LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS OBJEK
MODUL 10 DATABASE

Oleh :
Alif Alpian Sahrul Muharom
(20102007)

Dosen:
Agus Priyanto, M.Kom

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

PURWOKERTO

2022

I. TUJUAN

a. Mengerti prinsip polimorfisme dalam bahasa C++ dan Java
b. Mengerti tentang prinsip polimorfisme dan pemakaiannya dalam membentuk suatu kelas.

II. TOOL

1. Apache NetBeans IDE 13
2. Java SE Development Kit 18

III. DASAR TEORI

1. API untuk Aplikasi GUI di Java
   a. AWT (Abstract Window Toolkit)

      Library dan komponen GUI (java.awt) yang pertama kali diperkenalkan oleh Java, Sun tidak merekomendasikan lagi penggunaan komponen GUI dari AWT

   b. Swing or JFC (Java Foundation Class)

      Library dan komponen GUI (javax.swing) terbaru dari Java dan yang direkomendasikan Sun untuk pemrograman GUI. Komponen Swing sebagian besar adalah turunan AWT dan lebih lengkap daripada AWT

2. Fitur Swing
   a. Komponen GUI Lengkap → button, listbox, combobox, textarea, dsb
   b. Pluggable Look-and-Feel → tampilan GUI dapat diubah sesuai dengan kehendak (tidak perlu mengikuti native sistem operasi)
   c. Data Transfer Antar Komponen → drag and drop, copy and paste
   d. Internationalization → proses desain aplikasi yang memungkinkan aplikasi dijalankan sesuai dengan preferensi tanpa rekompilasi
   e. Localization → proses translasi teks ke bahasa lokal dan menambahkan komponen lokal
3. Komponen Swing
   a. Top-Level Container

      Kontainer dasar dimana komponen lainnya diletakkan (JFrame, JDialog dan Applet) a.

      Intermediate Container

      Kontainer perantara dimana komponen lainnya diletakkan (JPanel, JScrollPane, JTabbedPane, JToolbar, JSplitPane)

   b. Atomic Component

      Komponen yang memiliki fungsi spesifik dan menerima interaksi langsung dari user (JButton, JLabel, JTextArea, dsb)

   c. Layout Manager

      Mengatur tata letak dan posisi komponen dalam kontainer (BorderLayout, BoxLayout, FlowLayout, GridBagLayout, GridLayout)

   d. Event Handling

      Menangani event yang dilakukan user (klik mouse, ketik keyboard, perbesar frame, dsb) Pemrograman Berorientasi Objek
4. JDBC dan JDBC API

   JDBC adalah standar Java Database Connectivity , dan JDBC API merupakan Java Database Connectivity Application Programming Interface (API JDBC). Semua komponen dan teknik JDBC tertanam dan diimplementasikan dalam JDBC API . Pada dasarnya , JDBC API terdiri dari satu set kelas dan interface yang digunakan untuk berinteraksi dengan database dari aplikasi Java.

Umumnya, JDBC API melakukan 3 (tiga) fungsi berikut :

a. Membangun hubungan antara aplikasi Java dan database terkait
b. Membangun dan mengeksekusi pernyataan SQL
c. Memproses hasil

Beberapa vendor database yang berbeda menyediakan berbagai driver JDBC untuk mendukung aplikasi mereka ke database. Komponen JDBC paling populer terletak di paket berikut : a. java.sql : terdiri dari komponen JDBC standar

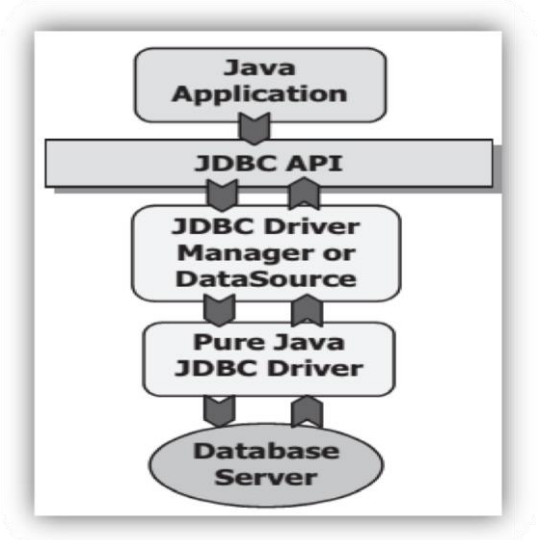
b. javax.sql : berisi standard pendukung JDBC , yang menyediakan fitur tambahan , seperti Java Naming and Directory Interface (JNDI) dan Java Transaction Service (JTS).
c. oracle.jdbc : berisi fungsi pendukung yang disediakan oleh java.sql dan javax.sql interface.
d. oracle.sql : berisi kelas dan interface yang menyediakan pemetaan Java untuk SQL

