LAPORAN PROJECT

PEMROGRAMAN BERORIENTASI OBJEK

Dosen Pembimbing :

SLAMET TRIYANTO, S. ST

Disusun Oleh :

MUHAMMAD FAUZAN

NIM : 202013012

PROGRAM STUDI TEKNIK INFORMATIKA

POLITEKNIK KAMPAR

2021

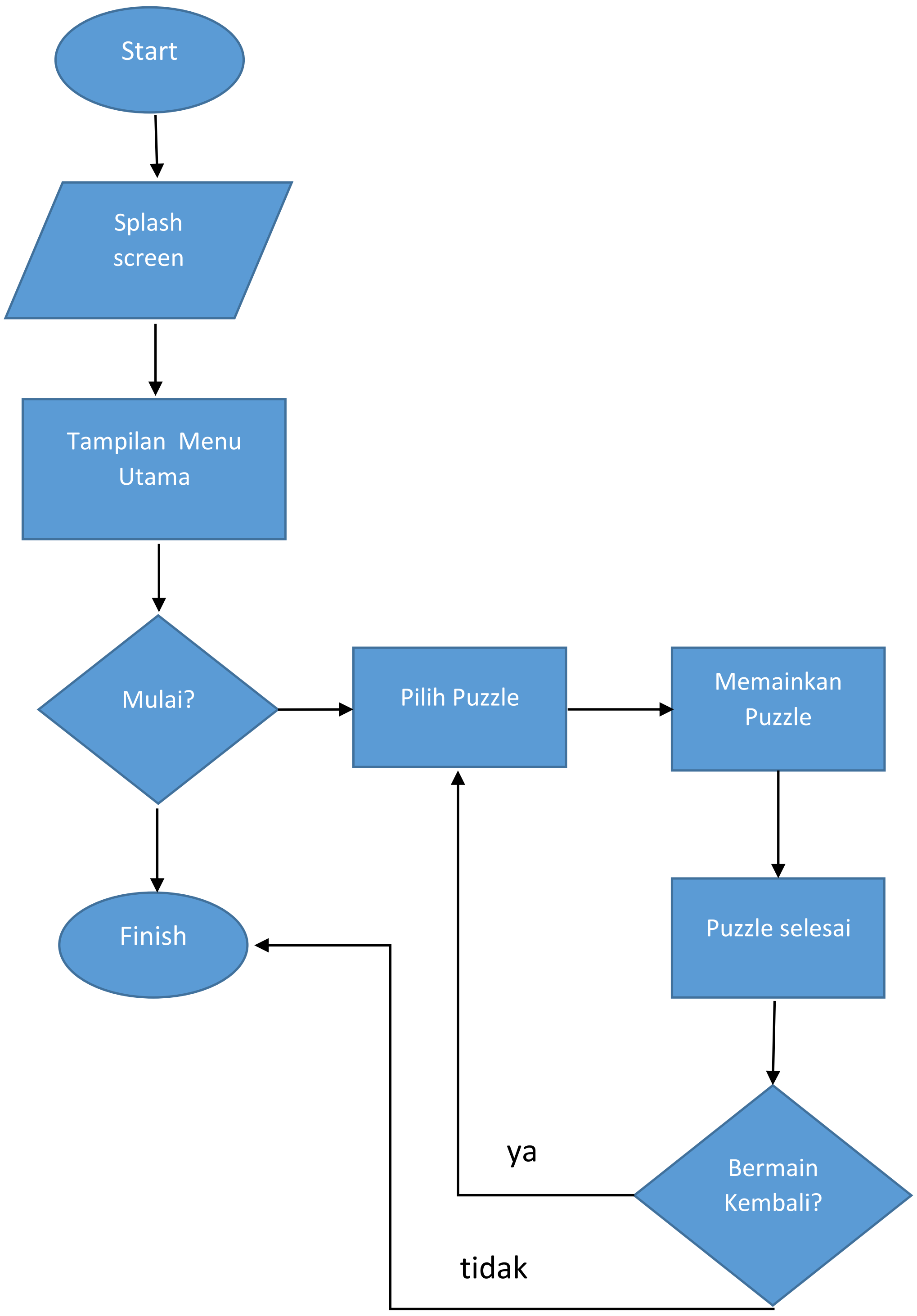
Perancangan Aplikasi Game Puzzle Tokoh Sejarah

1. Tujuan dan manfaat game
    a. Mempelajari dan mengenal lebih dalam tentang pembuatan game puzzle
    b. Mengetahui lebih dalam tentang bahasa pemrograman yang di pakai dalam pembuatan game puzzle
    c. Melatih otak dalam memecahkan suatu masalah
    d. Mempertajam daya ingat
    e. Mampu berfikir dan bertindak lebih cepat dalam mengambil suatu keputusan

