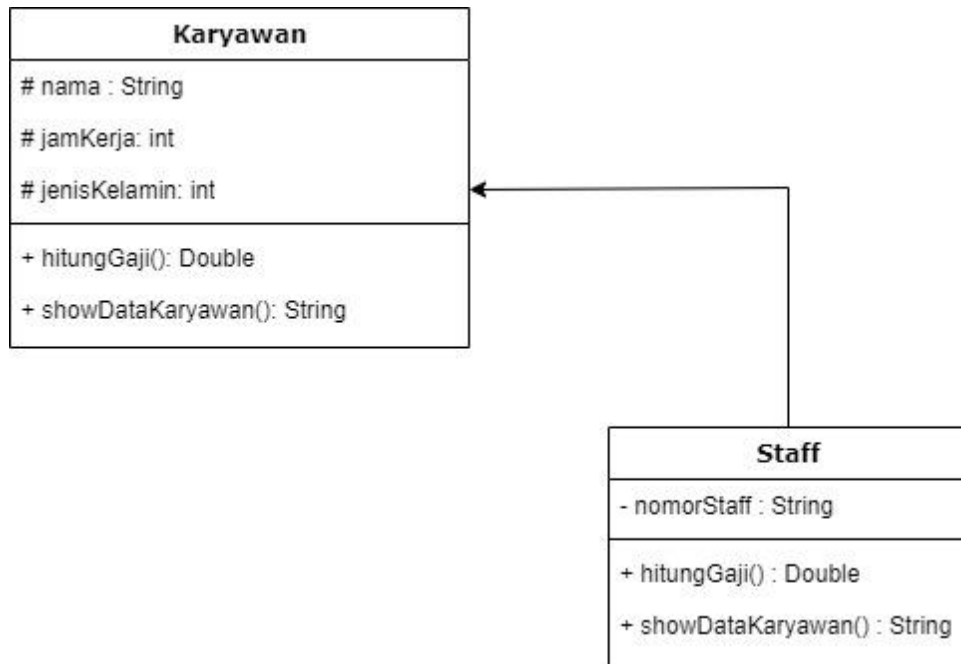


TEORI

A. OBJECT PERSISTENCE

Objek dapat menyimpan status dalam media penyimpanan, berdasarkan media penyimpanannya objek dibedakan menjadi transient objek dan persistent objek. Transient objek adalah objek yang menyimpan status dalam penyimpanan primer (memori), dimana status objek akan hilang ketika program di terminasi. Objek persistence adalah objek yang menyimpan status dalam penyimpanan sekunder (file, basis data). Keunggulan dari objek persistence adalah data atau status yang telah disimpan dalam media penyimpanan (storage), dapat diambil (restored) ketika aplikasi dijalankan kembali. Mekanisme dari objek persistence sangat sederhana, yaitu memberikan sebuah program kemampuan untuk membaca dan menuliskan keseluruhan objek ke dan dari stream byte.

GUIDED

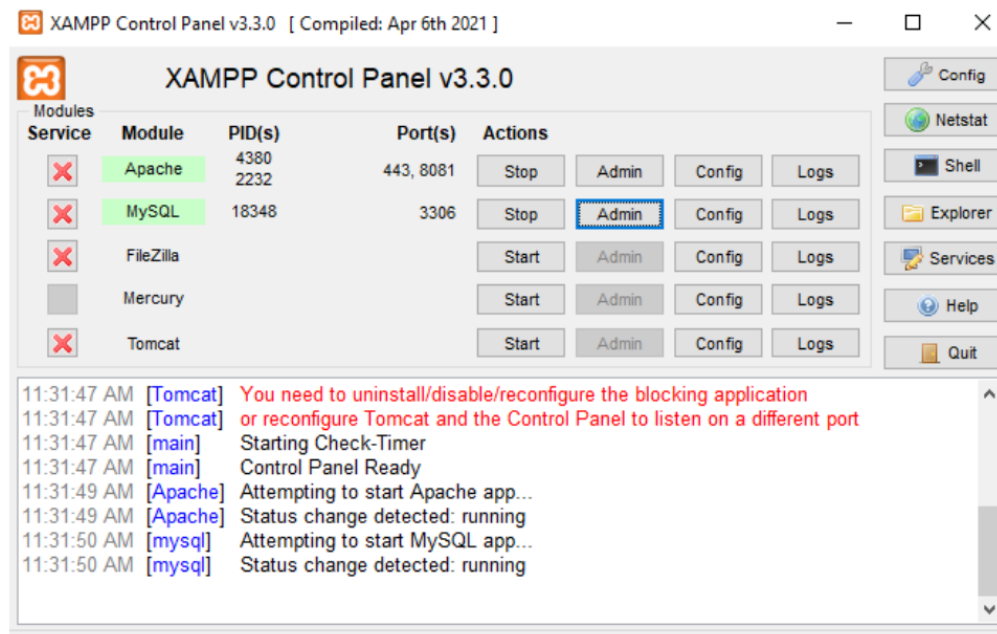


Suatu perusahaan ingin membuat sebuah sistem yang mengolah data Karyawan yang bekerja pada perusahaan tersebut. Data karyawan yang ingin dikelola adalah nama(String), jamKerja(int), jenisKelamin(String). Kelas karyawan merupakan kelas abstrak yang memiliki 2 **method abstract**, yakni hitungGaji() dan showDataKaryawan(). Lalu terdapat juga kelas Staff yang merupakan turunan dari kelas Staff dimana memiliki atribut tambahan noStaff (String). Dari data-data tersebut, buatlah sistem yang mana dapat memenuhi fitur-fitur seperti berikut :

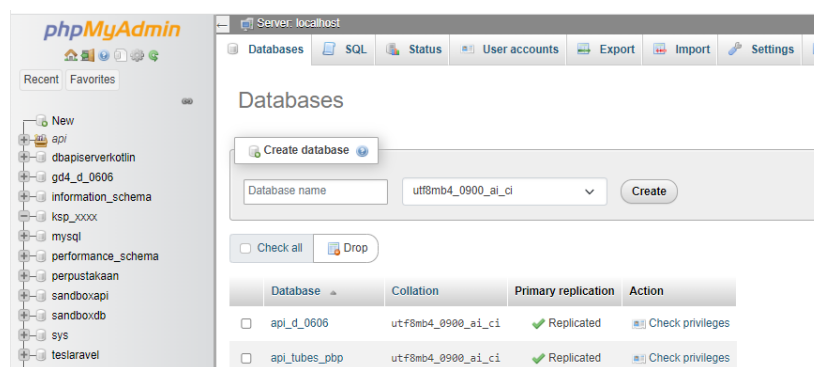
- Menambahkan data staff baru ke dalam sistem
- Menampilkan seluruh data-data yang ada di dalam sistem
- Mencari seluruh data-data staff yang ada di dalam sistem berdasarkan nomor staff
- Mengubah data staff dengan cara mencari dan mengecek berdasarkan nomor staff
- Menghapus data staff dengan berdasarkan nomor staff
- Dapat menghitung total gaji seorang staff dengan rumus = $\text{jamKerja} * 1000000$