5. JDBC Components dan Architecture

JDBC API disebut juga driver JDBC, yang mengimplementasikan semua komponen JDBC termasuk kelas dan interface, untuk membangun koneksi dan memanipulasi data antara aplikasi Java dan database yang dipilih.

JDBC API berisi 2 (dua) set utama dari antarmuka, yaitu :

a. JDBC API → aplikasi (antarmuka untuk aplikasi Java Anda)
b. JDBC driver API → driver (antarmuka untuk database Anda)



Gambar 9.1. The components and architecture of a JDBC API

Tabel 9.1. Classes defined in the JDBC API

Classes	Function
DriverManager	Handle loading and unloading of drivers and establish a connection to a database
DriverPropertyInfo	All methods defined in this class are used to setup or retrieve properties of a driver. The properties can then be used by the Connection object to connect to the database
Type	The Type class is only used to define the constants used for identifying of the SQL types
Date	This class contains methods to perform conversion of SQL date formats and Java Date objects
Time	This class is similar to the Date class, and it contains methods to convert between SQL time and Java Time object
TimeStamp	This class provides additional precision to the Java Date object by adding a nanosecond field

Tabel 9.2. Interfaces defined in the JDBC API

Interfaces	Function
Driver	The primary use of the Driver interface is to create the Connection objects. It can also be used for the collection of JDBC driver meta data and JDBC driver status checking
Connection	This interface is used for the maintenance and status monitoring of a database session. It also provides data access control through the use of transaction locking
Statement	The Statement methods are used to execute SQL statements and retrieve data from the ResultSet object
PreparedStatement	This interface is used to execute precompile SQL statements. Precompile statements allow for faster and more efficient statement execution, and more important, it allows running dynamic query with querying parameters ' variation. This interface can be considered as a subclass of the Statement
CallableStatement	This interface is mainly used to execute SQL stored procedures. Both IN and OUT parameters are supported. This interface can be considered as a subclass of the Statement
ResultSet	The ResultSet object contains the queried result in rows and columns format. This interface also provides methods to retrieve data returned by an SQL statement execution. It also contains methods for SQL data type and JDBC data type conversion
ResultSetMetaData	This interface contains a collection of metadata information or physical descriptions associated with the last ResultSet object
DatabaseMetaData	This interface contains a collection of metadata regarding to the database used, including the database version, table names, columns, and supported functions

