaDepartemenInput
3. Input banyak Anggota
   * Komponen : Text Field dan Label Nama variabel
   * Label : banyakAnggotaLabel
   * Font Label & Field: Bold- 14
   * Font foreground: [48,81,90]
   * Nama variabel Input : banyakAnggotaInput

Lalu tata Seperti pada gambar dibawah ini



Selanjutnya buatlah sebuah panel lagi dengan ukuran 445 x 60 dengan konfigurasi properties seperti panel panel sebelumnya

* 1. Panel
     + Ukuran 445 x 60
     + Variable name : searchPanel
     + Background: [210,204,186]
     + Border: titled border
     + Border title: Search
     + Border Title color: [48,81,90]
  2. Input & Button Search
     + - Komponen : Text Field dan Button Nama variabel
       - Nama variabel Input : seachDepartemenInput
       - Button Text: Search
       - Button Variable name: searchBtn
       - Button Background Color: [51,51,255]
       - Button foreground color: [247,246,244]
       - Button font Size: Bold -14

Lalu tata menjadi seperti ini



Sebelum masuk untuk membuat tabel, selanjutnya buat 2 buah button untuk save dan cancel pada Data panel dengan ketentuan seperti berikut:

* + - * 1. Button cancel
    - Nama variabel : cancelBtn
    - Text : Cancel
    - Background : [153,0,0]
    - Foreground: [247,246,244]
    - Font : - Bold – 14
      * 1. Button Save
    - Nama variabel : saveBtn
    - Text : Save
    - Background : [0,153,0]
    - Foreground: [247,246,244]
    - Font : - Bold – 14

Lalu tata menjadi seperti ini



Selanjutnya kita akan menaruh table dibawah searchpanel dengan menggunakan table dari pallete

* Table Variable name: tableDepartemen
* Ukuran : menyesuaikan tempat seperti contoh gambar dibawah

Graphical user interface

Description automatically generated

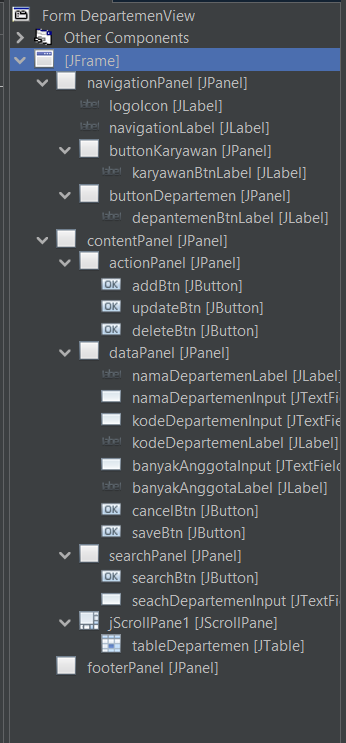
Selanjutnya kita akan membuat sebuah footer dengan panel yang ditaruh dibawah content panel dengan ukuran 900 x 27 dengan background color [205,154,73], sehingga jika ditata akan menjadi seperti ini



Untuk dapat berpindah halaman ke gui karyawan, klik kanan pada KaryawanBtnLabel lalu pilih seperti pada gambar dibawah ini.

Graphical user interface

Description automatically generated

Jika semua komponen sudah tertata, maka akan terlihat seperti ini pada navigation menu.  


Ketiikan code ini pada bagian seperti dibawah

**Text

Description automatically generated**

Setelah selesai merapikan untuk bagian UI, selanjutnya akan masukk e bagian coding.

