JURNAL MODUL 9

Dhiya Aghniyaar Rahman - 1301190284

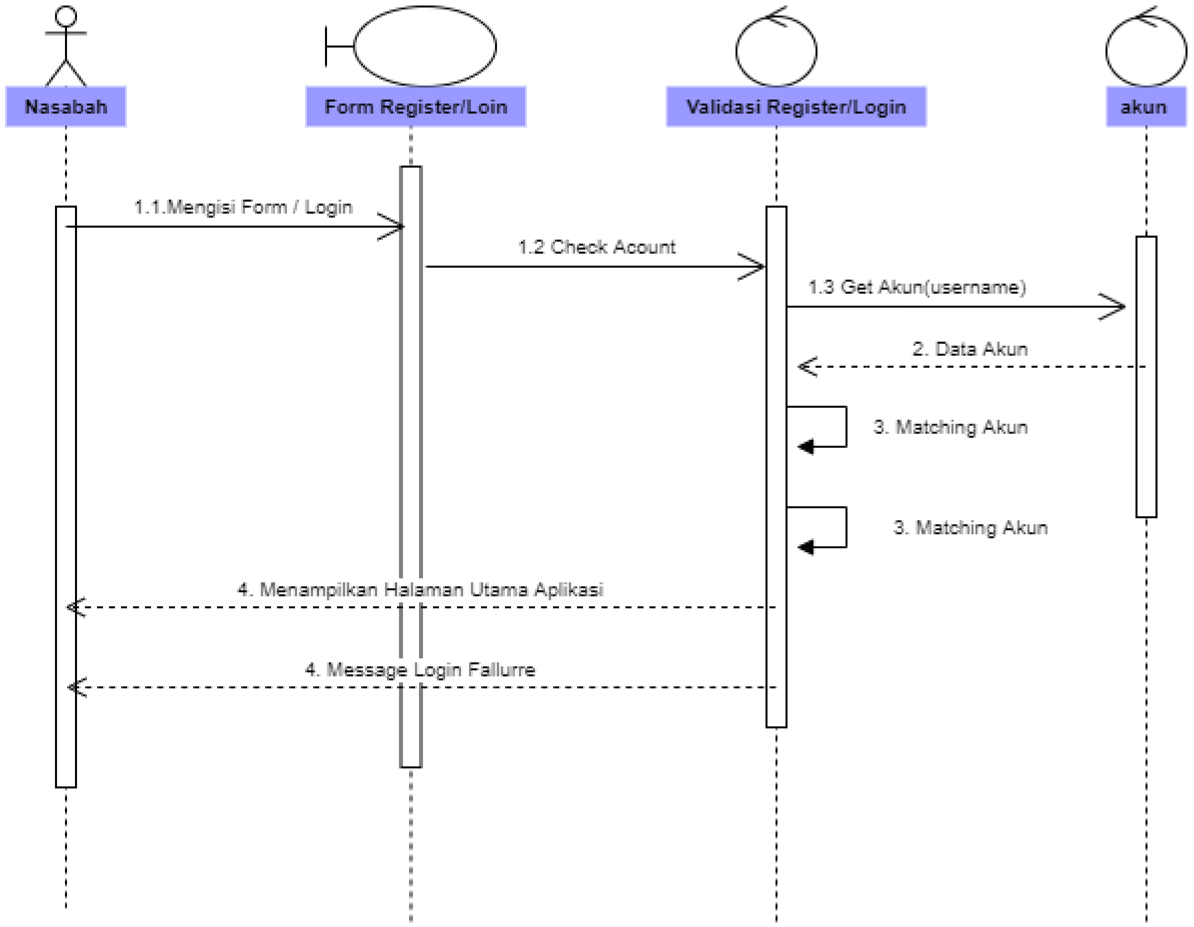
Aliza Rizka Firdani - 1301190297

Wina Munawaroh - 1301194036

Sinta Nur Maulina - 1301194135

