

**REPORT MANUAL OF JOBSHEET**

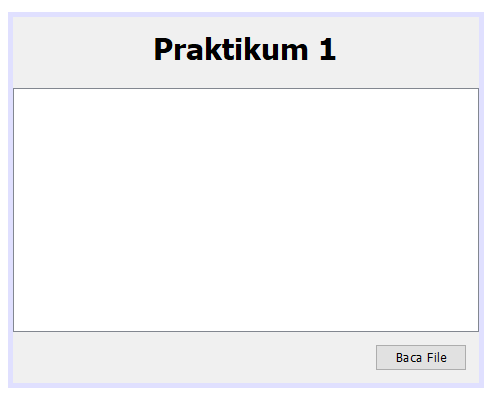
**Practicum, Tasks and Questions**

**(Network Programming)**

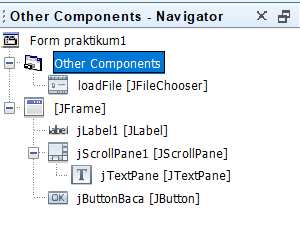
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Name** | **:** | **Brian Sayudha** |  |
|  | **Class / NIM** | **:** | **3G / 1841720158** |  |
|  | **Major** | **:** | **D-IV Informatics Enginering** |  |
|  |  |  |  |  |
|  |  |  |  |  |

Praktikum 1

Design



Navigator



Code

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package Praktikum1;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JTextPane;

/\*\*

 \*

 \* @author Asus

 \*/

public class praktikum1 extends javax.swing.*JFrame* {

    /\*\*

     \* Creates new form praktikum1

     \*/

     private praktikum1Controller controller;

     public praktikum1() {

         initComponents();

         controller = new praktikum1Controller(*this*);

     }

    public *JButton* getjButtonBaca() {

        return jButtonBaca;

    }

    public *JTextPane* getjTextPane() {

        return jTextPane;

    }

    public *JFileChooser* getLoadFile() {

        return loadFile;

    }

    /\*\*

     \* 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() {

        loadFile = new javax.swing.*JFileChooser*();

        jLabel1 = new javax.swing.*JLabel*();

        jScrollPane1 = new javax.swing.*JScrollPane*();

        jTextPane = new javax.swing.*JTextPane*();

        jButtonBaca = new javax.swing.*JButton*();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

        jLabel1.setFont(new java.awt.*Font*("Tahoma", 1, 24)); // NOI18N

        jLabel1.setText("Praktikum 1");

        jScrollPane1.setViewportView(jTextPane);

        jButtonBaca.setText("Baca File");

        jButtonBaca.addActionListener(new java.awt.event.*ActionListener*() {

            public void actionPerformed(java.awt.event.*ActionEvent* evt) {

                jButtonBacaActionPerformed(evt);

            }

        });

        javax.swing.*GroupLayout* layout = new javax.swing.*GroupLayout*(getContentPane());

        getContentPane().setLayout(layout);

        layout.setHorizontalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

                .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

                .addComponent(jButtonBaca)

                .addContainerGap())

            .addGroup(layout.createSequentialGroup()

                .addGap(112, 112, 112)

                .addComponent(jLabel1)

                .addContainerGap(114, Short.MAX\_VALUE))

        );

        layout.setVerticalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel1)

                .addGap(18, 18, 18)

                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 195, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

                .addComponent(jButtonBaca)

                .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

        );

        pack();

    }// </editor-fold>

    private void jButtonBacaActionPerformed(java.awt.event.*ActionEvent* evt) {

        // TODO add your handling code here:

    }

    /\*\*

     \* @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.html

         \*/

        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(praktikum1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(praktikum1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(praktikum1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(praktikum1.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }

        //</editor-fold>

        /\* Create and display the form \*/

        java.awt.EventQueue.invokeLater(new *Runnable*() {

            public void run() {

                new praktikum1().setVisible(true);

            }

        });

