APLIKASI

COURSE LIST BERBASIS GUI

Nama : Ahmad Ruslandia Papua

Nim : 13020200002

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS ILMU KOMPUTER

UNIVERSITAS MUSLIM INDONESIA

MAKASSAR

2022

ABSTRAK

Graphical User Interface (GUI) adalah antarmuka program yang bertindak sebagai media komunikasi antara pengguna dan perangkat lunak. GUI telah menjadi cara penting dan telah diterima dalam berinteraksi dengan perangkat lunak saat ini. GUI memungkinkan pengguna dapat lebih mudah dan lebih nyaman dalam menjalankan sebuah aplikasi. Untuk mendapatkan GUI yang sesuai perlu dilakukan perancangan saat perangkat lunak dikembangkan. Paper ini membahas perancangan aplikasi course list berbasis GUI dan mengintegrasikannya pada sebuah database. Aplikasi yang akan dirancang adalah aplikasi yang dapat menambahkan sebuah data dan menyimpan data tersebut ke dalam database.

BAB I

PENDAHULUAN

A. Latar Belakang

Graphical User Interface (GUI) adalah antarmuka program yang bertindak sebagai media interaksi antara pengguna dan perangkat lunak. Perancangan pada GUI yaitu perancangan terhadap aplikasi yang memiliki antarmuka dengan cara mendesain aplikasi GUI menggunakan komponen yang telah di sediakan IDE seperti tombol, text-field, mengatur teks atau label, dan lain-lain. Perancangan aplikasi GUI sangat penting karena fungsionalitas dari aplikasi GUI mempengaruhi kualitas seluruh sistem.

Perancangan aplikasi course list berbasis GUI berfungsi untuk memudahkan user berinteraksi dengan perangkat lunak, Khususnya dalam menginput sebuah data user akan lebih mudah menginput data menggunakan aplikasi beberbasi GUI yang sudah terintegrasi dengan database.

B. Rumusan Masalah

1. Apasaja komponen aplikasi GUI yang dirancang ?
2. Apa fungsi dari aplikasi GUI yang dirangcang?

C. Tujuan dan Manfaat

1. Mengetahui komponen aplikasi GUI yang dirancang ?
2. Mengetahui fungsi dari aplikasi GUI yang dirangcang?

Komponen Aplikasi GUI yang dirancang

Komponen yang di gunakan dalam merancang aplikasi course list berbasis GUI

Komponen Swing	***Deskripsi***
JPanel	Merupakan suatu kontainer dasar, yang sering dipakai untuk membuat complex layout dan dapat di isi dengan komponen lainnya
JLabel	Merupakan suatu komponen berupa teks, Biasanya teks yang di berikan merupakan petunjuk untuk text-field
JTextField	Merupakan suatu komponen yang berfungsi sebagai kolom tempat menulis
Jbutton	Merupakan suatu komponen yang berfungsi sebagai tombol, Tombol tersebut dapat berfungsi sesuai dengan perintah yang diberikan
Vertical Spacer	Vertical spacer merupakan komponen yang dapat memberikan jarak yang memanjang pada layout
Horizontal Spacer	Horizontal spacer merupakan komponen yang dapat memberikan jarak yang lebar pada layout

Fungsi Aplikasi GUI yang Dirancang

Aplikasi course list merupakan aplikasi berbasis GUI yang mana aplikasi tersebut di rancang untuk dapat memudahkan user apabila ingin menambahkan sebuah data kelas ke dalam sebuah database, Aplikasi course list dapat menyimpan data yang telah di inputkan ke dalam database, Hal ini memudahkan user untuk dapat menginput sebuah data pada aplilkasi.

BAB II

TINJAUAN PUSTAKA

Dasar Teori

Dalam merancang aplikasi GUI, Sangat penting untuk memperhatikan dan memahami OOP (*object oriented programming*) pada suatu class agar program dari aplikasi tersebut dapat berjalan semestinya.