Seluruh fitur tersebut akan terhubung dengan database XAMPP yang dimiliki oleh Perusahaan tersebut sehingga data masih tetap tersedia jika program dijalankan ulang. Pihak perusahaan juga meminta untuk membuat exception untuk error handling dengan ketentuan inputan **nomor staff tidak boleh ≥ 5 karakter** serta inputan **jamKerja hanya boleh angka saja**. Berikut adalah langkah-langkah pembuatannya:

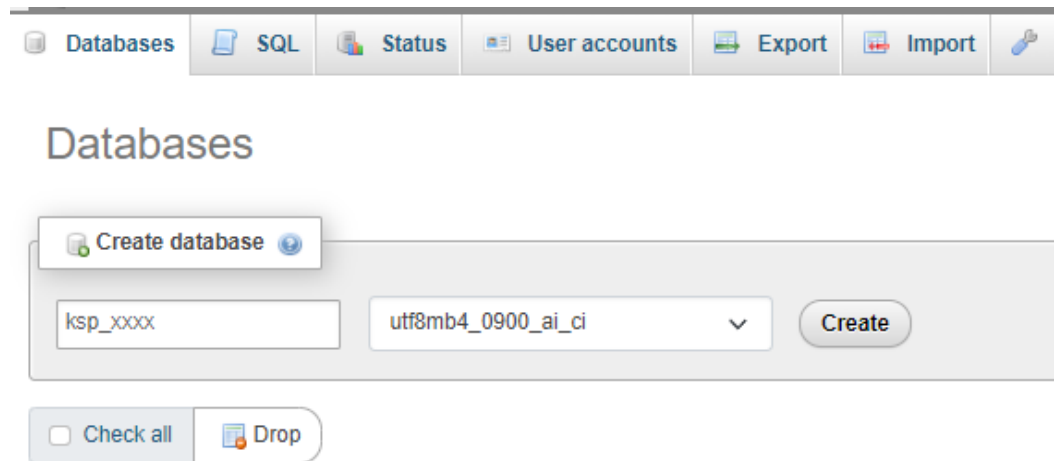
1. Buatlah sebuah project baru nama dengan KSP_XXXX (4 digit npm). **Uncheck create main class** (karena kita tidak akan menggunakan main class bawaan)
2. Buatlah database di XAMPP dengan menekan start pada apache dan mysql, kemudian silahkan tekan admin pada mysql
- 3.



4. Silahkan buka phpmyadmin kalian masing-masing dan menuju ke halaman database pada phpmyadmin.

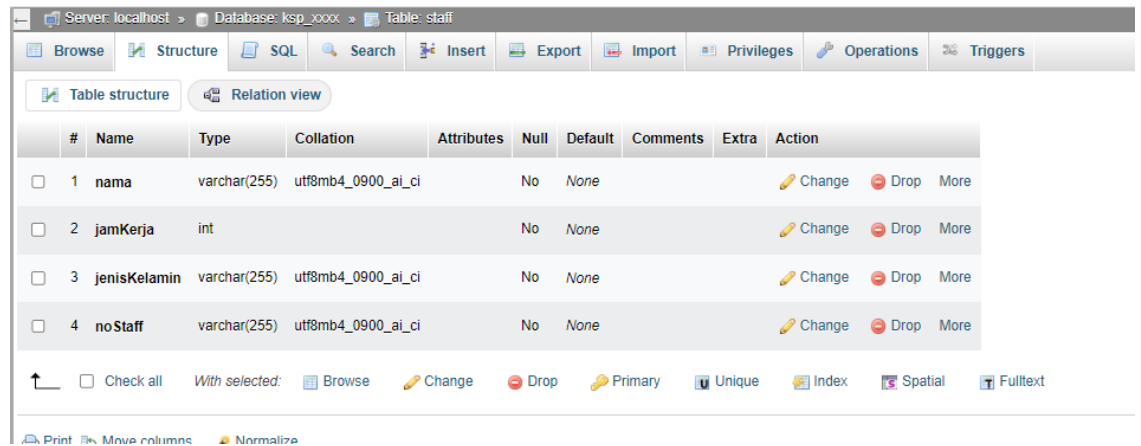


5. Buatlah database ksp_xxxx (x = 4 digit npm) pada phpmyadmin kalian



The screenshot shows the 'Databases' section of the phpMyAdmin interface. At the top, there is a navigation bar with links: Databases, SQL, Status, User accounts, Export, Import, and a help icon. Below this, the 'Databases' title is displayed. A 'Create database' button is visible. The main form contains a text input field with the value 'ksp_xxxx', a dropdown menu showing 'utf8mb4_0900_ai_ci', and a 'Create' button. Below the form, there are two buttons: 'Check all' and 'Drop'.

6. Selanjutnya buatlah table dengan nama staff dengan atribut sebagai berikut



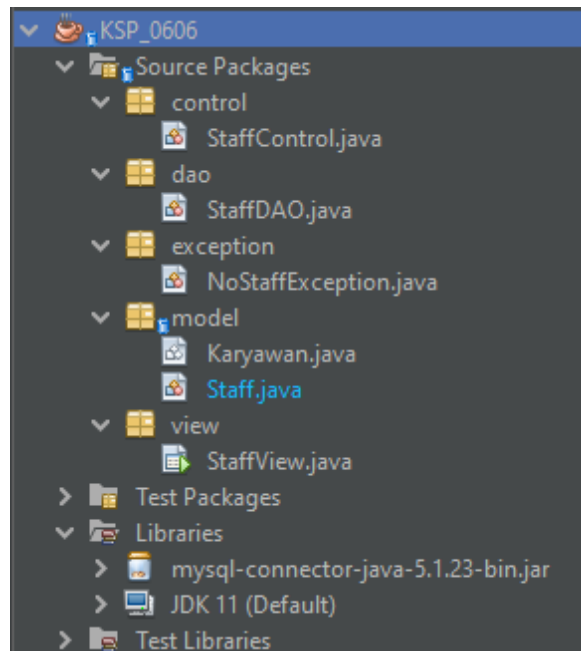
The screenshot shows the 'Table structure' view for a table named 'staff' in the 'ksp_xxxx' database. The table has four columns: 'nama' (varchar(255)), 'jamKerja' (int), 'jenisKelamin' (varchar(255)), and 'noStaff' (varchar(255)). All columns have the 'utf8mb4_0900_ai_ci' collation and are not nullable. The 'Default' column is 'None' for all. The 'Action' column provides links to 'Change', 'Drop', and 'More' for each column. Below the table structure, there are buttons for 'Check all', 'Browse', 'Change', 'Drop', 'Primary', 'Unique', 'Index', 'Spatial', and 'Fulltext'. At the bottom, there are links for 'Print', 'Move columns', and 'Normalize'.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 nama	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	2 jamKerja	int			No	None			Change Drop More
<input type="checkbox"/>	3 jenisKelamin	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More
<input type="checkbox"/>	4 noStaff	varchar(255)	utf8mb4_0900_ai_ci		No	None			Change Drop More

