International Journal of Informatics Technology (INJIT)
Vol. 1., No. 2, July 2023
e-ISSN : 2985-9646
https://jurnal.amrillah.net/index.php/injit

# Create A Crud App With The Library Class

Muhammad Thorik Alfiyan

Institut Teknologi Bisnis AAS Indonesia
Faculty of Technology Informatics Study Program
thorikalfian@gmail.com
* Corresponding Author

**ABSTRACT**

Java is an object-oriented programming language (OOP) and can be run on various operating system platforms. In object-oriented programming languages (OOP) many word class libraries. Class library is a function designed to make it easier for programmers to produce software. A software can be said to be good, if in it there is at least a CRUD (Create Read Update Delete) process. by using the concept of Object Oriented Programming (OOP) wrapped in a class library, complex CRUD syntax can be made simpler, so that CRUD application creation in Java Swing projects becomes faster and more practical. In experiments, the CRUD Library class is divided into 4 parts: Save class, View class, Change class, Delete group. As a result of this research, the Library class can make it easier to create CRUD applications, but it still has a drawback, namely the number of fields that can be used is limited to 10 fields.

**KEYWORDS**
Java
Class Library
CRUD
OOP



## 1. Introduction

Website programming is increasingly needed for people in all areas of life. Since the rapid development of IT, almost everything has been digitized and the use of software / software is a common activity for all circles, this is because there is a lot of software / software available for a simple small shop as well as for high-tech hospitals [1].

Java is an object-oriented programming language (OOP) and can be run on various operating system platforms. The development of Java is not only focused on one operating system, but is developed in various operating systems and is open source [5].

Object Oriented Programming (OOP) is a method of creating programs (programming paradigm) using the concept of "objects" that have data and code. This data is available in the form of attributes or properties, and code in the form of procedures [3]. So each part of the problem is an object, and the object itself is a composite of several smaller objects. An object in OOP has data or called a property that describes the properties of the object. An object in OOP also has a method in the form of a function that can be called to perform an action or change the value of the property contained in it [2].

Java is the second most popular programming language, Java syntax is difficult for beginners to understand, for example to create a simple CRUD (Create Read Update Delete) application that still requires many lines of syntax, an application will be able to meet the minimum standards if there is a CRUD process [4].

Using the concept of Object Oriented Programming (OOP) formed in a class library, complex CRUD syntax can be made simpler, so that making software in java swing projects will be faster and more practical. This research aims to form a class library that can facilitate the CRUD process in java swing projects by using the concept of Object Oriented Programming, in order to save more on writing syntax and form a simpler CRUD process, and can help programmers to focus more on making application designs without having to be bothered by the CRUD process.

## 2. Method

In this study, the authors made an experimental design that can be seen in figure 1:

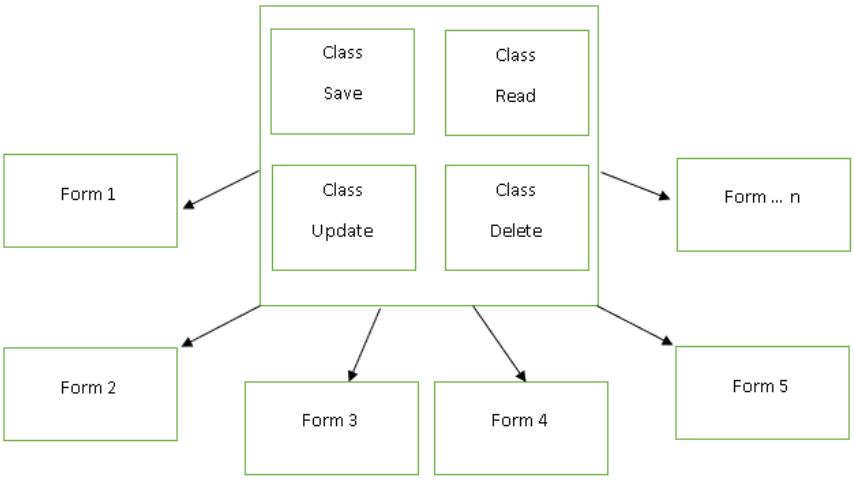


Figure 1. experiment 1

a. In the image above there is a Class Library that is created to have 4 parts in it:

b. Class Appears to display data from the database.

c. Save class to perform the process of saving data to the database.

d. Delete class to perform the process of deleting data to the database.

e. The Modify class to process changes from data to database.