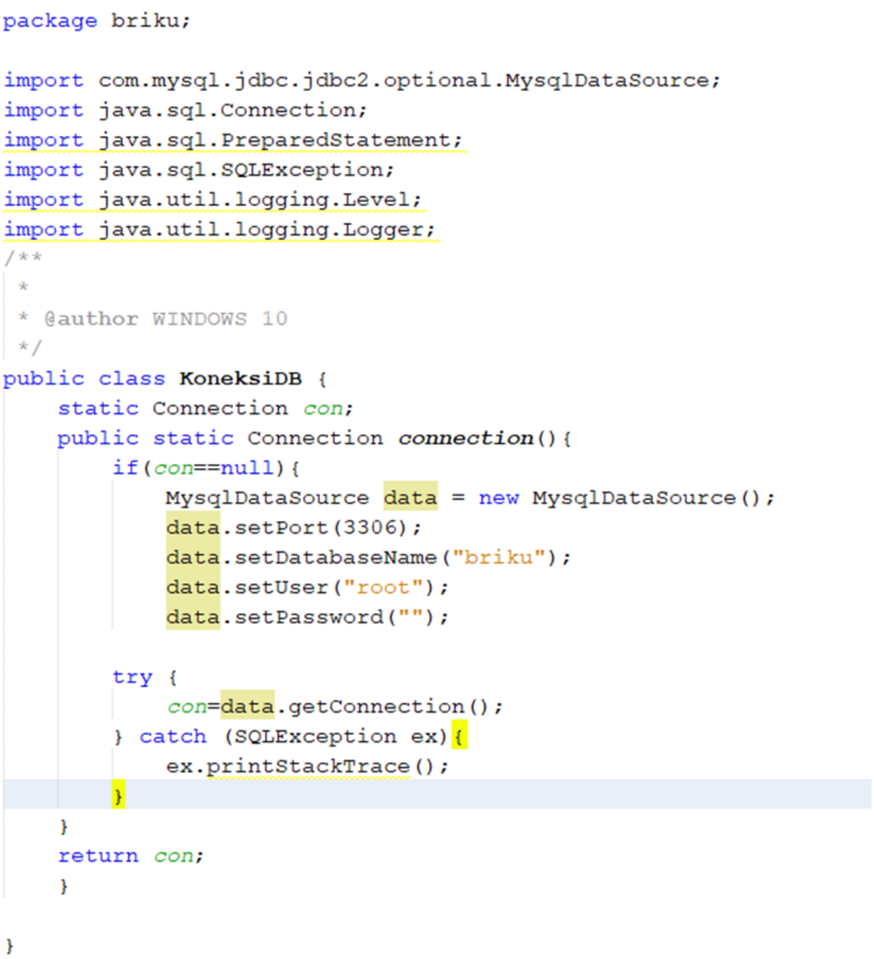
TP

Lakukan Analisis Robustness (Menyempurnakan Sequence Diagram) terhadap use case tersebut. Dan laporkan hasilnya dalam dokumen DPPL, sebelum masuk Lab