    }

    // Variables declaration - do not modify

    private javax.swing.*JButton* jButtonBaca;

    private javax.swing.*JLabel* jLabel1;

    private javax.swing.*JScrollPane* jScrollPane1;

    private javax.swing.*JTextPane* jTextPane;

    private javax.swing.*JFileChooser* loadFile;

    // End of variables declaration

}

Code controller

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package Praktikum1;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.InputStream;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.text.BadLocationException;

import javax.swing.text.StyledDocument;

public class praktikum1Controller {

     private praktikum1 view;

     public praktikum1Controller(praktikum1 view) {

*this*.view = view;

*this*.view.getjButtonBaca().addActionListener(new *ActionListener*() {

             @Override

             public void actionPerformed(*ActionEvent* e) {

                 proses();

             }

         });

     }

     private void proses(){

*JFileChooser* loadFile = view.getLoadFile();

*StyledDocument* doc = view.getjTextPane().getStyledDocument();

         if (JFileChooser.APPROVE\_OPTION == loadFile.showOpenDialog(view)) {

*InputStream* inputStream = null;

             try {

                 inputStream = new *FileInputStream*(loadFile.getSelectedFile());

                 int read = inputStream.read();

                 doc.insertString(0, "", null);

                 while (read != -1) {

                     doc.insertString(doc.getLength(), "" + read, null);

                     System.out.println("" + read);

                     read = inputStream.read();

                 }

             } catch (FileNotFoundException ex) {

                 Logger.getLogger(praktikum1Controller.class.getName()).log(Level.SEVERE, null, ex);

             } catch (IOException | BadLocationException ex) {

                 Logger.getLogger(praktikum1Controller.class.getName()).log(Level.SEVERE, null, ex);

             } finally {

                 if (inputStream != null) {

                     try {

                         inputStream.close();

                     } catch (IOException ex) {

                         Logger.getLogger(praktikum1Controller.class.getName()).log(Level.SEVERE, null, ex);

                     }

                 }

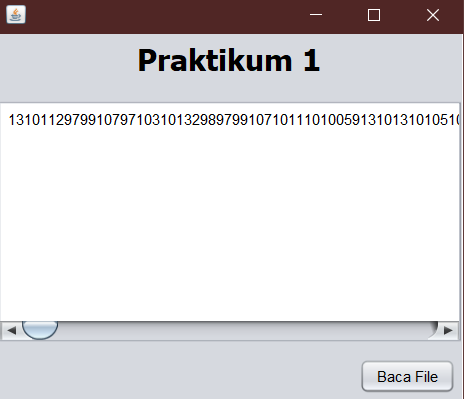
             }

         }

     }

}

Result



Pertanyaan prak1

1. Mengapa menggunakan sintak while (read != -1) untuk melakukan pembacaan?

Jawaban: jika byte read mencapain -1 maka stream telah berakhir

1. Seberapa penting perintah inputStream.close(); dipanggil setelah selesai membaca file?

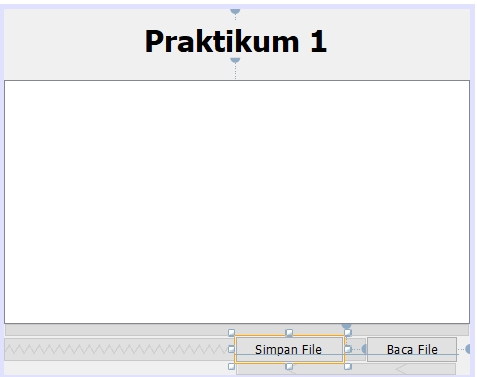
Jawaban : Sangat penting, karena untuk menghentikan alur pembacaan data

1. Mengapa isi file yang dibaca dengan yang ditampilkan berbeda?

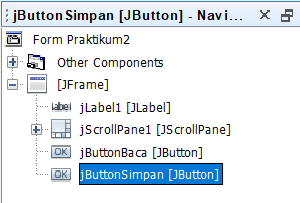
Jawaban : karena kita meminta untuk menampilkan byte dari data yang kita insert