The CRUD Class Library above will be called by a form in which there is a CRUD process, calling the CRUD Class Library can be done not only once by a form, but can be done repeatedly until the nth form.

## 3. Results and Discussion

In each class created in a project named "SiRUJaLi-1.0.1", the project class libray CRUD can be seen in figure 6. Once built, the library class becomes "SiRUJaLi-1.0.1.jar" and can be used in creating software in java swing projects, can be seen in figure 2 in the use of the java swing CRUD class library.

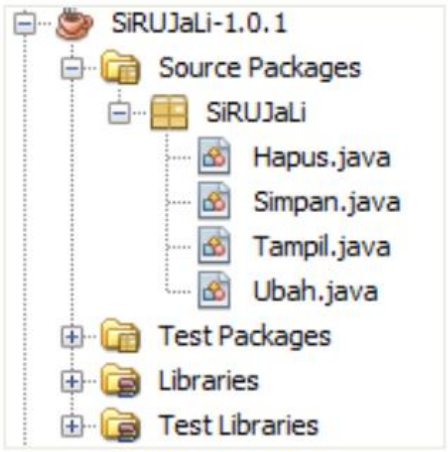


Figure 2. Project Class Library 1

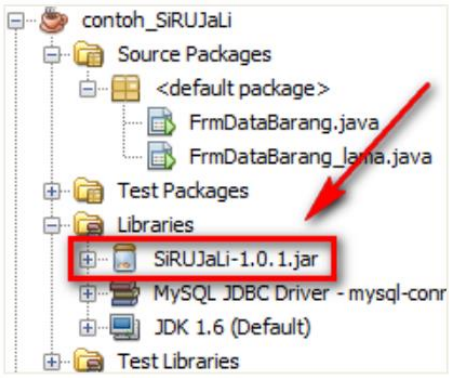


Figure 3. Usage example 1

Then in figure 4 and figure 5 can be seen the use of Class Library in the process of storing data.

```
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
                "jdbc:mysql://localhost/db_toko?user=root");

        try {
            PreparedStatement pStatement = conn.prepareStatement(""
            + "INSERT INTO tb_barang(kode_barang,nama_barang,harga_satuan)"
            + "VALUES (?,?,?)");

            pStatement.setString(1, txtKodebarang.getText());
            pStatement.setString(2, txtNamaBarang.getText());
            pStatement.setString(3, txtHargaSatuan.getText());

            if (pStatement.executeUpdate()>0){
               JOptionPane.showMessageDialog(null, "Data berhasil disimpan",
                    "Informasi", JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(null, "Data gagal disimpan",
                    "Informasi", JOptionPane.INFORMATION_MESSAGE);
            }
            pStatement.close();
            conn.close();
        }
        catch (SQLException e){
            System.out.println("kesalahan: " + e.toString());
        }

    } catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, "jdbc.Driver tidak ditemukan",
                "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    } catch (SQLException ex){
        JOptionPane.showMessageDialog(null, "koneksi gagal: \n"
                + ex.toString(), "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    }
}
```

Figure 4. No Class Library 1

```
private void btnSaveActionPerformed(java.awt.event.ActionEvent evt) {
    Simpan smp = new Simpan();
    smp.KoneksiMySQL("localhost", "db_toko", "root", "");
    smp.SimpanData(
            "tb_barang",
            "kode_barang",
            txtKodebarang.getText(),
            ("nama_barang,harga_satuan"),
            (txtNamaBarang.getText() + "," + txtHargaSatuan.getText())
    );
}
```

Figure 5. With Class Library 1

From the image above, you can see the syntax store data when using CRUD Class Library or not using CRUD Class Library. Seen in the image above, the syntas of storing data using CRUD Class Library becomes simpler.