Contohnya pada perancangan aplikasi course list dibutuhkan inheritance yang mana intheritance merupakan suatu proses dimana class dapat disebut sebagai super class. Berikut ini merupakan contoh program dari inheritance pada aplikasi :

```
public class CourseListForm extends JDialog {

    private JTextField CName;

    private JTextField CNumber;

    private JTextField CEnrollment;

    private JTextField CStart;

    private JTextField CEnd;

    private JButton btnAdd;

    private JButton btnCancel;

    private JPanel courselistlPanel;
```

Pada contoh program inheritance, class CourseListForm merupakan turunan dari JDialog Adapun konsep OOP yang digunakan selanjutnya yaitu method overriding, Method overriding terjadi dikarenakan method pada sub class di gunakan dan di

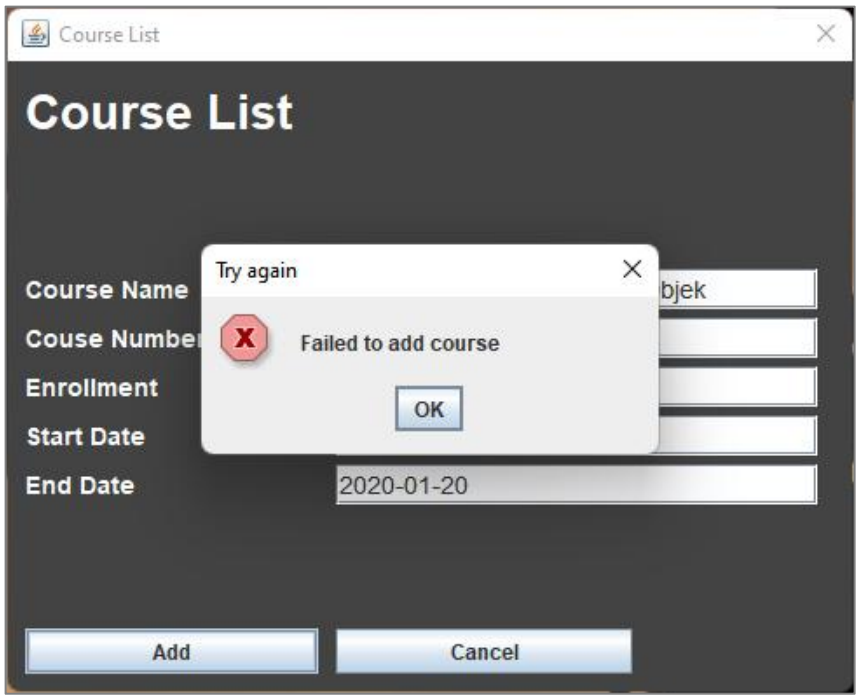
deklarasikan di class induk. Berikut ini merupakan contoh program dari method overriding pada aplikasi course list:

```
public CourseListForm(JFrame parent) {

    super(parent);

    setTitle("Course List");

    setContentPane(courselistlPanel);

    setMinimumSize(new Dimension(500, 400));

    setModal(true);

    setLocationRelativeTo(parent);

    setDefaultCloseOperation(DISPOSE_ON_CLOSE);

    btnAdd.addActionListener(new ActionListener() {

        @Override

        public void actionPerformed(ActionEvent e) {

            Course();

        }

    });

    btnCancel.addActionListener(new ActionListener() {

        @Override

        public void actionPerformed(ActionEvent e) {

            dispose();

        }

    });
```

```
        setVisible(true);

    }
```

Masalah pada Perancangan

Tidak menggunakan id pada struktur tabel sehingga menyebabkan data tidak dapat di inputkan dan tidak menggunakan mysql-connector sehingga aplikasi tidak dapat terintegrasi pada sebuah database.