Praktikum 2

Design



Navigator



Code

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package Praktikum2;

import Praktikum1.\*;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JTextPane;

/\*\*

 \*

 \* @author Asus

 \*/

public class *Praktikum2* extends javax.swing.*JFrame* {

    /\*\*

     \* Creates new form Praktikum2

     \*/

     private *Praktikum2Controller* controller;

     public Praktikum2() {

         initComponents();

         controller = new *Praktikum2Controller*(*this*);

     }

    public *JButton* getjButtonBaca() {

        return jButtonBaca;

    }

    public *JTextPane* getjTextPane() {

        return jTextPane;

    }

    public *JFileChooser* getLoadFile() {

        return loadFile;

    }

    public *JButton* getjButtonSimpan() {

        return jButtonSimpan;

    }

    /\*\*

     \* 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() {

        loadFile = new javax.swing.*JFileChooser*();

        jLabel1 = new javax.swing.*JLabel*();

        jScrollPane1 = new javax.swing.*JScrollPane*();

        jTextPane = new javax.swing.*JTextPane*();

        jButtonBaca = new javax.swing.*JButton*();

        jButtonSimpan = new javax.swing.*JButton*();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

        jLabel1.setFont(new java.awt.*Font*("Tahoma", 1, 24)); // NOI18N

        jLabel1.setText("Praktikum 1");

        jScrollPane1.setViewportView(jTextPane);

        jButtonBaca.setText("Baca File");

        jButtonBaca.addActionListener(new java.awt.event.*ActionListener*() {

            public void actionPerformed(java.awt.event.*ActionEvent* evt) {

                jButtonBacaActionPerformed(evt);

            }

        });

        jButtonSimpan.setText("Simpan File");

        jButtonSimpan.addActionListener(new java.awt.event.*ActionListener*() {

            public void actionPerformed(java.awt.event.*ActionEvent* evt) {

                jButtonSimpanActionPerformed(evt);

            }

        });

        javax.swing.*GroupLayout* layout = new javax.swing.*GroupLayout*(getContentPane());

        getContentPane().setLayout(layout);

        layout.setHorizontalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

                .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

                .addComponent(jButtonSimpan)

                .addGap(18, 18, 18)

                .addComponent(jButtonBaca)

                .addContainerGap())

            .addGroup(layout.createSequentialGroup()

                .addGap(112, 112, 112)

                .addComponent(jLabel1)

                .addContainerGap(114, Short.MAX\_VALUE))

        );

        layout.setVerticalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel1)

                .addGap(18, 18, 18)

                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 195, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(jButtonBaca)

                    .addComponent(jButtonSimpan))

                .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

        );

        pack();

    }// </editor-fold>

    private void jButtonBacaActionPerformed(java.awt.event.*ActionEvent* evt) {

        // TODO add your handling code here:

    }

    private void jButtonSimpanActionPerformed(java.awt.event.*ActionEvent* evt) {

        // TODO add your handling code here:

    }

    /\*\*

     \* @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.html

         \*/

        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(Praktikum2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Praktikum2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Praktikum2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Praktikum2.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        /\* Create and display the form \*/

        java.awt.EventQueue.invokeLater(new *Runnable*() {

            public void run() {

                new *Praktikum2*().setVisible(true);

            }

        });

    }

    // Variables declaration - do not modify

    private javax.swing.*JButton* jButtonBaca;

    private javax.swing.*JButton* jButtonSimpan;

    private javax.swing.*JLabel* jLabel1;

    private javax.swing.*JScrollPane* jScrollPane1;

    private javax.swing.*JTextPane* jTextPane;

    private javax.swing.*JFileChooser* loadFile;

    // End of variables declaration

}

Code Controller

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package Praktikum2;

import Praktikum1.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.text.BadLocationException;

import javax.swing.text.StyledDocument;

public class *Praktikum2Controller* {

    private *Praktikum2* view;

    private *List*<*Integer*> list = new *ArrayList*<>();

    public Praktikum2Controller(*Praktikum2* view) {

*this*.view = view;