In the process of storing data using the CRUD Class Library, simply call the SaveData() method which has parameters. Table name, table primary key, value for primary key, field name other than primary key, and value for field other than primary key. If there is more than one value for the primary key field, a comma "," is used as the separator. Before calling the SimoanData() method, you must call the ConnectionMySQL() method to connect to a MySQL database that has several parameters in order: server name, database name, username name, and password.

Then in figure 6 and figure 7, it is the use of the CRUD Class Library in the process of displaying data.

```
private void btnViewActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
                "jdbc:mysql://localhost/db_toko?user=root");

        final String[] headers = {
            "KODE BARANG", "NAMA BARANG", "HARGA SATUAN"
        };
        DefaultTableModel DftTabModel;
        DftTabModel = new DefaultTableModel(null, headers);
        jtblBarang.setModel(DftTabModel);

        try{

            String sql = "SELECT * FROM tb_barang";

            Statement stmt=conn.createStatement();
            ResultSet rs=stmt.executeQuery(sql);

            while(rs.next()){
                String dKodeBarang=rs.getString("kode_barang");
                String dNamaBarang=rs.getString("nama_barang");
                String dHargaSatuan=rs.getString("harga_satuan");
                String[] data={dKodeBarang,dNamaBarang,dHargaSatuan};
                DftTabModel.addRow(data);
            }
            rs.close();
            conn.close();
        }catch(Exception ex){
            JOptionPane.showMessageDialog(this, "ERROR: \n" + ex.toString(),
                    "Kesalahan", JOptionPane.ERROR_MESSAGE);
        }

    } catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, "jdbc.Driver tidak ditemukan",
                "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    } catch (SQLException ex){
        JOptionPane.showMessageDialog(null, "koneksi gagal: \n"
                + ex.toString(), "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    }
}
```

Figure 6. No Class Library 1

```
private void btnViewActionPerformed(java.awt.event.ActionEvent evt) {
    Tampil tmpl = new Tampil();
    tmpl.KoneksiMySQL("localhost", "db_toko", "root", "");
    tmpl.TampilData(
            "tb_barang",
            "kode_barang,nama_barang,harga_satuan",
            jtblBarang,
            "KODE BARANG,NAMA BARANG,HARGA SATUAN"
    );
}
```

Figure 7. With Class Library 1

In the picture above, it is a comparison of the syntax of displaying data when using CRUD Class Library or not using CRUD Class Library. The syntax of displaying data that uses CRUD Class Library becomes simpler. To perform the process of displaying data using the CRUD Class Library, the same as in the process of storing data by calling the ShowData() method which has parameters: the name of the table, the name of the field that will be displayed data, the name of the jtable to display data to java swing, and the title for the data displayed on the jtable. Before calling the showData() method, you must call the ConnectionMySQL() method to connect to the MySQL database which has several parameters in order: server name, database name, username name, and password.

Then in figure 8 and figure 9 is a comparison in the use of Syntax change data by using CRUD Class Library and not using CRUD Class Library.

```
private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
                "jdbc:mysql://localhost/db_toko?user=root");

        try {
            PreparedStatement pStatement = null;
            String sql ="UPDATE tb_barang " +
                "set nama_barang=?, harga_satuan=? " +
                " WHERE kode_barang=? ";
            pStatement = conn.prepareStatement(sql);

            pStatement.setString(1, txtNamaBarang.getText());
            pStatement.setString(2, txtHargaSatuan.getText());
            pStatement.setString(3, txtKodebarang.getText());

            if (pStatement.executeUpdate()>0){
               JOptionPane.showMessageDialog(null, "Data berhasil diubah",
                        "Informasi", JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(null, "Data gagal diubah",
                    "Informasi", JOptionPane.INFORMATION_MESSAGE);
            }
            pStatement.close();
            conn.close();
        }
        catch (SQLException e){
            System.out.println("kesalahan: " + e.toString());
        }

    } catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, "jdbc.Driver tidak ditemukan",
                "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    } catch (SQLException ex){
        JOptionPane.showMessageDialog(null, "koneksi gagal: \n"
                + ex.toString(), "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    }
}
```