Masalah Perangcangan Aplikasi 1

Table structure | Relation view

#	Name	Type	Collation	Attributes
1	course_name	varchar(30)	utf8mb4_general_ci	
2	course_number	varchar(50)	utf8mb4_general_ci	
3	course_enrollment	bigint(20)		
4	course_start	date		
5	course_end	date		

Masalah Perangcangan Aplikasi 2

```
java.sql.SQLException Create breakpoint : No suitable driver found for jdbc:mysql://localhost/
    at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:706)
    at java.sql/java.sql.DriverManager.getConnection(DriverManager.java:229)
    at CourseListForm.addCourseToDatabase(CourseListForm.java:81)
    at CourseListForm.allcourse(CourseListForm.java:61)
    at CourseListForm$1.actionPerformed(CourseListForm.java:33) <4 internal lines>
    at java.desktop/javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonL
    at java.desktop/java.awt.WaitDispatchSupport$2.run(WaitDispatchSupport.java:191) <13
```

Masalah Perangcangan Aplikasi 3

Metode yang Digunakan

Untuk mengatasi error pada aplikasi diperlukan pembuatan id pada struktur tabel database, Hal ini diperlukan untuk membedakan data-data yang di inputkan dengan angka integer pada id. Adapun saat proses pembuatan id pada struktur tabel membutuhkan AUTO_INCREMENT yang berfungsi untuk membuat angka berurut secara otomatis pada id.

Untuk mengkoneksikan aplikasi dan database sendiri diperlukan sebuah Jar file yaitu mysql-connector agar aplikasi dan database bisa saling terintegrasi

BAB III

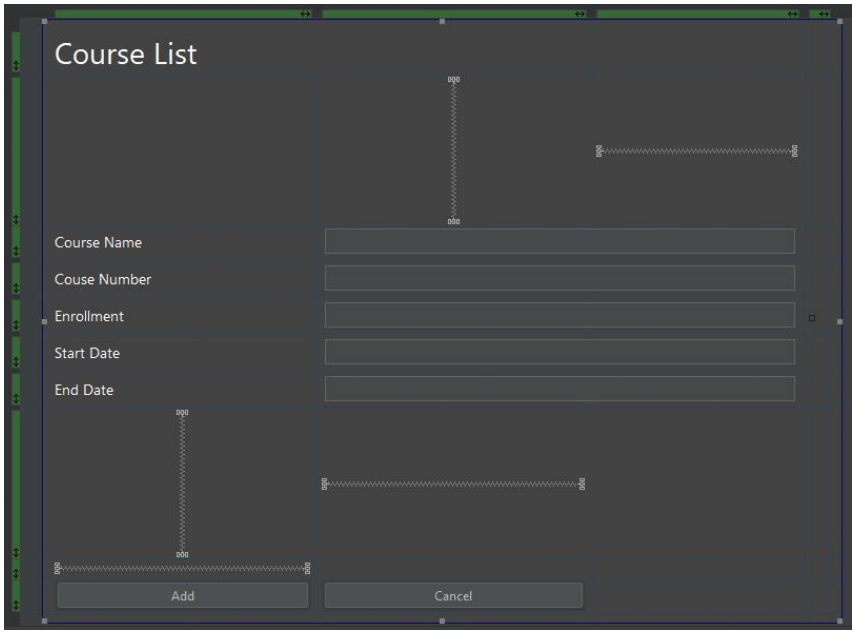
ANALISIS DAN DESAIN

Tahapan Pengembangan

Dalam merangcang dan mengembangkan aplikasi course list IDE yang digunakan yaitu IntelliJ IDE Communiry Edition, IDE tersebut digunakan karena memiliki tampilan yang lebih kompleks sehingga tahapan pengembangan dalam pembuatan aplikasi dapat dilakukan dengan mudah.

Selanjutnya dalam tahapan perancangan aplikasi diperlukan form, Form tersebut di desain sebagai tampilan pada sebuah aplikasi , Ketika form telah di desain maka tahapan selanjutnya yaitu pengkodean pada class form.