2. Gambaran Aplikasi

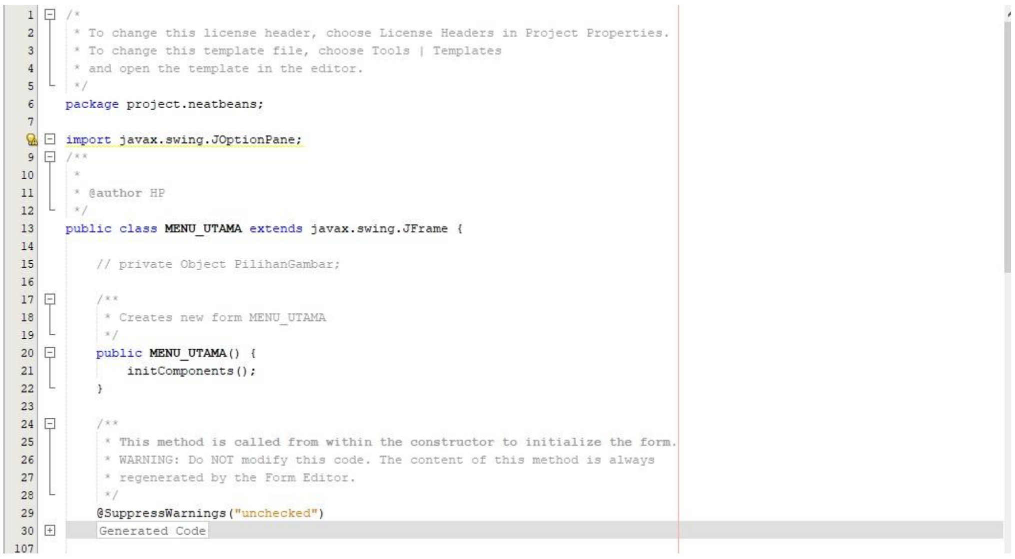
    Game puzzle merupakan permainan menyusun potongan gambar dengan aturan sebuah potongan potongan dapat dipindahkan dengan menggesernya ke ruang kosong. Umumnya orang yang memainkan puzzle butuh waktu lama dalam menyelesaikan permainan ini. Hal ini disebabkan karena pada permainan ini tidak ada informasi tambahan yang dimiliki untuk membantu melakukan pencarian solusi, sehingga saat proses penyusunan potongan-potongan menjadi suatu tantangan bagi kita yang memainkannya.

3. Flowchart

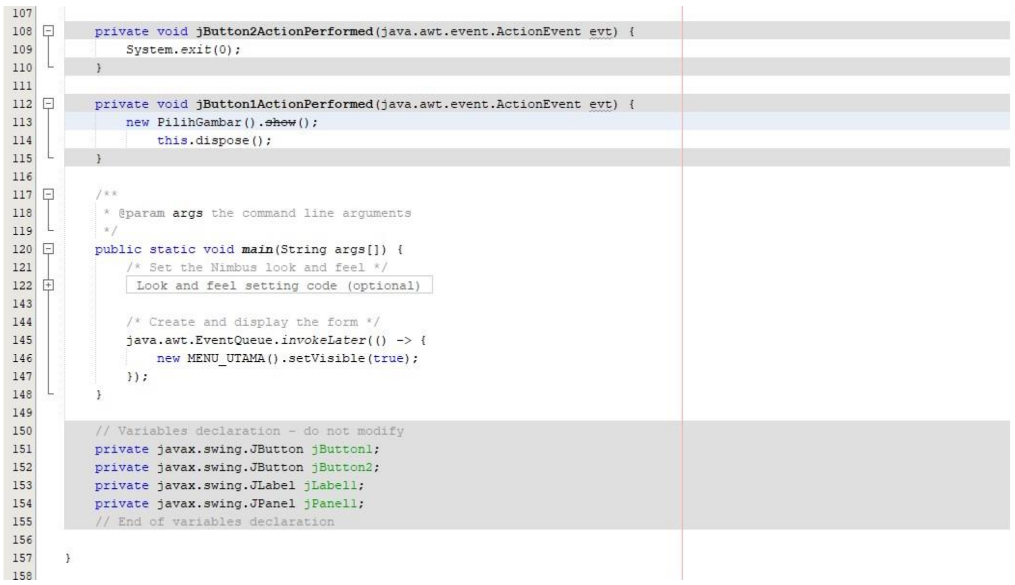

Start
Splash screen
Tampilan Menu Utama
Mulai?
Pilih Puzzle
Memainkan Puzzle
Puzzle selesai
Bermain Kembali?
ya
tidak
Finish