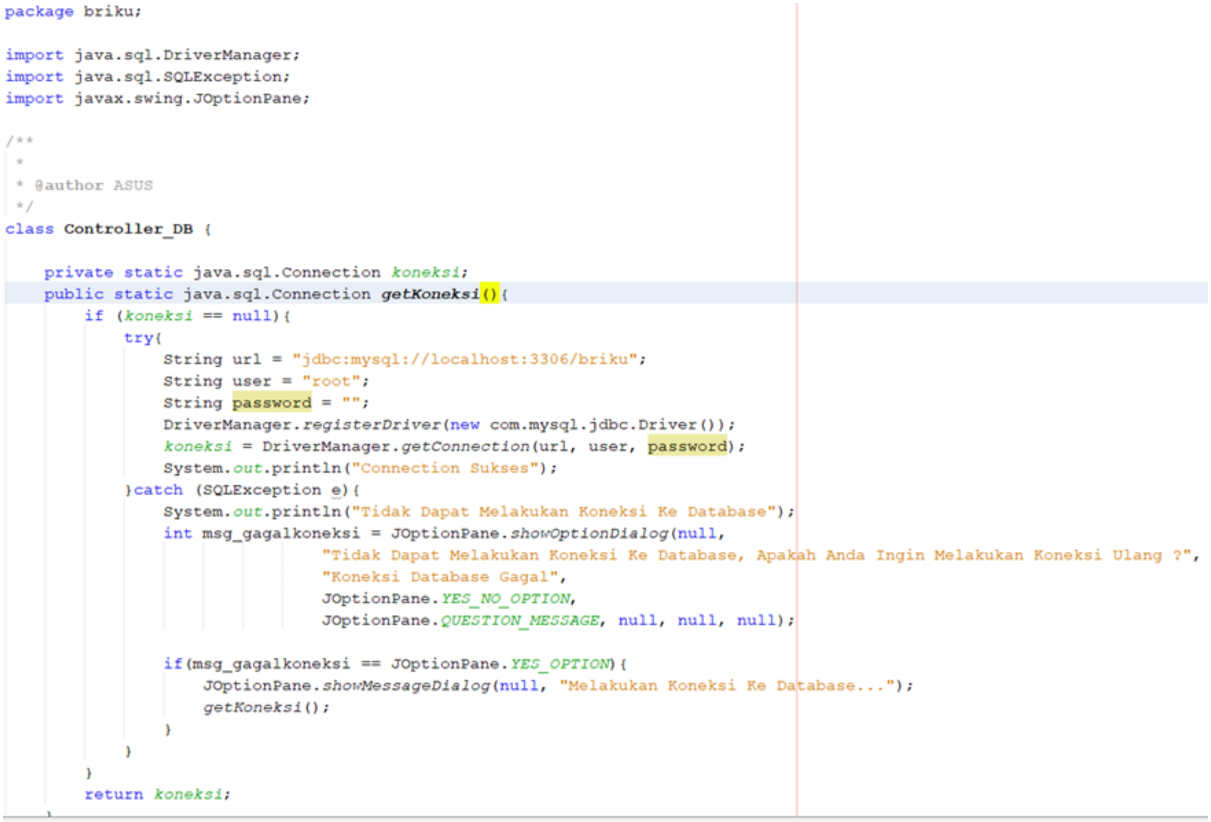


JURNAL :

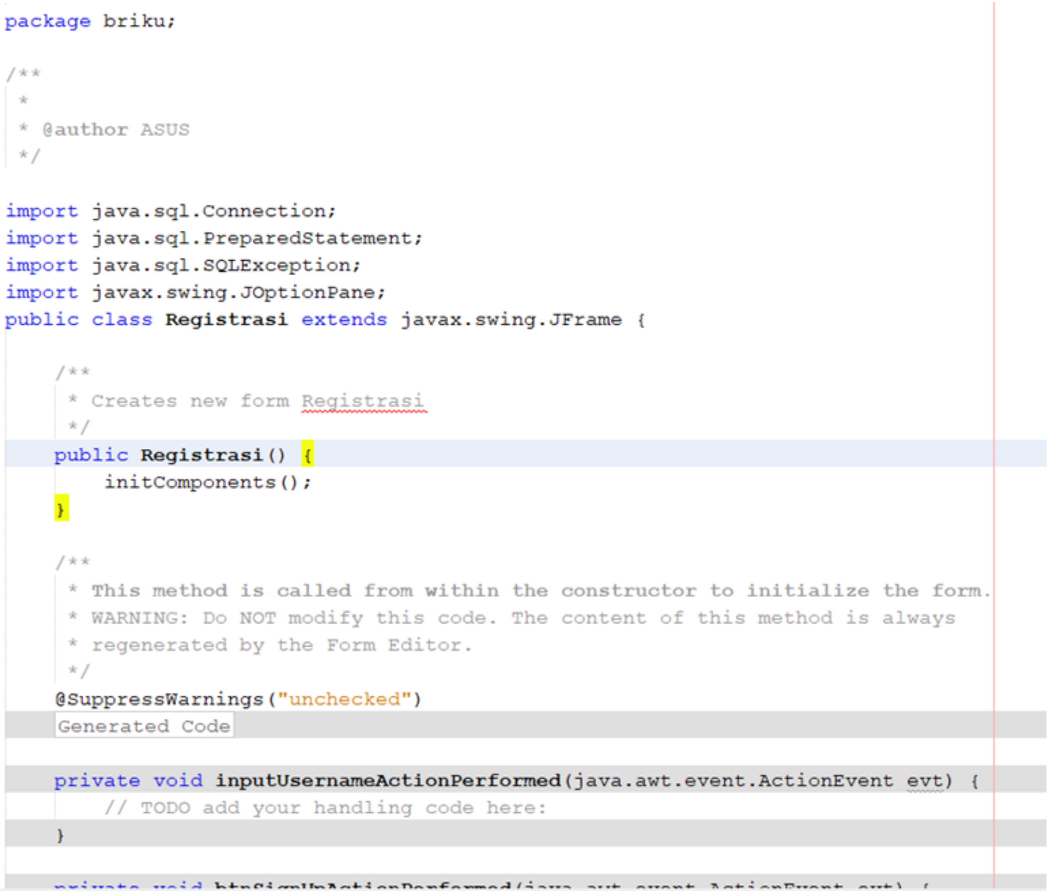
Implementasikan hasil Analisis Robustness Anda dalam Pemrograman Berorientasi Objek :