Desain



Desain Aplikasi 1

Pengkodean

import yang digunakan :

```
import javax.swing.*;

import javax.swing.JOptionPane;

import java.awt.*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.Statement;
```

Komponen Form :

```
    public class CourseListForm extends JDialog {

        private JTextField CName;

        private JTextField CNumber;

        private JTextField CEnrollment;

        private JTextField CStart;

        private JTextField CEnd;

        private JButton btnAdd;

        private JButton btnCancel;

        private JPanel courselistlPanel;}
```

Class Form :

```
        public CourseListForm(JFrame parent) {

            super(parent);

            setTitle("Course List");

            setContentPane(courselistlPanel);

            setMinimumSize(new Dimension(500, 400));

            setModal(true);

            setLocationRelativeTo(parent);

            setDefaultCloseOperation(DISPOSE_ON_CLOSE);



            btnAdd.addActionListener(new ActionListener() {

                @Override

                public void actionPerformed(ActionEvent e) {

                    GetCourse();

                }

            });

            btnCancel.addActionListener(new ActionListener() {

                @Override

                public void actionPerformed(ActionEvent e) {

                    dispose();

                }

            });
```

```
            setVisible(true);

        }
```

Koneksi database dan Mengeksekusi table database :

```
         public CrList cr;

        private CrList addCourseToDatabase(String course_name,
String course_number, String course_enrollment, String
course_start, String course_end) {

            CrList cr = null;

            final String DB_URL =
"jdbc:mysql://localhost/courselist_db";

            final String USERNAME = "root";

            final String PASSWORD = "";



            try{

                Connection conn =
DriverManager.getConnection(DB_URL, USERNAME, PASSWORD);

                // Connected to database successfully...



                Statement stmt = conn.createStatement();

                String sql = "INSERT INTO allcourse (course_name,
course_number, course_enrollment, course_start, course_end) " +

                        "VALUES (?, ?, ?, ?, ?)";

                PreparedStatement preparedStatement =
conn.prepareStatement(sql);
```

```
            preparedStatement.setString(1, course_name);

            preparedStatement.setString(2, course_number);

            preparedStatement.setString(3, course_enrollment);

            preparedStatement.setString(4, course_start);

            preparedStatement.setString(5, course_end);


            //Insert row into the table

            int addedRows = preparedStatement.executeUpdate();

            if (addedRows > 0) {

                cr = new CrList();

                cr.course_name = course_name;

                cr.course_number = course_number;

                cr.course_enrollment = course_enrollment;

                cr.course_start = course_start;

                cr.course_end = course_end;

            }

            stmt.close();

            conn.close();

        }catch(Exception e){

            e.printStackTrace();

        }


        return cr;
```

```
        }
```

Mendapatkan data yang di inputkan :

```
        private void GetCourse() {

            String course_name = CName.getText();

            String course_number = CNumber.getText();

            String course_enrollment = CEnrollment.getText();

            String course_start = CStart.getText();

            String course_end = CEnd.getText();

            if (course_name.isEmpty() || course_number.isEmpty()
|| course_enrollment.isEmpty() || course_start.isEmpty() ||
course_end.isEmpty()) {

                JOptionPane.showMessageDialog(this,

                        "Please enter all fields",

                        "Try again",

                        JOptionPane.ERROR_MESSAGE);

                return;

            }



            cr = addCourseToDatabase(course_name, course_number,
course_enrollment, course_start, course_end);

            if (cr != null) {

                dispose();
```