IV. GUIDED

1. CobaThrow.java

```
package main.java.com.Alpiann.pertemuan10;

import java.sql.*;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;
/**
 *
 * @author TUF F15
 */
public class FromKoneksi extends javax.swing.JFrame {
    private static Connection koneksi;
    private DefaultTableModel model;
    /**
     * Creates new form FromKoneksi
     */
    public FromKoneksi() {
        initComponents();
        model = new DefaultTableModel();
        dataTable.setModel(model);
        model.addColumn("ID");
        model.addColumn("Nama");
        model.addColumn("Alamat");
        model.addColumn("Telephon");
        ambilDataTable();
    }

    private static Connection bukaKoneksi(){
        if (koneksi == null){
            try{
                String url = "jdbc:mysql://localhost:3306/belajar";
                String user = "root";
                String password = "";

                DriverManager.registerDriver(new
com.mysql.cj.jdbc.Driver());
                koneksi = DriverManager.getConnection(url, user,
password);
            }catch(SQLException e){
                System.out.println("Eror Membuat Koneksi");
            }
        }
        return koneksi;
```

```
    }

    private void ambilDataTable(){
        model.getDataVector().removeAllElements();
        model.fireTableDataChanged();
        try{
            Connection c = bukaKoneksi();
            Statement s = c.createStatement();
            String sql = "SELECT * FROM anggota";
            ResultSet r = s.executeQuery(sql);

            while (r.next()){
                Object[] o = new Object[4];
                o[0] = r.getString("id");
                o[1] = r.getString("nama");
                o[2] = r.getString("alamat");
                o[3] = r.getString("telp");
                model.addRow(o);
            }
            r.close();
            s.close();
        }catch(SQLException e){
            System.out.println("Terjadi Kesalahan " +
e.getMessage());
        }
    }

    private void ambilTableKlik() {
        int i = dataTable.getSelectedRow();
        if (i == -1) {
            return;
        }
        String kode = (String) model.getValueAt(i, 0);
        kodeLabel.setText(kode);
        String nama = (String) model.getValueAt(i, 1);
        namaTextField.setText(nama);
        String alamat = (String) model.getValueAt(i, 2);
        alamatTextField.setText(alamat);
        String telp = (String) model.getValueAt(i, 3);
        telephonTextField.setText(telp);

        addButton.setEnabled(false);
    }
```

```
    private void backIntoDefaultState() {
        kodeLabel.setText("0");
        namaTextField.setText("");
        alamatTextField.setText("");
        telephonTextField.setText("");
        addButton.setEnabled(true);
    }

    /**
     * This method is called from within the constructor to
initialize the form.
     * WARNING: Do NOT modify this code. The content of this method
is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        namaLabel = new javax.swing.JLabel();
        namaTextField = new javax.swing.JTextField();
        alamatLabel = new javax.swing.JLabel();
        alamatTextField = new javax.swing.JTextField();
        telephonLabel = new javax.swing.JLabel();
        telephonTextField = new javax.swing.JTextField();
        kodeLabel = new javax.swing.JLabel();
        jScrollPane1 = new javax.swing.JScrollPane();
        dataTable = new javax.swing.JTable();
        addButton = new javax.swing.JButton();
        editButton = new javax.swing.JButton();
        deletButton = new javax.swing.JButton();
        refreshButton = new javax.swing.JButton();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("CRUD dengan Java Neatbens dan Mysql");

        namaLabel.setText("Nama");

        alamatLabel.setText("Alamat");

        telephonLabel.setText("Telephon");
```

```
        kodeLabel.setText("0");

        dataTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        dataTable.addMouseListener(new java.awt.event.MouseAdapter()
{
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                dataTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(dataTable);

        addButton.setText("Add");
        addButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                addButtonActionPerformed(evt);
            }
        });

        editButton.setText("Edit");
        editButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                editButtonActionPerformed(evt);
            }
        });

        deletButton.setText("Delete");
        deletButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                deletButtonActionPerformed(evt);
```

```
            }
        });

        refreshButton.setText("Refresh");
        refreshButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                refreshButtonActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(namaLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 37,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                        .addComponent(namaTextField)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                        .addComponent(kodeLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 17,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(telephonLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(telephonTextField))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(alamatLabel,
javax.swing.GroupLayout.PREFERRED_SIZE, 43,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(12, 12, 12)
                        .addComponent(alamatTextField))
```

```
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 375,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(addButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                                .addComponent(editButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                                .addComponent(deletButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                                .addComponent(refreshButton))
                            .addComponent(jLabel1))
                        .addGap(0, 0, Short.MAX_VALUE)))
                .addContainerGap())
        );

        layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {alamatLabel, namaLabel});

        layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {addButton, deletButton, editButton,
refreshButton});

        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(namaLabel)
                    .addComponent(namaTextField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(kodeLabel))
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(alamatTextField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(alamatLabel))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(telephonTextField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(telephonLabel))
                .addGap(9, 9, 9)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 275,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
12, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(addButton)
                    .addComponent(editButton)
                    .addComponent(deletButton)
                    .addComponent(refreshButton))
                .addContainerGap())
        );

        pack();
    }// </editor-fold>
    private void initComponent(){

       jLabel1 = new javax.swing.JLabel();
        namaLabel = new javax.swing.JLabel();
        namaTextField = new javax.swing.JTextField();
        alamatLabel = new javax.swing.JLabel();
        alamatTextField = new javax.swing.JTextField();
        telephonLabel = new javax.swing.JLabel();
        telephonTextField = new javax.swing.JTextField();
        kodeLabel = new javax.swing.JLabel();
```

```
        jScrollPane1 = new javax.swing.JScrollPane();
        dataTable = new javax.swing.JTable();
        addButton = new javax.swing.JButton();
        editButton = new javax.swing.JButton();
        deletButton = new javax.swing.JButton();
        refreshButton = new javax.swing.JButton();


setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);

        jLabel1.setText("CRUD dengan Java NetBeans dan MySQL");

        namaLabel.setText("Nama");

        alamatLabel.setText("Alamat");

        telephonLabel.setText("Telepon");

        kodeLabel.setText("0");

        dataTable.setModel(new javax.swing.table.DefaultTableModel(
            new Object [][] {