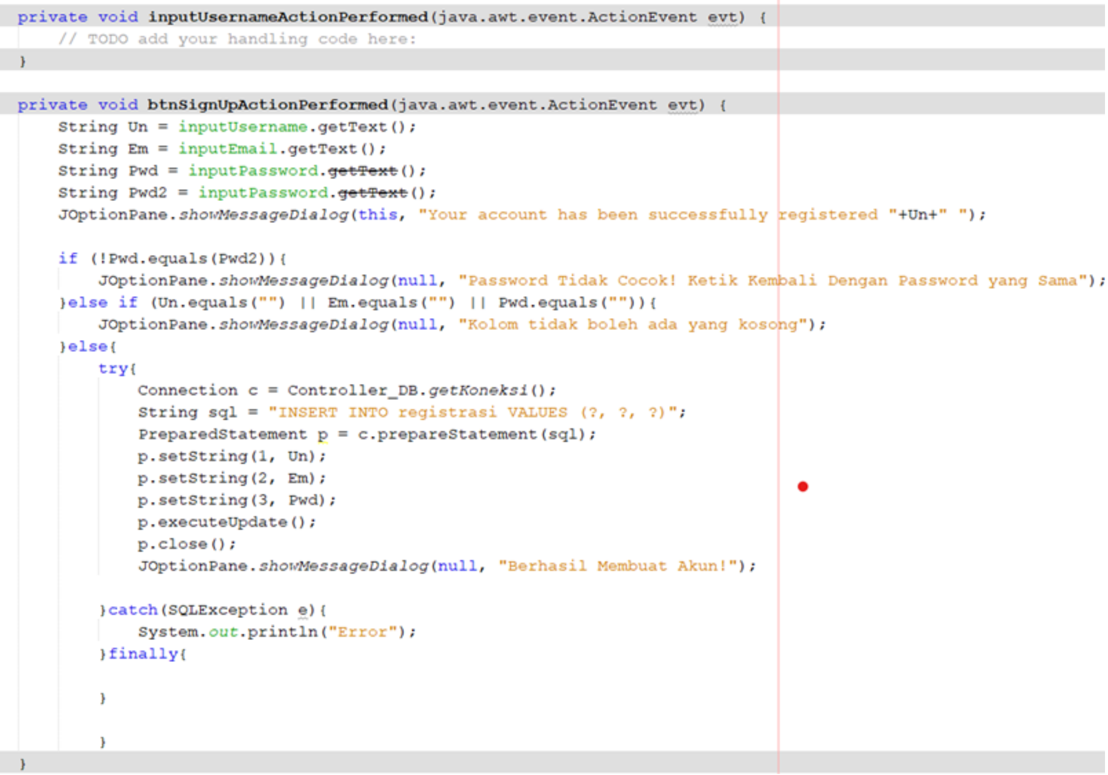
```
package briku;

import com.mysql.jdbc.jdbc2.optional.MysqlDataSource;
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
/**
 *
 * @author WINDOWS 10
 */
public class KoneksiDB {
    static Connection con;
    public static Connection connection(){
        if(con==null){
            MysqlDataSource data = new MysqlDataSource();
            data.setPort(3306);
            data.setDatabaseName("briku");
            data.setUser("root");
            data.setPassword("");

        try {
            con=data.getConnection();
        } catch (SQLException ex){
            ex.printStackTrace();
        }
    }
    return con;
    }

}
```