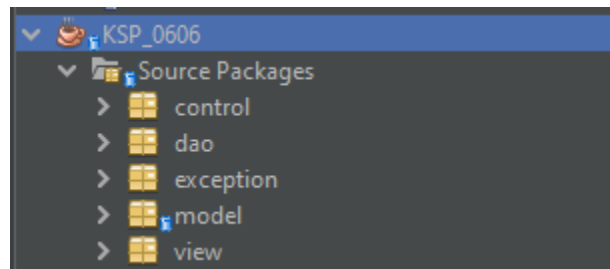
7. Langkah selanjutnya, kembali kepada netbeans dan lakukanlah import library terlebih dahulu (download mysql-connector pada link modul yang telah disediakan)

 mysql-connector-java-5.1.23-bin 4/23/2022 8:55 PM JAR File 815 KB

Lakukan import jar diatas kedalam project kalian masing-masing agar kita dapat melakukan akses ke database dengan cara klik kanan pada folder Libraries pada Netbeans - Add JAR/Folder - Blok semua library yang telah di download lalu Open. Jika berhasil, maka seperti ini tampilannya:



8. Buatlah package pada source package project kalian,



Untuk memisahkan fungsi dari masing-masing kelas, maka dibentuk 5 package yang berbeda dalam 1 project,yaitu control,dao,model,view,dan exception dengan cara klik kanan pada Source Packages>New>Folder>beri nama>Finish

9. Entity Class adalah kelas yang merepresentasikan data yang akan disimpan. Dalam konteks pembuatan aplikasi ini, entity class yang akan dibuat ini adalah kelas Karyawan sebagai kelas abstrak dan kelas turunan yaitu kelas Staff, yang merepresentasikan data yang akan disimpan. Kelas Karyawan dan staff ini merupakan bagian dari **package model**. Berikut codenya, perhatikan penempatan code pada **package model** :

a. Kelas Karyawan

```
package model;

public abstract class Karyawan {
    protected String nama;
    protected int jamKerja;
    protected String jenisKelamin;

    public Karyawan(){}

    public Karyawan(String nama, int jamKerja, String jenisKelamin) {
        this.nama = nama;
        this.jamKerja = jamKerja;
        this.jenisKelamin = jenisKelamin;
    }

    public String getNama() {
        return nama;
    }

    public void setNama(String nama) {
        this.nama = nama;
    }

    public int getJamKerja() {
        return jamKerja;
    }

    public void setJamKerja(int jamKerja) {
        this.jamKerja = jamKerja;
    }

    public String getJenisKelamin() {
        return jenisKelamin;
    }

    public void setJenisKelamin(String jenisKelamin) {
        this.jenisKelamin = jenisKelamin;
    }

    public abstract long hitungGaji();
    public abstract String showDataKaryawan();
}
```

b. Kelas Staff

```
package model;

public class Staff extends Karyawan{
    private String noStaff;

    public Staff(String noStaff, String nama, int jamKerja, String jenisKelamin) {
        super(nama, jamKerja, jenisKelamin);
        this.noStaff = noStaff;
    }

    public String getNoStaff() {
        return noStaff;
    }

    public void setNoStaff(String noStaff) {
        this.noStaff = noStaff;
    }

    @Override
    public long hitungGaji(){
        return jamKerja * 1000000;
    }

    @Override
    public String showDataKaryawan(){
        return "Staff dengan nomor " + noStaff + ", bernama " + nama +
            ", berjenis kelamin " + jenisKelamin +
            " , memiliki gaji total = " +this.hitungGaji();
    }
}
```

10. 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. Langkah pertama adalah membuat kelas dengan nama StaffDAO pada package dao. Setelah itu, jangan lupa untuk mengimport beberapa library yang akan digunakan dalam pembuatan dao. Berikut potongan code untuk library-library yang harus diimport

```
package dao;

import model.Staff;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import java.util.ArrayList;
import java.util.List;
```

Setelah selesai melakukan pembuatan kelas DAO dan juga mengimport library yang dibutuhkan, maka sekarang kita dapat mulai mengimplementasikan beberapa operasi yang dapat kita lakukan terhadap database, yaitu :

a. Membuka dan Menutup Database

Url(string) dan *path(String)* merupakan variable yang berfungsi untuk menampung url dan path mana yang akan di akses oleh netbeans. Untuk direktori path bisa menyesuaikan sesuai dengan letak file database yang disimpan.

Lalu terdapat method **makeConnection()** yang berfungsi untuk membuka file database agar dapat di akses oleh program Java dan method **closeConnection()** yang berfungsi untuk menutup file database setelah database selesai di akses. Berikut adalah kodenya :


```

public class StaffDAO {
    public static Connection CON;
    public static final String URL = "jdbc:mysql://";
    public static final String PATH = "localhost:3306/ksp_xxxx";

    public void makeConnection(){
        System.out.println("Opening Database.....");
        try {
            CON = DriverManager.getConnection(URL + PATH,"root","");
            System.out.println("Success!");
        } catch (Exception e) {
            System.out.println("Error opening");
            System.out.println(e);
        }
    }

    public void closeConnection(){
        System.out.println("Closing database...");
        try {
            CON.close();
            System.out.println("Success");
        } catch (Exception e) {
            System.out.println("Error closing database");
            System.out.println(e);
        }
    }
}

```

b. Insert Data

```

public void insert(Staff s){
    String sql = "insert into staff(noStaff, nama, jamKerja, jenisKelamin) values('"
        + s.getNoStaff() + "','" + s.getNama() + "','" + s.getJamKerja() + "','"
        + s.getJenisKelamin() + "')";
    System.out.println("Adding Staff...");

    try {
        Statement statement = CON.createStatement();
        int result = statement.executeUpdate(sql);
        System.out.println("Added" + result + "staff \n");
        statement.close();
    } catch (Exception e) {
        System.out.println("Error adding Staff...");
        System.out.println(e);
    }
}