*this*.view.getjButtonBaca().addActionListener(new *ActionListener*() {

            @Override

            public void actionPerformed(*ActionEvent* e) {

                proses();

            }

        });

*this*.view.getjButtonSimpan().addActionListener(new *ActionListener*() {

             @Override

             public void actionPerformed(*ActionEvent* e) {

                 save();

             }

         });

    }

    private void proses() {

*JFileChooser* loadFile = view.getLoadFile();

*StyledDocument* doc = view.getjTextPane().getStyledDocument();

        if (JFileChooser.APPROVE\_OPTION == loadFile.showOpenDialog(view)) {

*InputStream* inputStream = null;

            try {

                inputStream = new *FileInputStream*(loadFile.getSelectedFile());

                int read = inputStream.read();

                doc.insertString(0, "", null);

                while (read != -1) {

                    doc.insertString(doc.getLength(), "" + read, null);

                    System.out.println("" + read);

                    read = inputStream.read();

                }

                while (read != -1) {

                    list.add(read);// tambahkan 1 baris

                    doc.insertString(doc.getLength(), "" + read, null);

                    System.out.println("" + read);

                    read = inputStream.read();

                }

            } catch (FileNotFoundException ex) {

                Logger.getLogger(Praktikum2Controller.class.getName()).log(Level.SEVERE, null, ex);

            } catch (IOException | BadLocationException ex) {

                Logger.getLogger(Praktikum2Controller.class.getName()).log(Level.SEVERE, null, ex);

            } finally {

                if (inputStream != null) {

                    try {

                        inputStream.close();

                    } catch (IOException ex) {

                        Logger.getLogger(Praktikum2Controller.class.getName()).log(Level.SEVERE, null, ex);

                    }

                }

            }

        }

    }

    private void save() {

*JFileChooser* loadFile = view.getLoadFile();

        if (JFileChooser.APPROVE\_OPTION == loadFile.showSaveDialog(view)) {

*OutputStream* os = null;

            try {

                if (!list.isEmpty()) {

                    os = new *FileOutputStream*(loadFile.getSelectedFile());

                    byte[] dt = new byte[list.size()];

                    for (int i = 0; i < list.size(); i++) {

                        dt[i] = list.get(i).byteValue();

                    }

                    os.write(dt);

                }

            } catch (FileNotFoundException ex) {

                Logger.getLogger(Praktikum2Controller.class.getName()).log(Level.SEVERE, null, ex);

            } catch (IOException ex) {

                Logger.getLogger(Praktikum2Controller.class.getName()).log(Level.SEVERE, null, ex);

            } finally {

                if (os != null) {

                    try {

                        os.flush();

                        os.close();

                        list.clear();

                    } catch (IOException ex) {

                        Logger.getLogger(Praktikum2Controller.class.getName()).log(Level.SEVERE, null, ex);