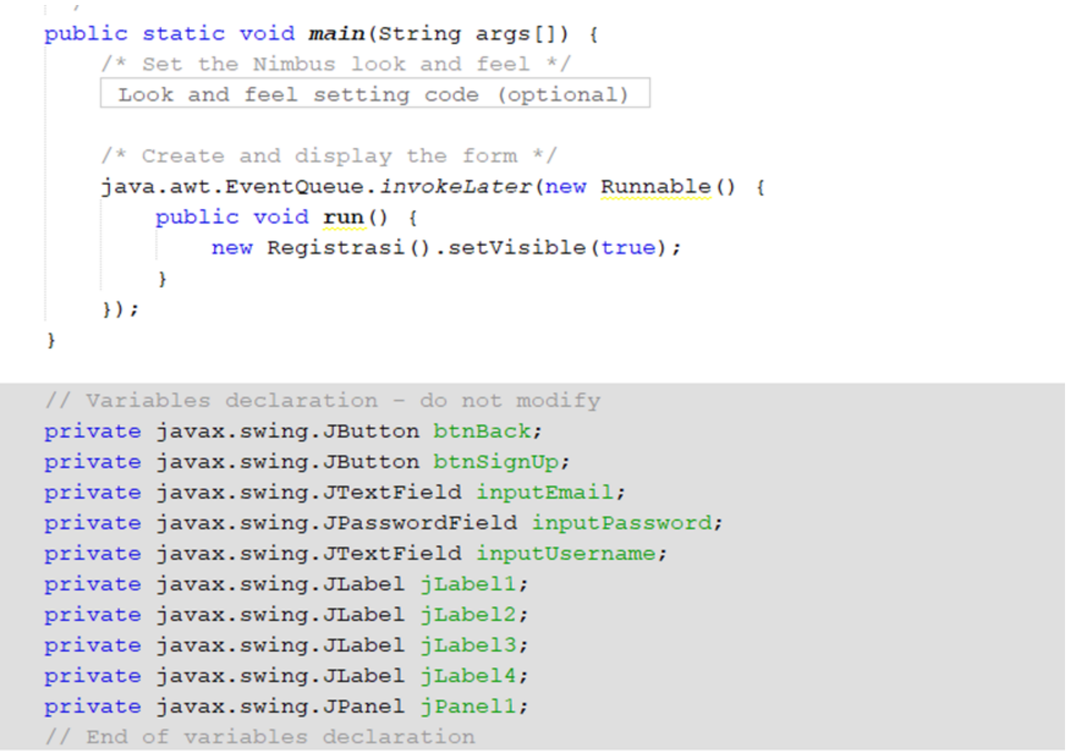
```
package briku;

import java.sql.DriverManager;
import java.sql.SQLException;
import javax.swing.JOptionPane;

/**
 *
 * @author ASUS
 */
class Controller_DB {

    private static java.sql.Connection koneksi;
    public static java.sql.Connection getKoneksi(){
        if (koneksi == null){
            try{
                String url = "jdbc:mysql://localhost:3306/briku";
                String user = "root";
                String password = "";
                DriverManager.registerDriver(new com.mysql.jdbc.Driver());
                koneksi = DriverManager.getConnection(url, user, password);
                System.out.println("Connection Sukses");
            }catch (SQLException e){
                System.out.println("Tidak Dapat Melakukan Koneksi Ke Database");
                int msg_gagalkoneksi = JOptionPane.showOptionDialog(null,
                                "Tidak Dapat Melakukan Koneksi Ke Database, Apakah Anda Ingin Melakukan Koneksi Ulang ?",
                                "Koneksi Database Gagal",
                                JOptionPane.YES_NO_OPTION,
                                JOptionPane.QUESTION_MESSAGE, null, null, null);

                if(msg_gagalkoneksi == JOptionPane.YES_OPTION){
                    JOptionPane.showMessageDialog(null, "Melakukan Koneksi Ke Database...");
                    getKoneksi();
                }
            }
        }
        return koneksi;
```



```java
package briku;

/**
 *
 * @author ASUS
 */

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
public class Registrasi extends javax.swing.JFrame {

    /**
     * Creates new form Registrasi
     */
    public Registrasi() {
        initComponents();
    }

    /**
     * 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")
    Generated Code

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

    private void btnSignUpActionPerformed(java.awt.event.ActionEvent evt) {
```



```java
private void inputUsernameActionPerformed(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
}

private void btnSignUpActionPerformed(java.awt.event.ActionEvent evt) {
    String Un = inputUsername.getText();
    String Em = inputEmail.getText();
    String Pwd = inputPassword.getText();
    String Pwd2 = inputPassword.getText();
    JOptionPane.showMessageDialog(this, "Your account has been successfully registered "+Un+" ");

    if (!Pwd.equals(Pwd2)){
        JOptionPane.showMessageDialog(null, "Password Tidak Cocok! Ketik Kembali Dengan Password yang Sama");
    }else if (Un.equals("") || Em.equals("") || Pwd.equals("")){
        JOptionPane.showMessageDialog(null, "Kolom tidak boleh ada yang kosong");
    }else{
        try{
            Connection c = Controller_DB.getKoneksi();
            String sql = "INSERT INTO registrasi VALUES (?, ?, ?)";
            PreparedStatement p = c.prepareStatement(sql);
            p.setString(1, Un);
            p.setString(2, Em);
            p.setString(3, Pwd);
            p.executeUpdate();
            p.close();
            JOptionPane.showMessageDialog(null, "Berhasil Membuat Akun!");

        }catch(SQLException e){
            System.out.println("Error");
        }finally{

        }

    }
}
```



```
public static void main(String args[]) {
    /* Set the Nimbus look and feel */
    Look and feel setting code (optional)

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

// Variables declaration - do not modify
private javax.swing.JButton btnBack;
private javax.swing.JButton btnSignUp;
private javax.swing.JTextField inputEmail;
private javax.swing.JPasswordField inputPassword;
private javax.swing.JTextField inputUsername;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JPanel jPanel1;
// End of variables declaration
```