```

Langkah selanjutnya adalah membuat method untuk menambahkan data staff ke dalam tabel staff yang ada pada database yang kita gunakan, sesuai dengan atribut yang telah dibuat dari tabel staff dengan query insert data seperti gambar dibawah ini. Berikut

diberikan tabel yang berisi penjelasan singkat mengenai query-query yang sering digunakan dalam pengaksesan database

```
// 1. Insert
// Keterangan (Perhatikan nama kolom yang ada pada query dan database harus sama, juga perhatikan urutan kolom dan
// nilai harus bersesuaian.)
// code
INSERT INTO nama_tabel(kolom1, kolom2,kolom3, dst..) VALUES (nilai1, nilai2, nilai3, dst..);

// 2. Select (View)
// Secara umum terdapat dua variasi penggunaan query SELECT. Untuk menampilkan isi dari beberapa kolom saja gunakan
// query pertama. Kita juga dapat menampilkan semua data dengan menggunakan query kedua.
// code
SELECT kolom1, kolom2, kolom3, dst .. FROM nama_tabel;
SELECT * FROM nama_tabel;

// 3. Update
// Kita dapat melakukan update nilai dari beberapa kolom pada tabel sekaligus, dan perhatikan juga nilai yang
// dimasukkan harus ke dalam kolom yang sesuai. Perhatikan juga kondisi yang digunakan, agar tidaksalah melakukan update
// code
UPDATE nama_tabel SET kolom1 = nilai1, kolom2 = nilai2, kolom3 = nilai3, dst.. WHERE condition;

// 4. Delete
// Perhatikan bagian kondisi saat melakukan query delete, jika salah memberikan kondisi, maka data yang terhapus akan
// berbeda dengan yang ingin kita hapus. Jika kita lupa memberikan kondisi, maka data pada seluruh tabel dapat terhapus
// code
DELETE FROM nama_tabel WHERE condition;
```

NB : Dalam membuat query pada program, hal yang perlu diperhatikan yaitu nama atribut/kolom dalam query harus sesuai dengan atribut database yang dibuat. Kemudian, untuk values atribut dengan tipe data selain string/char, tidak perlu ditambahkan tanda petik satu (').

c. Show Data

```
public List<Staff> showStaff(){
    String sql = "select * from staff";
    System.out.println("Mengambil data staff....");

    List<Staff> list = new ArrayList<>();

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

        if (rs!=null) {
            while (rs.next()) {
                Staff s = new Staff(rs.getString("noStaff"), rs.getString("nama"),
                    Integer.parseInt(rs.getString("jamKerja")), rs.getString("jenisKelamin"));
                list.add(s);
            }
        }
        rs.close();
        statement.close();
    } catch (Exception e) {
        System.out.println("Error reading database...");
        System.out.println(e);
    }
    return list;
}
```

d. Search Data

Kode diatas merupakan kode untuk cari data dengan query select semua data dari tabel staff yang mana inputan nomor staff sama dengan nilai nomor staff pada database.

```
public Staff searchStaff(String noStaff){
    String sql = "select * from staff where noStaff = '"+noStaff+"'";
    System.out.println("Searching Staff...");

    Staff s = null;

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

        if (rs!=null) {
            while (rs.next()) {
                s = new Staff(rs.getString("noStaff"), rs.getString("nama"),
                    Integer.parseInt(rs.getString("jamKerja")), rs.getString("jenisKelamin"));
            }
        }
        rs.close();
        statement.close();
    } catch (Exception e) {
        System.out.println("Error reading database...");
        System.out.println(e);
    }
    return s;
}
```

e. Update Data

Kode dibawah merupakan kode untuk edit data berdasarkan nomor staff. Data pada tabel staff masing-masing nilai nya akan diupdate sesuai dengan inputan user.

```
public void updateStaff(Staff s, String noStaff){
    String sql = "update staff set nama = '"+ s.getNama() +
        "',jamKerja = '"+s.getJamKerja() +
        "',jenisKelamin = '" + s.getJenisKelamin() +
        "' where noStaff = '" + noStaff + "'";
    System.out.println("Editing Staff...");

    try {
        Statement statement = CON.createStatement();
        int result = statement.executeUpdate(sql);
        System.out.println("Edit " + result + "staff\n" + noStaff);
        statement.close();
    } catch (Exception e) {
        System.out.println("Error editing Staff...");
        System.out.println(e);
    }
}
```

f. Delete Data

Menghapus data berdasarkan inputan nomor staff dengan query hapus semua data dari tabel staff yang mana inputan nomor staff sama dengan nilai nomor staff pada database.

Berikut adalah kodenya :

```
public void deleteStaff(String noStaff){
    String sql = "delete from staff where noStaff = '" + noStaff + "'";
    System.out.println("Deleting staff...");

    try {
        Statement s = CON.createStatement();
        int result = s.executeUpdate(sql);
        System.out.println("Delete "+result+" Staff\n");
        s.close();
    } catch (Exception e) {
        System.out.println("Error deleting staff...");
        System.out.println(e);
    }
}
```

11. Control Class

Control Class di sini merupakan layer yang berisi tentang bussiness logic dari sebuah aplikasi. Hal pertama yang harus dilakukan setelah membuat kelas StaffControl adalah mengimport package dao dan model yang akan digunakan, berikut adalah potongan code nya :

```
package control;

import java.util.List;
import dao.StaffDAO;
import java.util.ArrayList;
import model.Staff;

public class StaffControl {

    private StaffDAO sDAO = new StaffDAO();

    public void insertDataStaff(Staff s){
        sDAO.makeConnection();
        sDAO.insert(s);
        sDAO.closeConnection();
    }

    public String showDataStaff(){
        sDAO.makeConnection();
        List<Staff> staffData = sDAO.showStaff();
        sDAO.closeConnection();

        String staffString = "";
        for (int i = 0; i < staffData.size(); i++) {
            staffString = staffString + staffData.get(i).showDataKaryawan()
                + " | ";
        }
        return staffString;
    }

    public Staff searchStaff(String noStaff){
        Staff s = null;
        sDAO.makeConnection();
        s = sDAO.searchStaff(noStaff);
        sDAO.closeConnection();

        return s;
    }

    public void updateDataStaff(Staff s, String noStaff){
        sDAO.makeConnection();
        sDAO.updateStaff(s, noStaff);
        sDAO.closeConnection();
    }

    public void deleteDataStaff(String noStaff){
        sDAO.makeConnection();
        sDAO.deleteStaff(noStaff);
        sDAO.closeConnection();
    }

    public ArrayList<Staff> showTable(){
        sDAO.makeConnection();
        ArrayList<Staff> list = sDAO.staffList();
        sDAO.closeConnection();
        return list;
    }

}
```

Secara keseluruhan, pada masing-masing control method akan melakukan pembacaan terlebih dahulu dengan membuka database menggunakan method **makeConnection()**, lalu melakukan proses selanjutnya sesuai masing-masing method baik itu tambah, tampil, cari, edit, ataupun hapus. Setelah proses selesai, selanjutnya adalah menutup database dengan menggunakan method **closeConnection()**.