                    }

                }

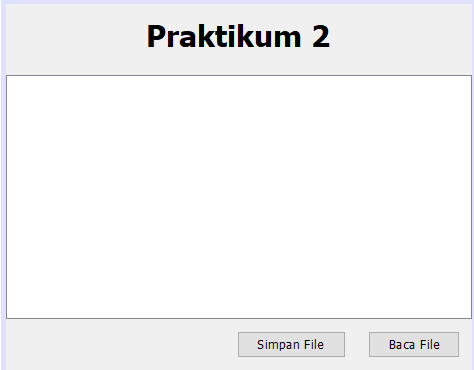
            }

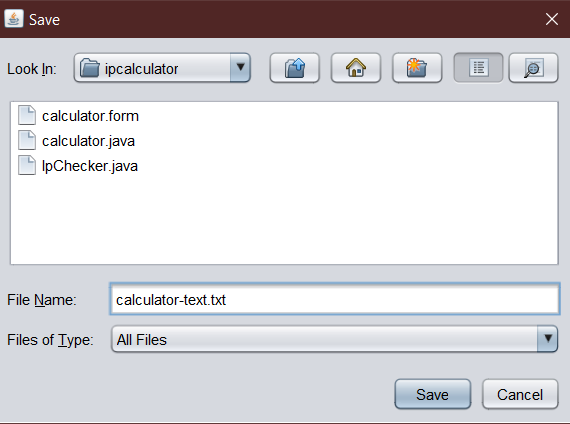
        }

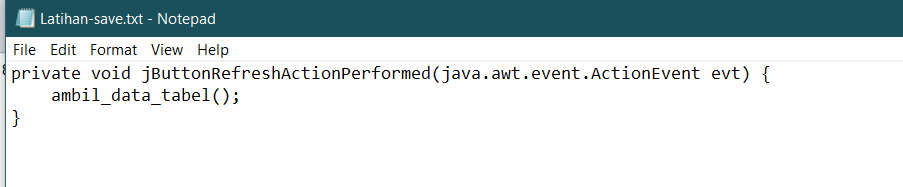
    }

}

Result







Pertanyaan prak2

1. Mengapa OutputStream os = null; di-instance bukan menggunakan new OutputStream, tetapi os = new FileOutputStream(loadFile.getSelectedFile());?

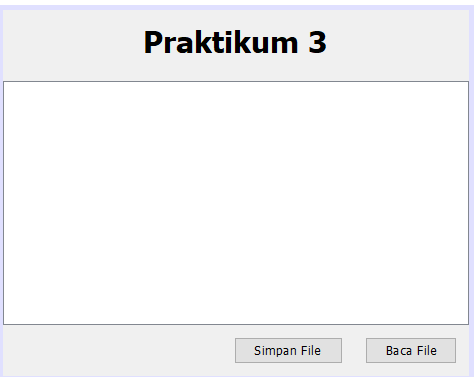
Jawaban : Karena outpout stream sudah diberi alias os

1. Hilangkan os.flush(); dan os.close();, apa yang terjadi? Mengapa demikian?

Jawaban : Tidak terjadi apa apa, karena hanya untuk mengoptimalkan kerja dari sistem

Praktikum 3

Design



Code

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package Praktikum3;

import Praktikum2.\*;

import Praktikum1.\*;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JTextPane;

/\*\*

 \*

 \* @author Asus

 \*/

public class *Praktikum3* extends javax.swing.*JFrame* {

    /\*\*

     \* Creates new form Praktikum3

     \*/

     private *Praktikum3Controller* controller;

     public Praktikum3() {

         initComponents();

         controller = new *Praktikum3Controller*(*this*);

     }

    public *JButton* getjButtonBaca() {

        return jButtonBaca;

    }

    public *JTextPane* getjTextPane() {

        return jTextPane;

    }

    public *JFileChooser* getLoadFile() {

        return loadFile;

    }

    public *JButton* getjButtonSimpan() {

        return jButtonSimpan;

    }

    /\*\*

     \* 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() {

        loadFile = new javax.swing.*JFileChooser*();

        jLabel1 = new javax.swing.*JLabel*();

        jScrollPane1 = new javax.swing.*JScrollPane*();

        jTextPane = new javax.swing.*JTextPane*();

        jButtonBaca = new javax.swing.*JButton*();

        jButtonSimpan = new javax.swing.*JButton*();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

        jLabel1.setFont(new java.awt.*Font*("Tahoma", 1, 24)); // NOI18N

        jLabel1.setText("Praktikum 3");

        jScrollPane1.setViewportView(jTextPane);

        jButtonBaca.setText("Baca File");

        jButtonBaca.addActionListener(new java.awt.event.*ActionListener*() {

            public void actionPerformed(java.awt.event.*ActionEvent* evt) {

                jButtonBacaActionPerformed(evt);

            }

        });

        jButtonSimpan.setText("Simpan File");

        jButtonSimpan.addActionListener(new java.awt.event.*ActionListener*() {

            public void actionPerformed(java.awt.event.*ActionEvent* evt) {

                jButtonSimpanActionPerformed(evt);