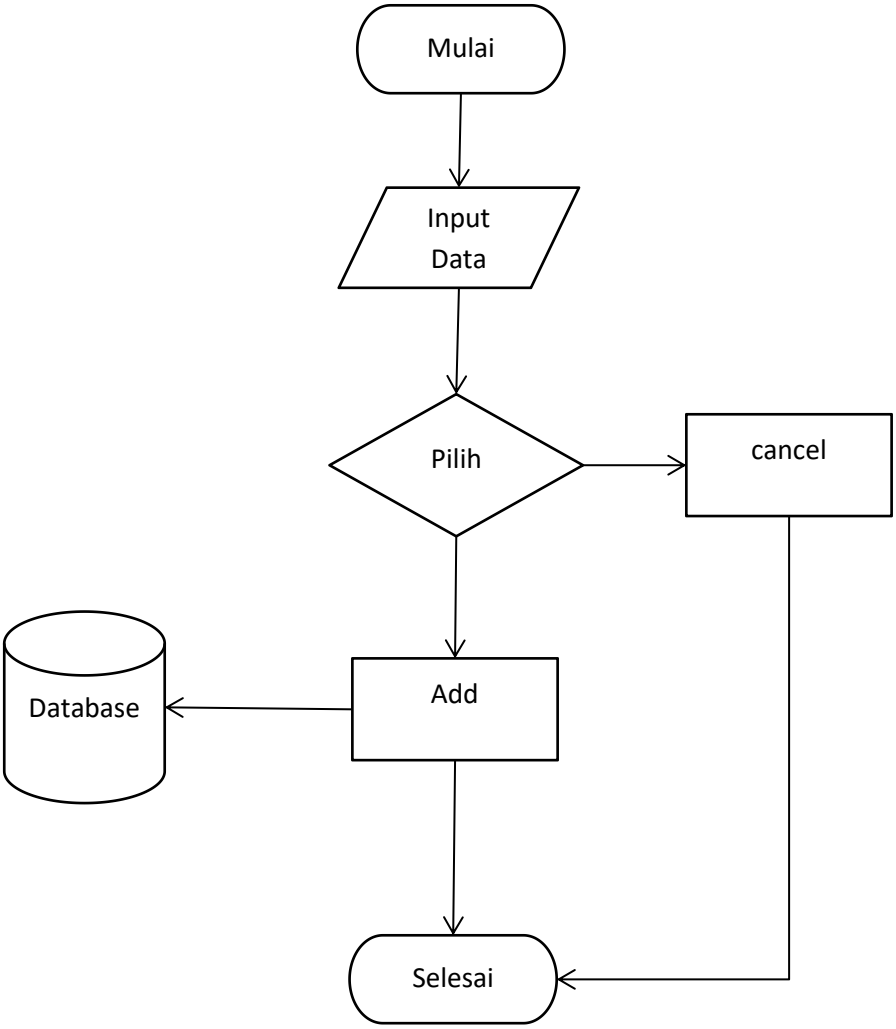
```
            }

            else {

                JOptionPane.showMessageDialog(this,

                        "Failed to add course",

                        "Try again",

                        JOptionPane.ERROR_MESSAGE);

            }

        }
```

Main :

```
        public static void main(String[] args) {

            CourseListForm myForm = new CourseListForm(null);

            CrList cr = myForm.cr;

            if (cr != null) {

                System.out.println("Successful Add : " +
cr.course_name);

            }

            else {

                System.out.println("Canceled");

            }

        }

    }
```