12. Exception Class

Selanjutnya kita akan mengimplementasikan Exception Handling untuk menangani error ketika program dijalankan dengan membuat kelas baru bernama NoStaffException pada package exception. Ketikkanlah kode dalam kelas NoStaffException seperti gambar dibawah ini.

```
package exception;

public class NoStaffException extends Exception{
    public String message(){
        return "noStaff harus kurang dari 5 karakter !";
    }
}
```

13. User Interface Class

Design Preview [StaffView]

GUIDED KSP OBJECT PERSISTENCE

Aksi

Pencarian

Data

Nama

Jam Kerja

Jenis Kelamin ☒ Pria ☐ Wanita

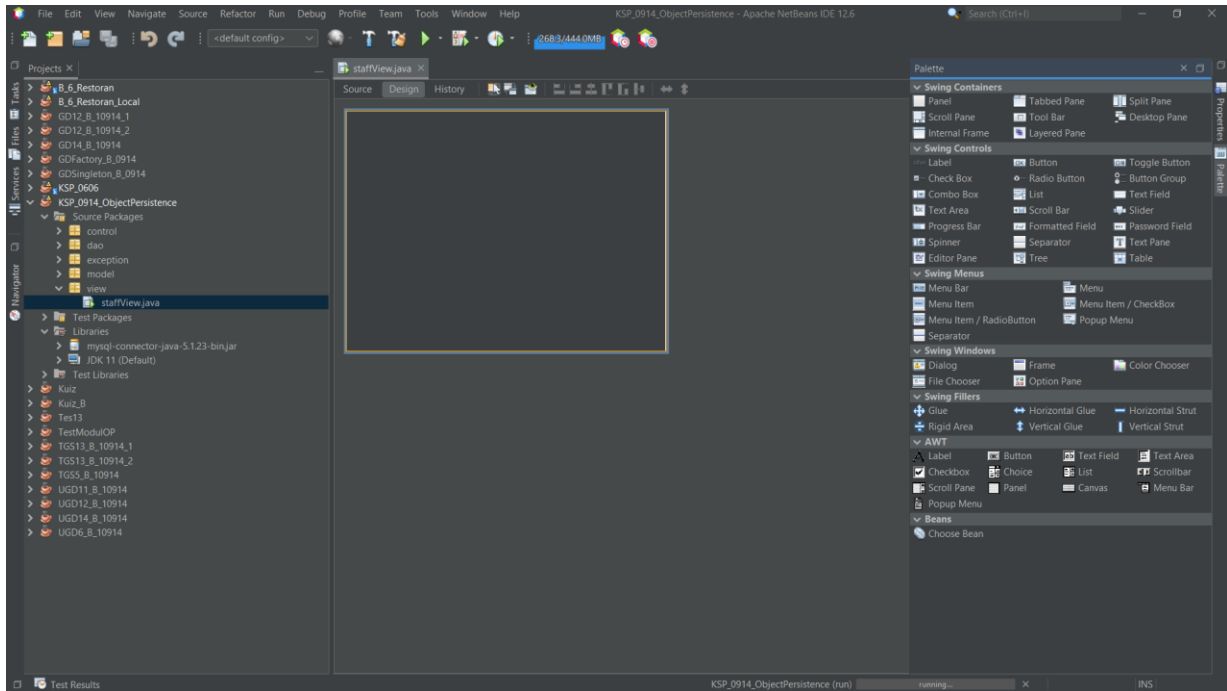
☒ Staff

No Staff

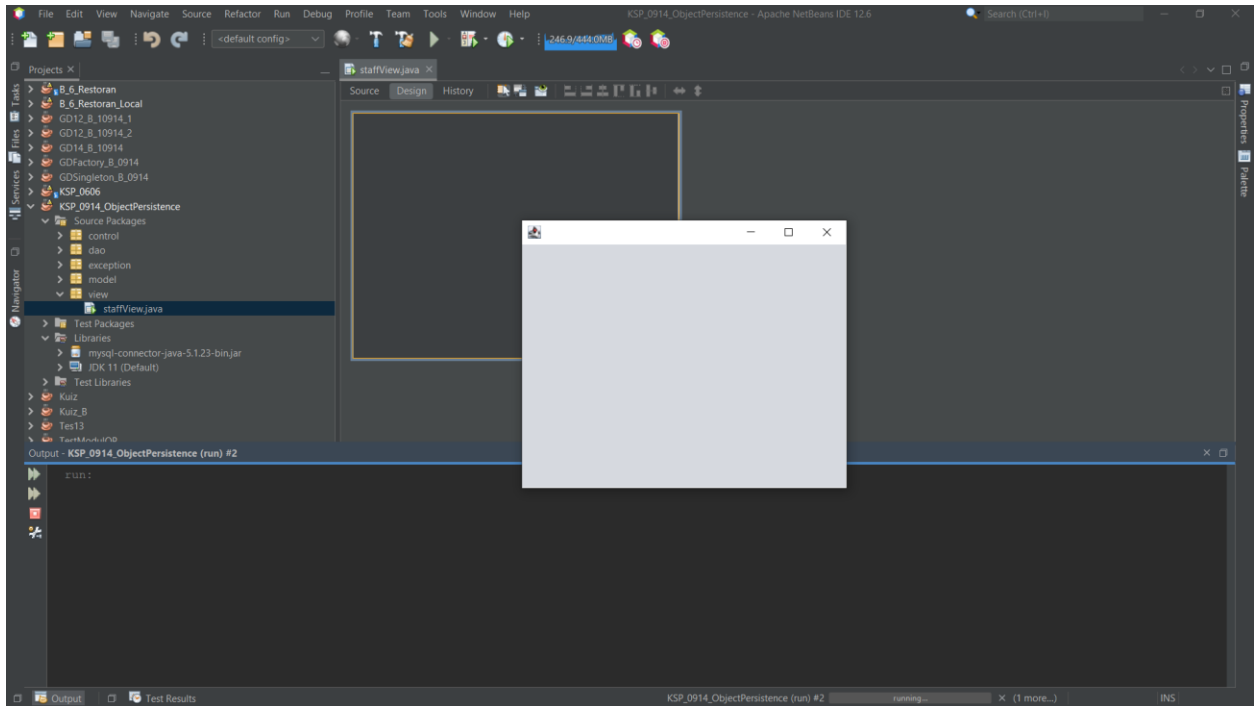
Tampil Data

--

- a. Pertama klik kanan pada package view - new - JFrame Form - isi dengan nama “StaffView”. Jika berhasil, maka tampilan nya seperti ini :



- b. Sekarang kita akan coba menjalankan (compile) project kita. Apabila terdapat pilihan untuk memilih staffView sebagai main class, pilih saja OK. Jika berhasil, maka tampilan sesudah program di compile adalah seperti ini:

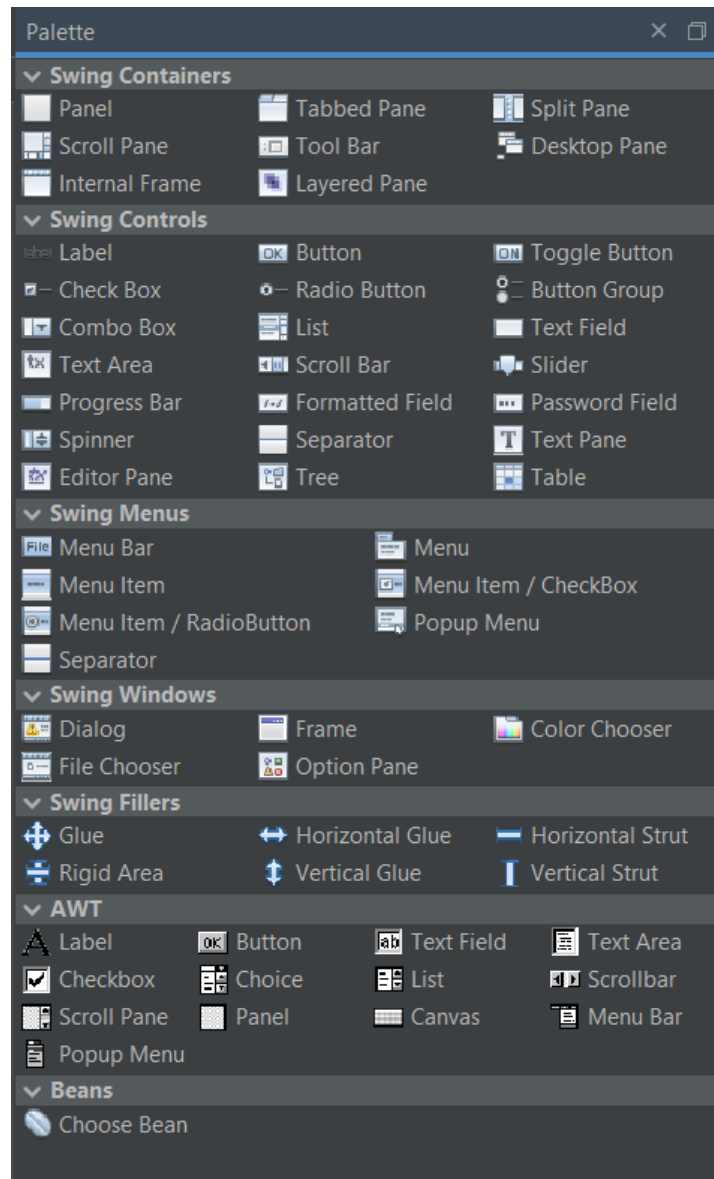


Environment GUI Java/JForm

Setelah mencoba membuat proyek GUI menggunakan Jform, maka sekarang kalian dapat memilih tab design, dimana fitur “Feel of Java” atau “Drag and Drop” dapat anda rasakan. Untuk memudahkan anda dalam pengerjaan, berikut adalah halhal yang harus anda perhatikan.

1. Pallette

Berisi komponen-komponen Swing seperti Button, Label, Text Field, dll. Untuk menggunakan komponen-komponen tersebut, kita cukup melakukan drag and drop ke bagian form yang diinginkan.