Figure 8. No Class Library 1

```
private void btnEditActionPerformed(java.awt.event.ActionEvent evt) {
    Ubah ubh = new Ubah();
    ubh.KoneksiMySQL("localhost", "db_toko", "root", "");
    ubh.UbahData(
            "tb_barang",
            "kode_barang",
            txtKodebarang.getText(),
            ("nama_barang,harga_satuan"),
            (txtNamaBarang.getText() + "," + txtHargaSatuan.getText())
    );
}
```

Figure 9. With Class Library 1

To save data using CRUD Class Library, simply call the ChangeData() method which has parameters: table name, table primary key, value for primary key, field name other than primary key, value for field other than primary key. If the value for the primary key field is more than one then a comma "," is used as the separator. Before calling the ChangeData() method, you must call the ConnectionMySQL() method to connect to the MySQL database which has several parameters in order: server name, database name, username name, and password.

In figure 10 and figure 11 is a comparison of data delete syntas using CRUD Class Library and not using CRUD Class Library. Like other syntax, it will look simple and easy by using CRUD Class Library.

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(
                "jdbc:mysql://localhost/db_toko?user=root");

        try {
            PreparedStatement pStatement = conn.prepareStatement(""
                    + "DELETE FROM tb_barang "
                    + " WHERE kode_barang=? ");

            pStatement.setString(1, txtKodebarang.getText());

            if (pStatement.executeUpdate() > 0) {
                JOptionPane.showMessageDialog(this, "Data berhasil dihapus",
                        "Informasi", JOptionPane.INFORMATION_MESSAGE);
            } else {
                JOptionPane.showMessageDialog(this, "Data gagal dihapus",
                        "Informasi", JOptionPane.INFORMATION_MESSAGE);
            }
            pStatement.close();
            conn.close();
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(this, "ERROR: \n" + e.toString(),
                    "Kesalahan", JOptionPane.WARNING_MESSAGE);
        }

    } catch (ClassNotFoundException e) {
        JOptionPane.showMessageDialog(null, "jdbc.Driver tidak ditemukan",
                "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    } catch (SQLException ex){
        JOptionPane.showMessageDialog(null, "koneksi gagal: \n"
                + ex.toString(), "Kesalahan", JOptionPane.ERROR_MESSAGE);
        System.exit(0);
    }
}
```

Figure 10. No Class Library 1

```
private void btnDeleteActionPerformed(java.awt.event.ActionEvent evt) {
    Hapus hps = new Hapus();
    hps.KoneksiMySQL("localhost", "db_toko", "root", "");
    hps.HapusData(
            "tb_barang",
            "kode_barang",
            txtKodebarang.getText()
    );
}
```

Figure 11. With Class Library 1

To delete data using the CRUD Class Library, simply call the DeleteData() method which has parameters: table name, table primary key, and value for primary key. Before calling the DeleteData() method, you must call the ConnectionMySQL() method to connect to a MySQL database that has several parameters in order: server name, database name, username name, and password.

## 4. Conclusion

The conclusion of the trial research above, has produced a Class Library that can facilitate the creation of CRUD applications in the Java Swing project. Using this Class Library can make the process of making CRUD applications easier and faster because programmers do not need to linger in the CRUD process and programmers can focus more on making application designs. This Class Library can make it easier to create CRUD applications, but the Class Library still lacks a limited number of usable fields of 10 fields. This CRUD Class Library can be used by anyone, both beginners and experts.

## Reference

[1] Farooq M.S., Abid A., et al., 2012, “A Qualitative Framework For Introducing Programming Language At High School”, Journal of Quality and Technology Management, Volume VIII, Issue II, December 2012, Page 135–151.

[2] Abdulloh, R. (2017). Creating an Online Store with OOP, MVC, and AJAX Techniques. Elex Media Komputindo.

[3] "C++ Object-Oriented Programming Understanding | Worldilkom." https://www.duniailkom.com/tutorial-belajar-oop-c-plus-plus-pengertianpemrograman-berorientasi-objek/ (accessed Jul. 14, 2021).

[4] Cahyono, E. N. (2019). MINI SCADA SOLAR POWER PLANT BASED ON ARDUINO UNO AND VISUAL BASIC USING PHP CRUD API (Doctoral dissertation, Bhayangkara University).

[5] Barclay, K., & Savage, J. (2003). Object-oriented Design with UML and Java. Elsevier.