            }

        });

        javax.swing.*GroupLayout* layout = new javax.swing.*GroupLayout*(getContentPane());

        getContentPane().setLayout(layout);

        layout.setHorizontalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING)

            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

                .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

                .addComponent(jButtonSimpan)

                .addGap(18, 18, 18)

                .addComponent(jButtonBaca)

                .addContainerGap())

            .addGroup(layout.createSequentialGroup()

                .addGap(112, 112, 112)

                .addComponent(jLabel1)

                .addContainerGap(114, Short.MAX\_VALUE))

        );

        layout.setVerticalGroup(

            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

            .addGroup(layout.createSequentialGroup()

                .addContainerGap()

                .addComponent(jLabel1)

                .addGap(18, 18, 18)

                .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 195, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

                    .addComponent(jButtonBaca)

                    .addComponent(jButtonSimpan))

                .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

        );

        pack();

    }// </editor-fold>

    private void jButtonBacaActionPerformed(java.awt.event.*ActionEvent* evt) {

        // TODO add your handling code here:

    }

    private void jButtonSimpanActionPerformed(java.awt.event.*ActionEvent* evt) {

        // TODO add your handling code here:

    }

    /\*\*

     \* @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.html

         \*/

        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(Praktikum3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (InstantiationException ex) {

            java.util.logging.Logger.getLogger(Praktikum3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (IllegalAccessException ex) {

            java.util.logging.Logger.getLogger(Praktikum3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        } catch (javax.swing.UnsupportedLookAndFeelException ex) {

            java.util.logging.Logger.getLogger(Praktikum3.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

        }

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        //</editor-fold>

        /\* Create and display the form \*/

        java.awt.EventQueue.invokeLater(new *Runnable*() {

            public void run() {

                new *Praktikum3*().setVisible(true);

            }

        });

    }

    // Variables declaration - do not modify

    private javax.swing.*JButton* jButtonBaca;

    private javax.swing.*JButton* jButtonSimpan;

    private javax.swing.*JLabel* jLabel1;

    private javax.swing.*JScrollPane* jScrollPane1;

    private javax.swing.*JTextPane* jTextPane;

    private javax.swing.*JFileChooser* loadFile;

    // End of variables declaration

}

Code controller

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package Praktikum3;

import Praktikum2.\*;

import Praktikum1.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.io.InputStream;

import java.io.OutputStream;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JFileChooser;

import javax.swing.text.BadLocationException;

import javax.swing.text.StyledDocument;

public class *Praktikum3Controller* {

    private *Praktikum3* view;

    private *List*<*Integer*> list = new *ArrayList*<>();

    public Praktikum3Controller(*Praktikum3* view) {

*this*.view = view;

*this*.view.getjButtonBaca().addActionListener(new *ActionListener*() {

            @Override

            public void actionPerformed(*ActionEvent* e) {

                proses();

            }

        });

*this*.view.getjButtonSimpan().addActionListener(new *ActionListener*() {

             @Override

             public void actionPerformed(*ActionEvent* e) {

                 save();

             }

         });

    }

     private void proses() {

*JFileChooser* loadFile = view.getLoadFile();

*StyledDocument* doc = view.getjTextPane().getStyledDocument();

         if (JFileChooser.APPROVE\_OPTION == loadFile.showOpenDialog(view)) {

*BufferedReader* reader = null;

             try {

                 reader = new *BufferedReader*(new *FileReader*(loadFile.getSelectedFile()));

*String* data = null;

                 doc.insertString(0, "", null);

                 while ((data = reader.readLine()) != null) {

                     doc.insertString(doc.getLength(), data, null);

                 }

             } catch (FileNotFoundException ex) {

                 Logger.getLogger(Praktikum3Controller.class.getName()).log(Level.SEVERE, null, ex);

             } catch (IOException | BadLocationException ex) {

                 Logger.getLogger(Praktikum3Controller.class.getName()).log(Level.SEVERE, null, ex);