Flowchart

Flowchart dibuat untuk merepresentasikan aplikasi yang akan di rancang

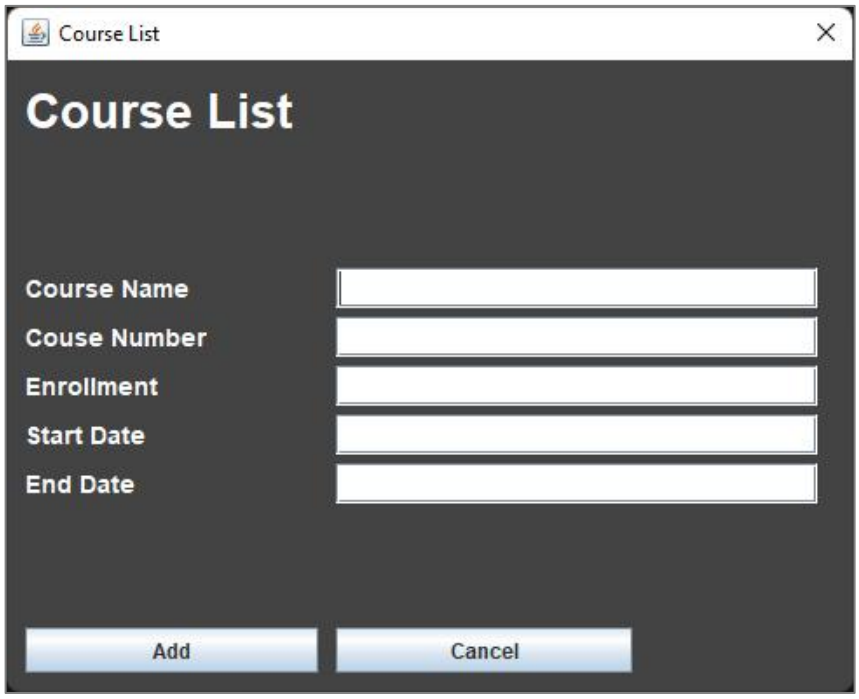


BAB IV

IMPLEMENTASI DAN PENGUJIAN

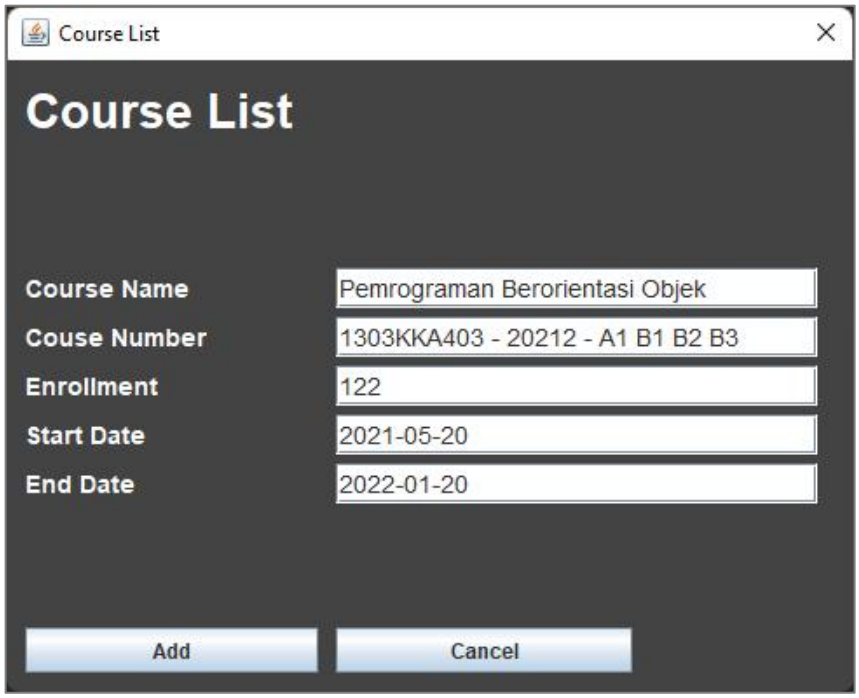
Pengujian