Import kelas kelas yang dibutuhkan dalam DepartemenView

Text

Description automatically generated

Lalu tambah code seperti ini pada public class DepartemenView



Text

Description automatically generated

Untuk mendambahkan kodingan pada add data, double klik pada button **Add**Graphical user interface, application

Description automatically generated

Lalu tambahkan code berikut…

Text

Description automatically generated

Untuk edit button, juga lakukan double click pada **Edit** button

Graphical user interface, application, website

Description automatically generated

Untuk delete button, juga lakukan double click pada **Delete** button

Text

Description automatically generated

Untuk save button, juga lakukan double click pada **Save** button

Text

Description automatically generated

Untuk cancel button, juga lakukan double click pada **Cancel** button

Graphical user interface, text

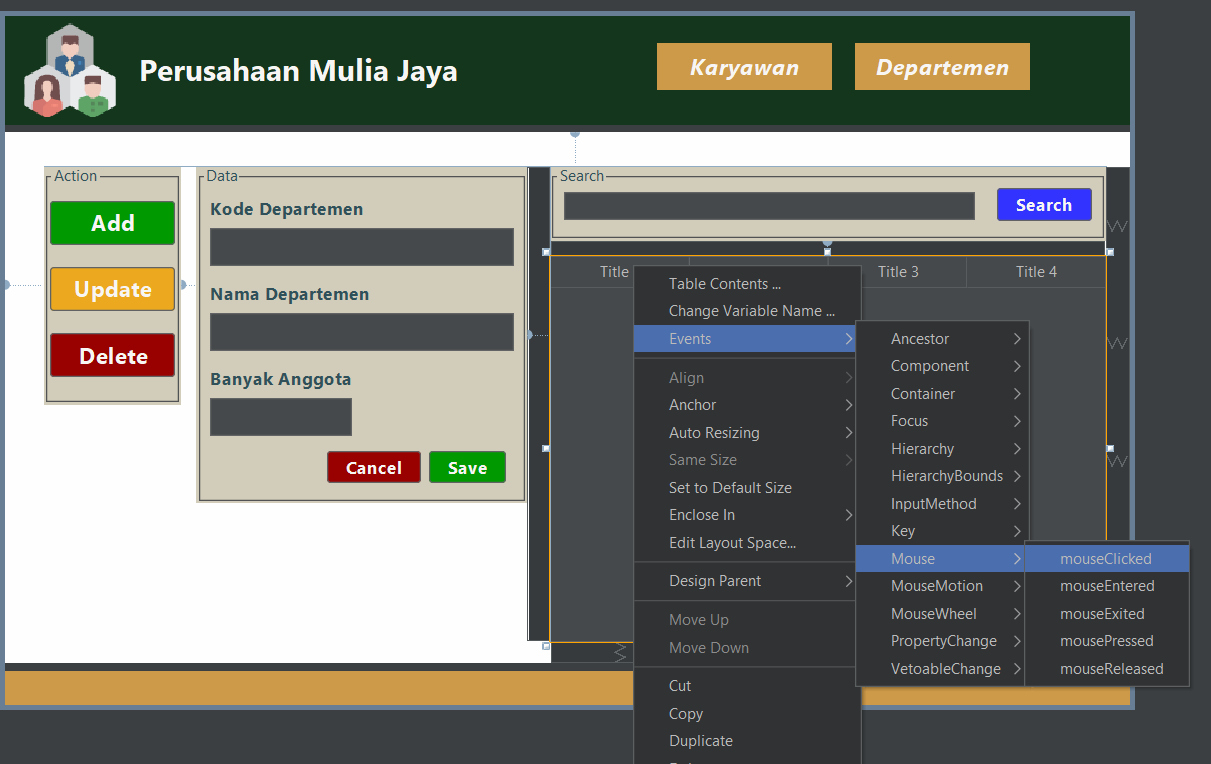
Description automatically generated

Untuk search button, juga lakukan double click pada **Search** button

Text

Description automatically generated

Agar tabel yang sudah dibuat dapat digunakan, klik kanan dan pilih seperti gambar dibawah ini.



Jika diklick maka akan pindah ke bagian coding, lalu isikan seperti dibawah ini.

Text

Description automatically generated

Selanjutnya klik kanan pada buttonKaryawan lalu pilih sesuai dengan gambar dibawah, hal ini bertujuan agar bisa mengarahkan ke halaman staff ketika kita mengeklik panel tersebut

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, text

Description automatically generated