             } finally {

                 if (reader != null) {

                     try {

                         reader.close();

                     } catch (IOException ex) {

                         Logger.getLogger(Praktikum3Controller.class.getName()).log(Level.SEVERE, null, ex);

                     }

                 }

             }

         }

     }

     private void save() {

*JFileChooser* loadFile = view.getLoadFile();

         if (JFileChooser.APPROVE\_OPTION == loadFile.showSaveDialog(view)) {

*BufferedWriter* writer = null;

             try {

*String* contents = view.getjTextPane().getText();

                 if (contents != null && !contents.isEmpty()) {

                     writer = new *BufferedWriter*(new *FileWriter*(loadFile.getSelectedFile()));

                     writer.write(contents);

                 }

             } catch (FileNotFoundException ex) {

                 Logger.getLogger(Praktikum3Controller.class.getName()).log(Level.SEVERE, null, ex);

             } catch (IOException ex) {

                 Logger.getLogger(Praktikum3Controller.class.getName()).log(Level.SEVERE, null, ex);

             } finally {

                 if (writer != null) {

                     try {

                         writer.flush();

                         writer.close();

                         view.getjTextPane().setText("");

                     } catch (IOException ex) {

                         Logger.getLogger(Praktikum3Controller.class.getName()).log(Level.SEVERE, null, ex);