2. Tab JForm

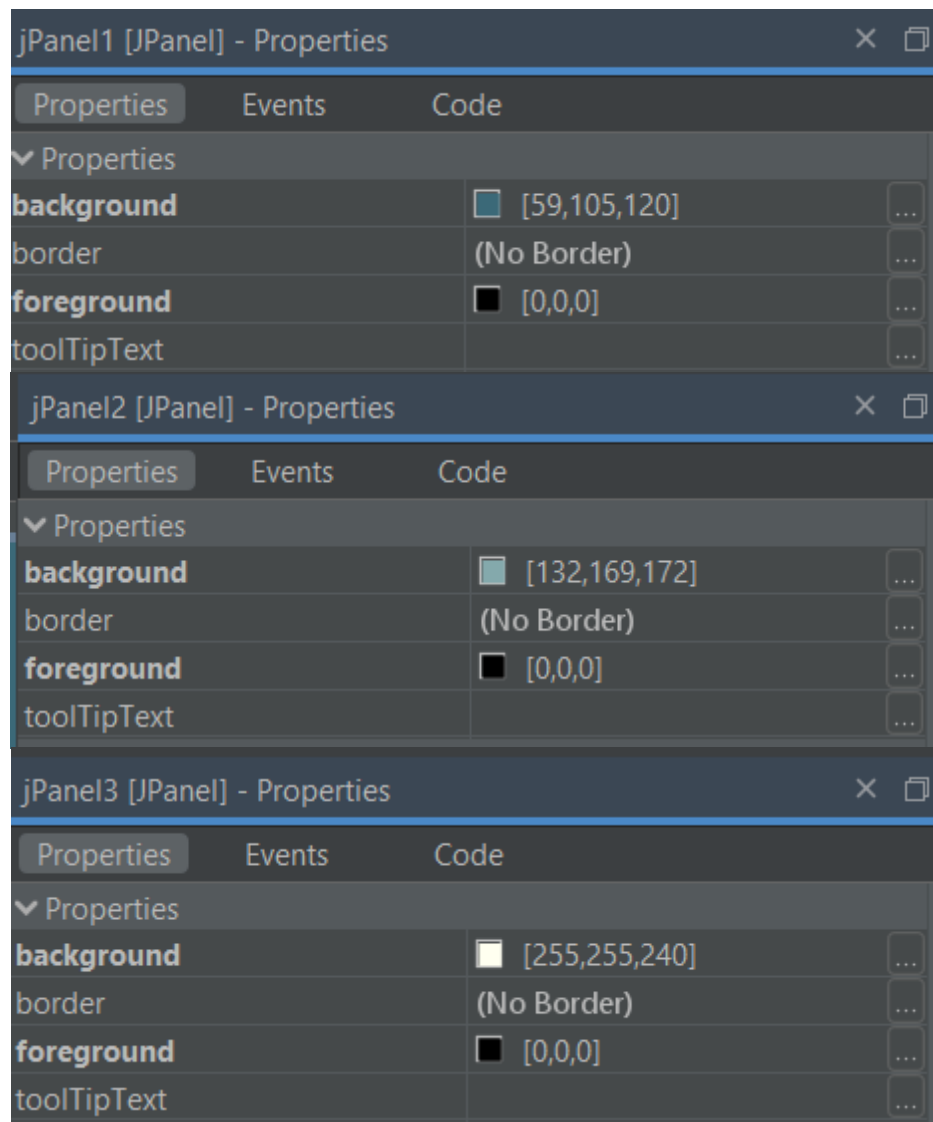


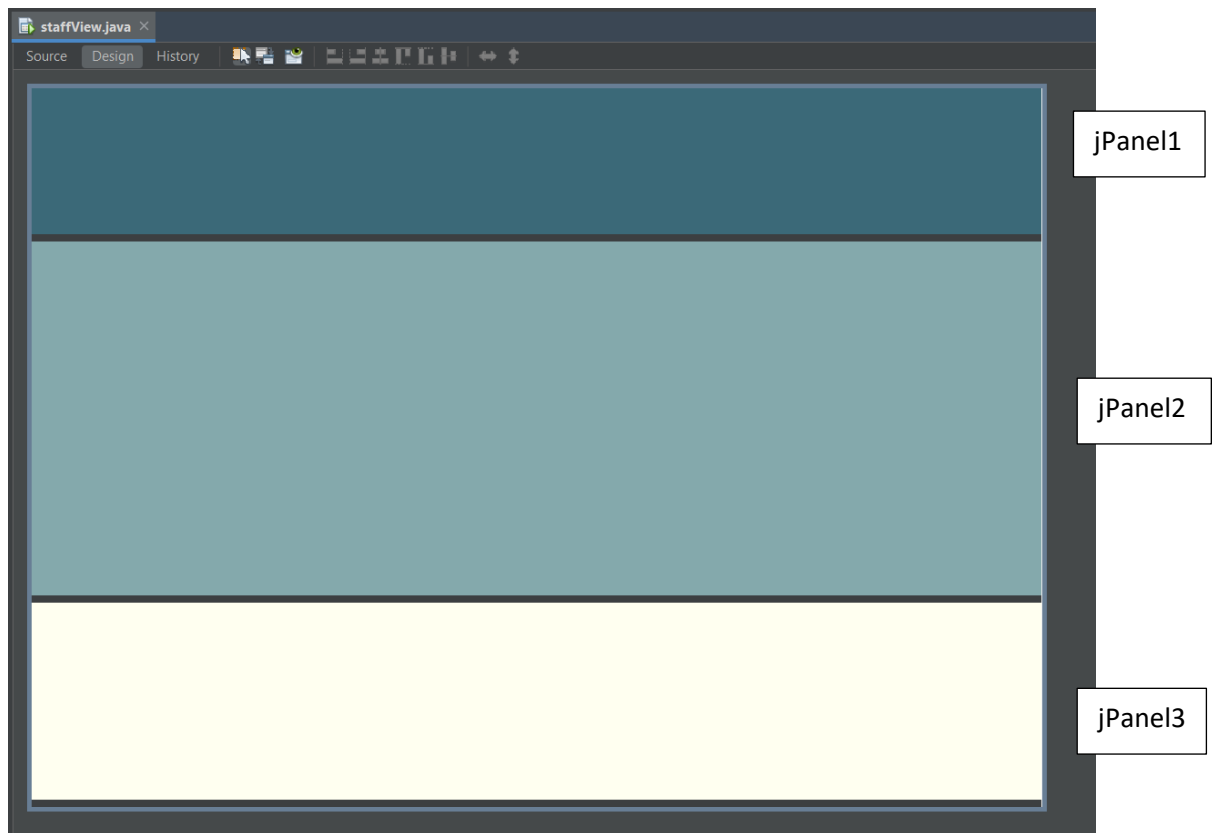
Tab Source: Berisi kode dari Swing yang kita drag and drop, dan juga kode dari Jform yang kita buat. Sebenarnya dengan fitur drag and drop, kita telah dimudahkan dalam proses kode coding karena code nya sudah tergenerate secara otomatis.

Tab Design: Sesuai dengan nama nya, tab ini akan menampilkan “design” dari project kita yang berisi Pallette, Properties Swing/Form, dsb. Design juga merupakan tempat dimana kita bisa melakukan drag and drop item yang ingin kita gunakan di form kita.

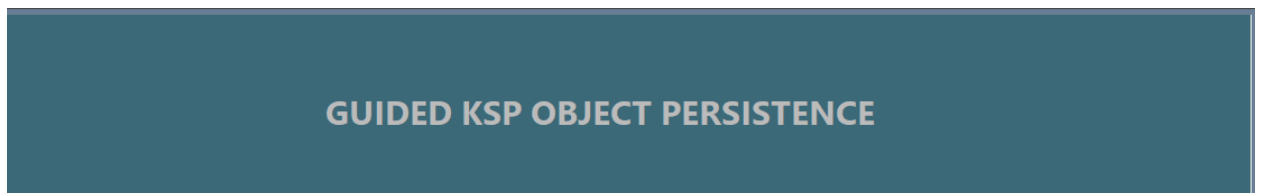
NB: Jika tools Pallette dan Properties tidak muncul, maka pilih menu Window → IDE Tools → Pilih Pallette atau Properties.

3. Kita akan menambahkan panel agar tampilan dasar lebih tampak lebih menarik dengan memberi warna background, dengan cara drag swing panel ke form yang telah dibuat. Lalu pada bagian Properties, ganti background dengan warna yang sesuai seperti contoh dibawah ini:





4. Lalu tambahkan label pada panel header dengan cara drag label pada bagian swing controls sehingga hasilnya:



Jangan lupa juga untuk mengubah warna tulisan pada label menjadi putih agar dapat terbaca dengan jelas. Atur juga alignment dari label nya menjadi center.

5. Selanjutnya tambahkan swing button, radio button, panel, text field, dan label pada panel bagian tengah untuk pengisian data dan tombol aksinya. Hasilnya akan menjadi seperti berikut:

Ubah swing tersebut dengan text dan nama variabel yang sesuai dengan fungsi nya masing-masing. Berikut cara mengubah nama nya:

- Untuk mengubah text swing dapat dilakukan dengan cara klik kanan pada swing → edit text → beri nama yang sesuai.
- Lalu untuk mengubah nama variabel seperti contoh btnTambah, txtCari, dll, dapat dilakukan dengan cara klikkanan pada swing → pilih change variabel name → beri nama yang sesuai ketentuan tabel dibawah → ok.

Component	Variable Name
PANEL AKSI	
Button Add	btnAdd
Button Update	btnUpdate
Button Delete	btnDelete
Button View	btnView
PANEL PENCARIAN	
Text Field Search	txtSearch
Button Search	btnSearch
PANEL DATA	
Text Field Nama	txtNama
Text Field Jam Kerja	txtJamKerja
Radio Button Staff	jrButtonStaff
Text Field No Staff	txtNoStaff
Button Cancel	btnCancel
Button Done	btnDone
Button About	btnAbout

6. Berikutnya adalah menambahkan swing Text Field pada bagian bawah panel yang berfungsi untuk menampilkan data member. Lalu ubah nama variabel dengan nama txtData. Dan hasilnya adalah:

Tampil Data

Setelah selesai membuat form, maka sekarang kita akan melakukan “coding” untuk form kita. Kita masuk ke bagian Source JFrame yang telah kita buat sebelumnya.