Menjalankan Aplikasi Berbasis GUI

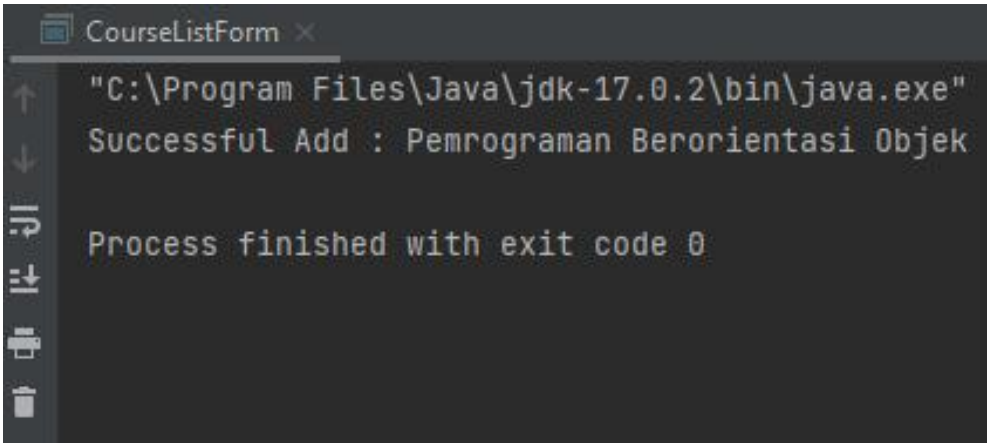


Menjalankan Aplikasi

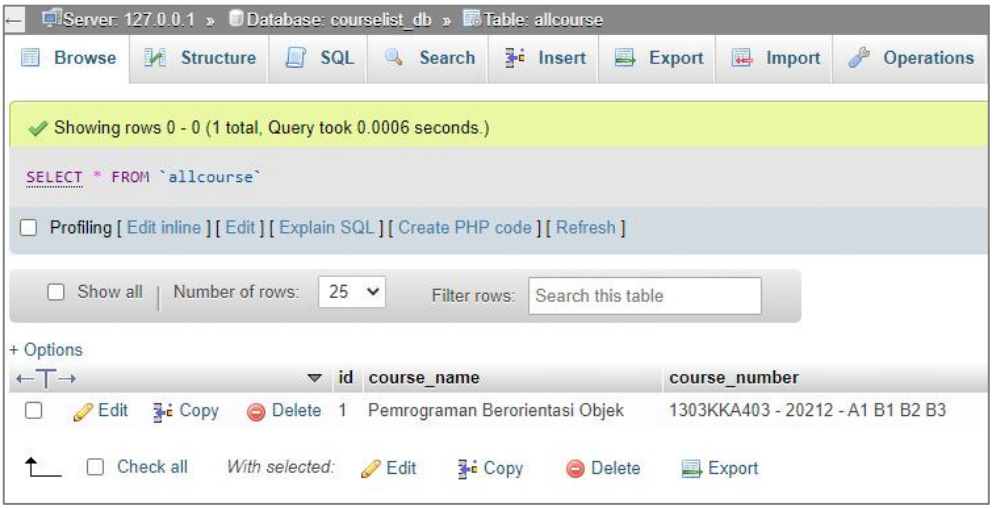
Menginputkan suatu data dan memastikan data yang telah di inputkan tersimpan pada database