                     }

                 }

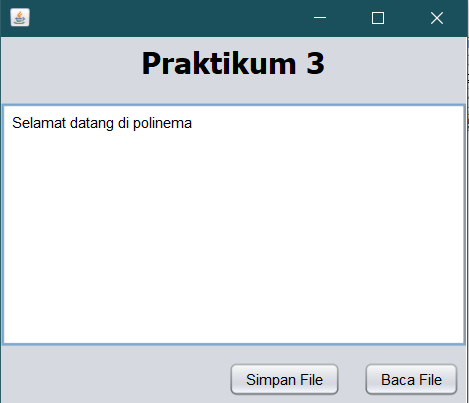
             }

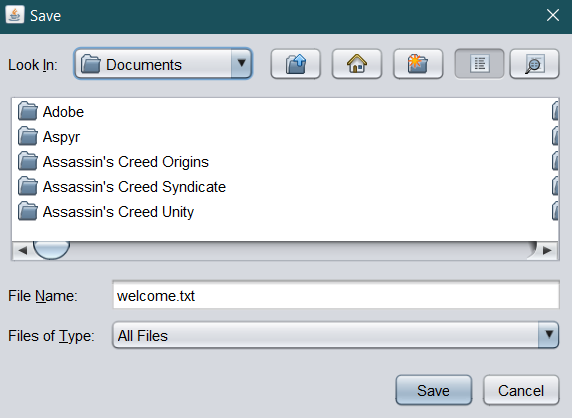
         }

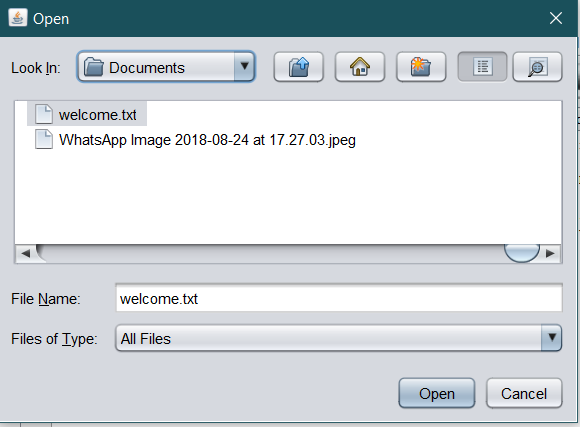
     }

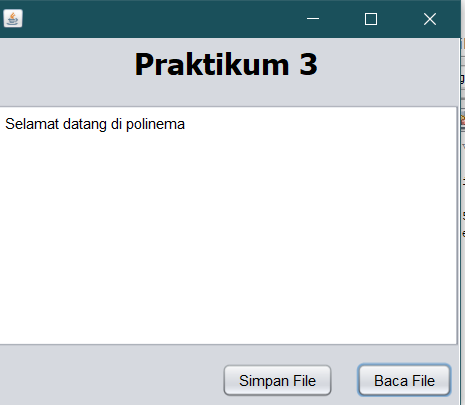
}

Result









Pertanyaan 3

1. Silakan inputkan beberapa baris kalimat, selanjutnya simpan dan dibaca kembali? apa yang terjadi?

Jawaban : sama seperti yang kita simpan

1. Jika yang disimpan dengan yang dibaca tidak sesuai, silakan disesuaikan!

Jawaban : Sudah sesuai

Tugas

1. Pada **praktikum 1** data yang dibaca masih berupa angka-angka, silakan disesuaikan sehingga data yang dituliskan sesuai dengan yang dibaca!

private void proses(){

    JFileChooser loadFile = view.getLoadFile();

    StyledDocument doc = view.getjTextPane().getStyledDocument();

    if (JFileChooser.APPROVE\_OPTION == loadFile.showOpenDialog(view)) {

        InputStream inputStream = null;

        try {

            inputStream = new FileInputStream(loadFile.getSelectedFile());

            InputStreamReader reader = new InputStreamReader(inputStream);

            BufferedReader br= new BufferedReader(reader);

            StringBuilder sb=new StringBuilder();

            String str;

            while ((str = br.readLine())!=null) {

                sb.append(str);

            }

            doc.insertString(0, sb.toString(), null);

        } catch (FileNotFoundException ex) {

            Logger.getLogger(praktikum1Controller.class.getName()).log(Level.SEVERE, null, ex);

        } catch (IOException | BadLocationException ex) {

            Logger.getLogger(praktikum1Controller.class.getName()).log(Level.SEVERE, null, ex);

        } finally {

            if (inputStream != null) {

                try {

                    inputStream.close();

                } catch (IOException ex) {

                    Logger.getLogger(praktikum1Controller.class.getName()).log(Level.SEVERE, null, ex);

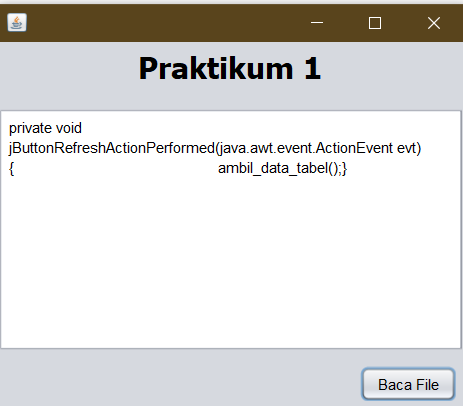
                }

            }

        }

    }

}



1. Pada **praktikum 3**, ketika diinputkan kalimat beberapa baris end of line tidak berfungsi. Perbaiki hal tersebut agar end of line bisa berfungsi dengan baik!

try {

    reader = new BufferedReader(new FileReader(loadFile.getSelectedFile()));

    String data = null;

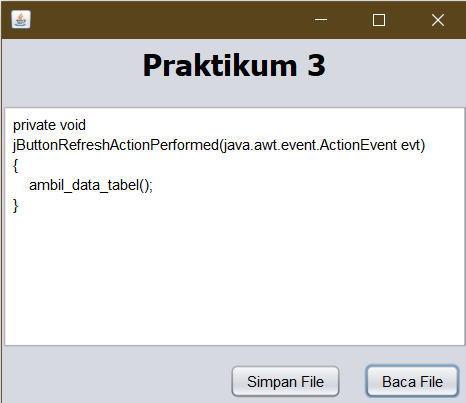
    doc.insertString(0, "", null);

    while ((data = reader.readLine()) != null) {

        doc.insertString(doc.getLength(), data+"\n", null);

    }

Design



1. Tambahkan pesan dialog mengunakan JoptionPane ketika operasi sudah selesai atau terjadi error, misalkan operasi baca data atau menulis!