- a. Import library dan kelas yang dibutuhkan dari package yang berbeda:

```
import javax.swing.JOptionPane;
import control.StaffControl;
import exception.NoStaffException;
import model.Staff;
```

- b. Lakukan deklarasi variabel dan pembuatan object controller pada constructor:

```
public class StaffView extends javax.swing.JFrame {

    /**
     * Creates new form StaffView
     *
     */

    private StaffControl cu;
    Staff s;
    String action;

    public StaffView() {
        initComponents();
        setEnFieldAnak(false);
        setEnDisStaff(false);
        cu = new StaffControl();
    }
}
```

- c. Tambahkan method seperti pada gambar di bawah ini, untuk men-disable beberapa komponen UI yang tidak diperlukan inputan saat program dijalankan, dan melakukan clear Text Field.


```

private void setEnDisStaff(boolean val){
    txtNama.setEnabled(val);
    txtJamKerja.setEnabled(val);
    jRadioPria.setEnabled(val);
    jRadioWanita.setEnabled(val);
    jrButtonStaff.setEnabled(val);

    btnCancel.setEnabled(val);
}

private void setEnFieldAnak(boolean val ){
    txtNoStaff.setEnabled(val);
    btnDone.setEnabled(val);
    jrButtonStaff.setEnabled(val);
}

public void clearTxt(){
    txtNama.setText("");
    txtJamKerja.setText("");
    txtNoStaff.setText("");
}

```

- d. Tambahkan juga prosedur baru yang bernama NoStaffException untuk mengimplementasikan User Defined Exception yang telah kita buat tadi pada kelas NoStaffException. Fungsi prosedur ini nantinya akan berfungsi untuk mengecek apakah inputan pada text field NoStaffException tidak memiliki karakter ≥ 5 atau kosong. Berikut adalah potongan code nya.

```

public void NoStaffException() throws NoStaffException{
    if (txtNoStaff.getText().length() >= 5 || txtNoStaff.getText().equalsIgnoreCase("")) {
        throw new NoStaffException();
    }
}

```

- e. Klik 2x pada tombol btnAdd, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```

private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    setEnDisStaff(true);
    action = "Tambah";
}

```

- f. Klik 2x pada tombol btnUpdate, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```
private void btnUpdateActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (action.equalsIgnoreCase("editAtauDeleteStaff")) {
        setEnDisStaff(true);
        txtNoStaff.setEnabled(true);
        jrButtonStaff.setEnabled(true);
        btnDone.setEnabled(true);
    }else{
        setEnFieldAnak(false);
        setEnDisStaff(false);
    }
}
}
```

- g. Klik 2x pada tombol btnDelete, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    if (action.equalsIgnoreCase("editAtauDeleteStaff")) {
        cu.deleteDataStaff(txtNoStaff.getText());
        JOptionPane.showMessageDialog(this, "Data Staff Berhasil dihapus");
    }
    clearTxt();
    txtSearch.setText("");
    buttonGroup1.clearSelection();
    jrButtonStaff.setSelected(false);
    setEnDisStaff(false);
    btnViewActionPerformed(evt);
}
}
```

- h. Klik 2x pada tombol btnView, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```
private void btnViewActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtData.setText(cu.showDataStaff());
}
}
```

- i. Klik 2x pada tombol btnSearch, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```
private void btnSearchActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    clearTxt();  
  
    if(cu.searchStaff(txtSearch.getText()) != null){  
        s = cu.searchStaff(txtSearch.getText());  
        action = "editAtauDeleteStaff";  
  
        txtNoStaff.setText(s.getNoStaff());  
        txtNama.setText(s.getNama());  
  
        if(s.getJenisKelamin().equalsIgnoreCase("Pria")){  
            jRadioPria.setSelected(true);  
        } else {  
            jRadioWanita.setSelected(true);  
        }  
  
        txtJamKerja.setText(String.valueOf(s.getJamKerja()));  
        jrButtonStaff.setSelected(true);  
        btnAdd.setEnabled(false);  
    } else {  
        JOptionPane.showMessageDialog(this, "No Staff tidak ditemukan!");  
        jrButtonStaff.setSelected(false);  
        buttonGroup1.clearSelection();  
    }  
}
```

- j. Klik 2x pada tombol btnDone, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```
private void btnDoneActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    try {  
        if (action.equalsIgnoreCase("Tambah")) {  
            if (jRadioButtonStaff.isSelected() == true) {  
                NoStaffException();  
                String jenisKelamin = "";  
                if (jRadioButtonPria.isSelected() == true) {  
                    jenisKelamin = "Pria";  
                } else if (jRadioButtonWanita.isSelected() == true) {  
                    jenisKelamin = "Wanita";  
                }  
                s = new Staff(txtNoStaff.getText(), txtNama.getText(),  
                    Integer.parseInt(txtJamKerja.getText()), jenisKelamin);  
                cu.insertDataStaff(s);  
                JOptionPane.showMessageDialog(this,  
                    "Data Staff berhasil ditambah!");  
            }  
            btnViewActionPerformed(evt);  
        } else if (action.equalsIgnoreCase("editAtauDeleteStaff")) {  
            NoStaffException();  
            String jenisKelamin = "";  
            if (jRadioButtonPria.isSelected() == true) {  
                jenisKelamin = "Pria";  
            } else if (jRadioButtonWanita.isSelected() == true) {  
                jenisKelamin = "Wanita";  
            }  
            s = new Staff(txtNoStaff.getText(), txtNama.getText(),  
                Integer.parseInt(txtJamKerja.getText()), jenisKelamin);  
            cu.updateDataStaff(s, txtSearch.getText());  
            JOptionPane.showMessageDialog(this, "Data berhasil diedit");  
            btnViewActionPerformed(evt);  
        }  
    } catch (NoStaffException e) {  
        JOptionPane.showMessageDialog(this, e.getMessage());  
    } catch (NumberFormatException e2) {  
        JOptionPane.showMessageDialog(this, e2.getMessage());  
    }  
}
```

- k. Klik 2x pada tombol btnCancel, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```
private void btnCancelActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    clearTxt();
    setEnFieldAnak(false);
    setEnDisStaff(false);
    jrButtonStaff.setSelected(false);
    buttonGroup1.clearSelection();
    txtSearch.setText("");
    btnAdd.setEnabled(true);
    btnUpdate.setEnabled(true);
    btnDelete.setEnabled(true);
    btnView.setEnabled(true);
}
}
```

- l. Klik 2x pada tombol btnAbout, maka akan muncul tampilan seperti gambar di bawah, kemudian ketikkan code ini:

```
private void btnAboutActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    txtData.setText("Rakai Anandhi Mahardhika - 200710914 - 1");
}
}
```

Dibagian ini, saudara bisa isikan dengan Nama, NPM, dan juga kelompok KSP saudara sekalian.

- m. Lalu khusus untuk kode pada bagian radio buttonstaff, klik kanan pada bagian radio button staff → Properties → Events → pada baris mouseClicked, pilih dan ganti dengan jrButtonMemberMouseClicked, lalu close. Kodenya adalah sebagai berikut:

```
private void jrButtonStaffMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if(jrButtonStaff.isEnabled() == true){
        if(txtNoStaff.isEnabled() == false){
            txtNoStaff.setEnabled(true);
            btnDone.setEnabled(true);
        } else {
            txtNoStaff.setEnabled(false);
            btnDone.setEnabled(false);
        }
    }
}
}
```

Selesai !

Link github 😊 : [https://github.com/puxxbu/Guided-KSP-OP-1-](https://github.com/puxxbu/Guided-KSP-OP-1-2023)

2023