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null},
                {null, null, null, null}
            },
            new String [] {
                "Title 1", "Title 2", "Title 3", "Title 4"
            }
        ));
        dataTable.addMouseListener(new java.awt.event.MouseAdapter()
{
            public void mouseClicked(java.awt.event.MouseEvent evt) {
                dataTableMouseClicked(evt);
            }
        });
        jScrollPane1.setViewportView(dataTable);

         addButton.setText("Add");
        addButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                addButtonActionPerformed(evt);
```

```
            }
        });

        editButton.setText("Edit");
        editButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                editButtonActionPerformed(evt);
            }
        });

        deletButton.setText("Delete");
        deletButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                deletButtonActionPerformed(evt);
            }
        });

        refreshButton.setText("Refresh");
        refreshButton.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                refreshButtonActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(namaLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                        .addComponent(namaTextField)
```

```
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                        .addComponent(kodeLabel))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(alamatLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                        .addComponent(alamatTextField))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(telephonLabel)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED
)
                        .addComponent(telephonTextField))
                    .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 0, Short.MAX_VALUE)
                    .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
                            .addComponent(jLabel1)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(addButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(editButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(deletButton)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                .addComponent(refreshButton)))
                        .addGap(0, 70, Short.MAX_VALUE)))
                .addContainerGap())
        );

         layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {alamatLabel, namaLabel, telephonLabel});

        layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {addButton, deletButton, editButton,
refreshButton});

        layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
```

```
                .addContainerGap()
                .addComponent(jLabel1)
                .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(namaLabel)
                    .addComponent(namaTextField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(kodeLabel))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(alamatLabel)
                    .addComponent(alamatTextField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(telephonLabel)
                    .addComponent(telephonTextField,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE, 275,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.BASELINE)
                    .addComponent(addButton)
                    .addComponent(editButton)
                    .addComponent(deletButton)
                    .addComponent(refreshButton))
                .addContainerGap())
        );
```

```
        pack();
    }
    private void
deletButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Connection c = bukaKoneksi();
        String sqlKode = "DELETE FROM anggota " + "WHERE id = '" +
kodeLabel.getText() + "';";


        try {
            PreparedStatement p2 = (PreparedStatement)
c.prepareStatement(sqlKode);
            p2.executeUpdate();
            p2.close();
        } catch(SQLException e) {
            JOptionPane.showMessageDialog(this, "Terjadi kesalahan "
+ e.getMessage());
        }
    }


    private void addButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
        Connection c = bukaKoneksi();


        if ("Add".equals(addButton.getText())) {
            addButton.setText("Save");
            editButton.setText("Cancel");
            deletButton.setEnabled(false);
            refreshButton.setEnabled(false);
        }else if ("Save".equals(addButton.getText())){
            String sqlKode =
                    "INSERT INTO anggota (nama, alamat, telp) VALUES"
+
                    " ('" + namaTextField.getText() + "', " +
                    " '" + alamatTextField.getText() + "', " +
                    " '" + telephonTextField.getText() + "')";


            try {
                PreparedStatement p2 = (PreparedStatement)
c.prepareStatement(sqlKode);
                p2.executeUpdate();
                p2.close();

                backIntoDefaultState();
```

```
            }catch(SQLException e) {
                JOptionPane.showMessageDialog(this, "Terjadi
kesalahan " + e.getMessage());
            }

            addButton.setText("Add");
            editButton.setText("Edit");
            deletButton.setEnabled(true);
            refreshButton.setEnabled(true);
        }else if ("Update".equals(addButton.getText())){
            String sqlKode = "Update anggota SET nama = '" +
namaTextField.getText() + "'," +
                    "alamat = '" + alamatTextField.getText() + "'" +
                    "WHERE ID = '" + kodeLabel.getText() + "';";

            try {
                PreparedStatement p2 = (PreparedStatement)
c.prepareStatement(sqlKode);
                p2.executeUpdate();
                p2.close();

                backIntoDefaultState();
            }catch(SQLException e) {
                JOptionPane.showMessageDialog(this, "Terjadi
kesalahan " + e.getMessage());

                addButton.setText("Add");
                editButton.setText("Edit");
                deletButton.setEnabled(true);
                refreshButton.setEnabled(true);
            }
        }
    }

    private void editButtonActionPerformed(java.awt.event.ActionEvent
evt) {
        // TODO add your handling code here:
        if ("Edit".equals(editButton.getText())) {
            addButton.setText("Update");
            editButton.setText("Cancel");
            addButton.setEnabled(true);
            deletButton.setEnabled(false);
            refreshButton.setEnabled(false);
        } else if ("Cancel".equals(editButton.getText())) {
            addButton.setText("Add");
```

```
            editButton.setText("Edit");
            deletButton.setEnabled(true);
            refreshButton.setEnabled(true);
        }
    }

    private void
refreshButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        ambilDataTable();
        backIntoDefaultState();
    }

    private void dataTableMouseClicked(java.awt.event.MouseEvent evt)
{
         // TODO add your handling code here:
         ambilTableKlik();
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel
setting code (optional) ">
        /* If Nimbus (introduced in Java SE 6) is not available, stay
with the default look and feel.
         * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.h
tml
         */
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(FromKoneksi.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
```

```
java.util.logging.Logger.getLogger(FromKoneksi.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(FromKoneksi.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(FromKoneksi.class.getName()).log(j
ava.util.logging.Level.SEVERE, null, ex);
        }
        //</editor-fold>



        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new FromKoneksi().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton addButton;
    private javax.swing.JLabel alamatLabel;
    private javax.swing.JTextField alamatTextField;
    private javax.swing.JTable dataTable;
    private javax.swing.JButton deletButton;
    private javax.swing.JButton editButton;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JLabel kodeLabel;
    private javax.swing.JLabel namaLabel;
    private javax.swing.JTextField namaTextField;
    private javax.swing.JButton refreshButton;
    private javax.swing.JLabel telephonLabel;
    private javax.swing.JTextField telephonTextField;
    // End of variables declaration
}
```