Menginput dan Memastikan Data 1

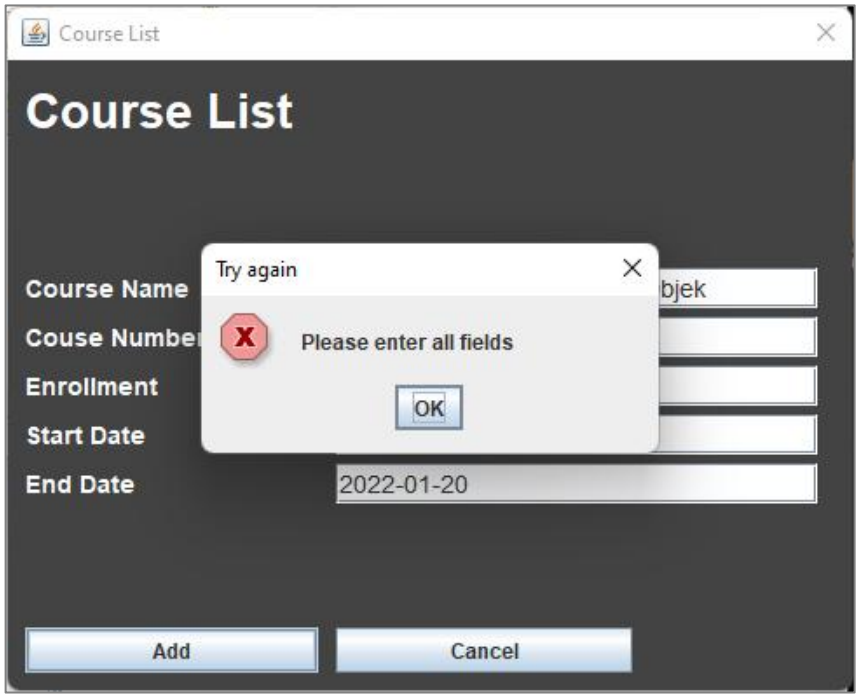


Menginput dan Memastikan Data 2



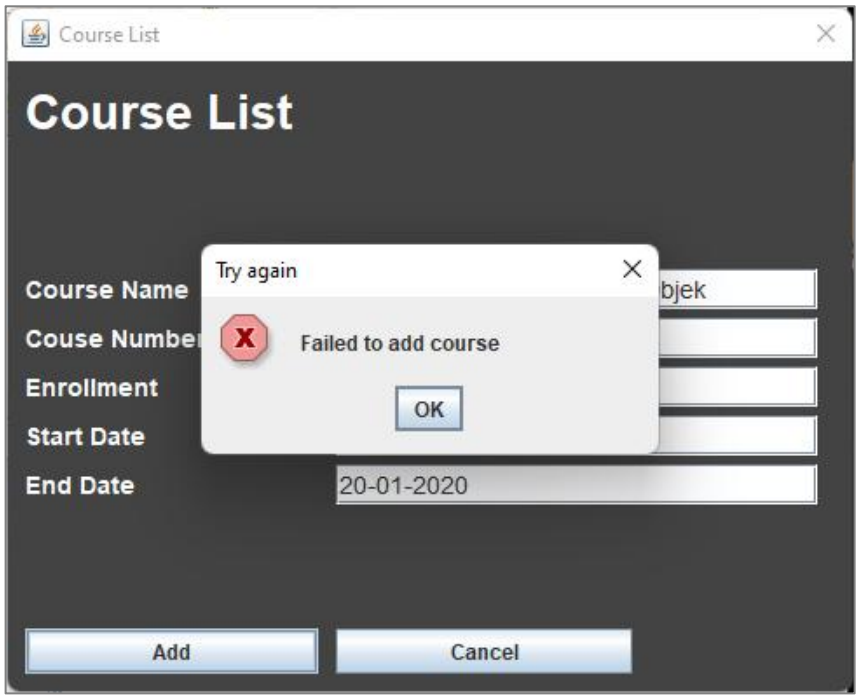
Menginput dan Memastikan Data 3

Menginputkan data yang tidak lengkap pada aplikasi

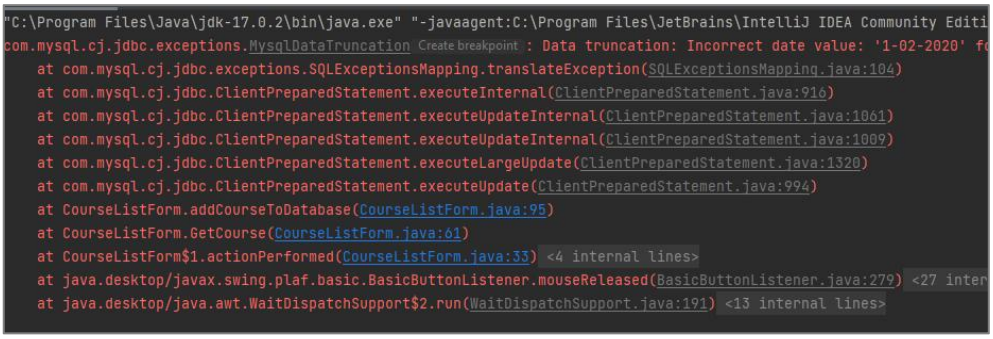


Menginput Data yang Tidak Lengkap

Menginputkan format tanggal yang salah



Menginput Format Tanggal yang Salah 1



Menginput Format Tanggal yang Salah 2

Cancel

```
Canceled

Process finished with exit code 0
```

Cancel

BAB V

PENUTUP

Kesimpulan

Aplikasi GUI merupakan program yang bertindak sebagai media komunikasi antara pengguna dan perangkat lunak. Adapuin aplikasi GUI yang dirancang adalah aplikasi yang telah terintegrasi dengan database dan aplikasi tersebut berguna untuk menginput data kedalam database.