CRUD dengan Java Neatbens dan Mysql
Nama
Alif Alpian
1
Alamat
Majalengka
Telephon
20102007
ID
Nama
Alamat
Telephon
1
Alif Alpian
Majalengka
20102007
Add
Edit
Delete
Refresh

V. KESIMPULAN

JDBC adalah standar Java Database Connectivity , dan JDBC API
merupakan Java Database Connectivity Application Programming Interface (API JDBC). Semua komponen dan teknik JDBC tertanam dan diimplementasikan dalam JDBC API . Pada dasarnya , JDBC API terdiri dari satu set kelas dan interface yang digunakan untuk berinteraksi dengan database dari aplikasi Java. Umumnya, JDBC API melakukan 3 (tiga) fungsi berikut :

a. Membangun hubungan antara aplikasi Java dan database terkait
b. Membangun dan mengeksekusi pernyataan SQL
c. Memproses hasil

Beberapa vendor database yang berbeda menyediakan berbagai driver JDBC untuk mendukung aplikasi mereka ke database. Komponen JDBC paling populer terletak di paket berikut :

a. java.sql : terdiri dari komponen JDBC standar
b. javax.sql : berisi standard pendukung JDBC , yang menyediakan fitur tambahan , seperti Java Naming and Directory Interface (JNDI) dan Java Transaction Service (JTS).
c. oracle.jdbc : berisi fungsi pendukung yang disediakan oleh java.sql dan javax.sql interface.
d. oracle.sql : berisi kelas dan interface yang menyediakan pemetaan Java untuk SQL

JDBC Components dan Architecture

JDBC API disebut juga driver JDBC, yang mengimplementasikan semua komponen JDBC termasuk kelas dan interface, untuk membangun koneksi dan memanipulasi data antara aplikasi Java dan database yang dipilih.

JDBC API berisi 2 (dua) set utama dari antarmuka, yaitu :

a. JDBC API aplikasi (antarmuka untuk aplikasi Java Anda)
b. JDBC driver API driver (antarmuka untuk database Anda)