4. Source Code

   a. Source Code Tampilan Utama

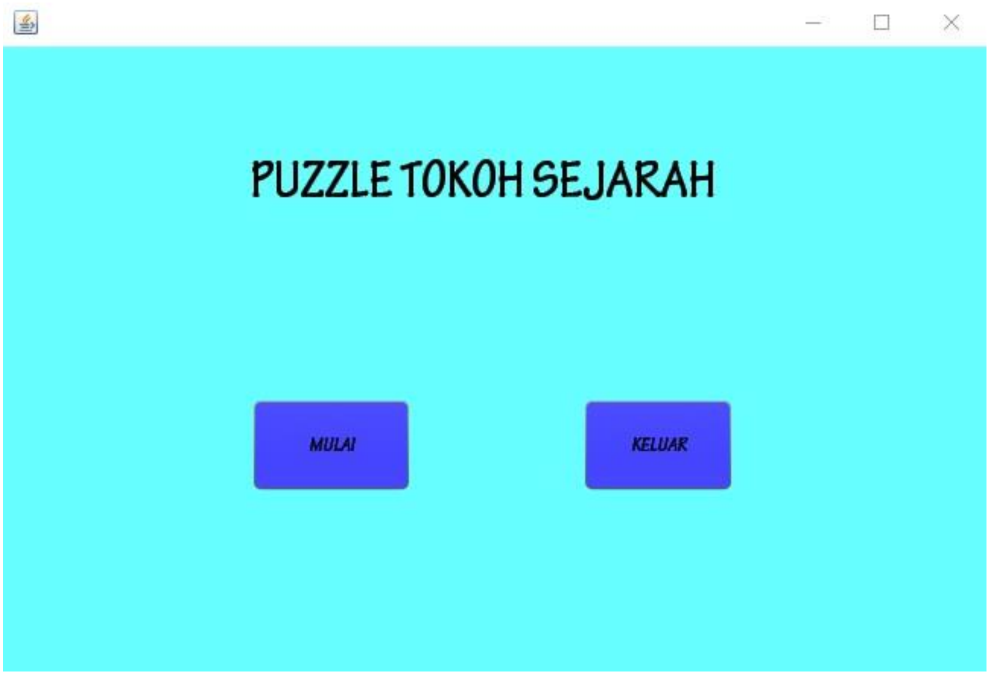


```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package project.neatbeans;

import javax.swing.JOptionPane;
/**
 *
 * @author HP
 */
public class MENU_UTAMA extends javax.swing.JFrame {

    // private Object PilihanGambar;

    /**
     * Creates new form MENU_UTAMA
     */
    public MENU_UTAMA() {
        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
```

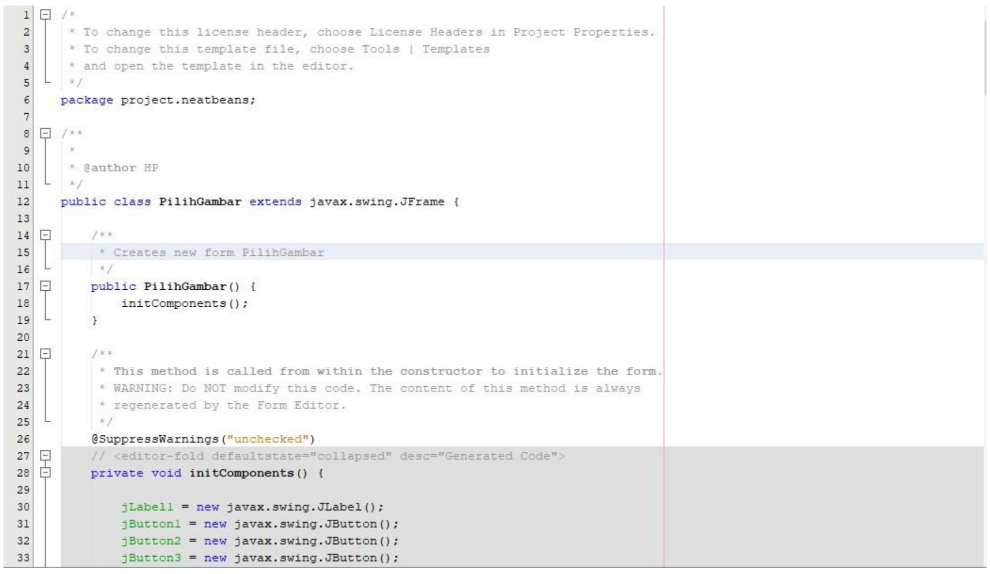


```java

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        System.exit(0);
    }

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        new PilihGambar().~~show~~();
            this.dispose();
    }

    /**
     * @param args the command line arguments
     */
    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 MENU_UTAMA().setVisible(true);
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JPanel jPanel1;
    // End of variables declaration

}
```

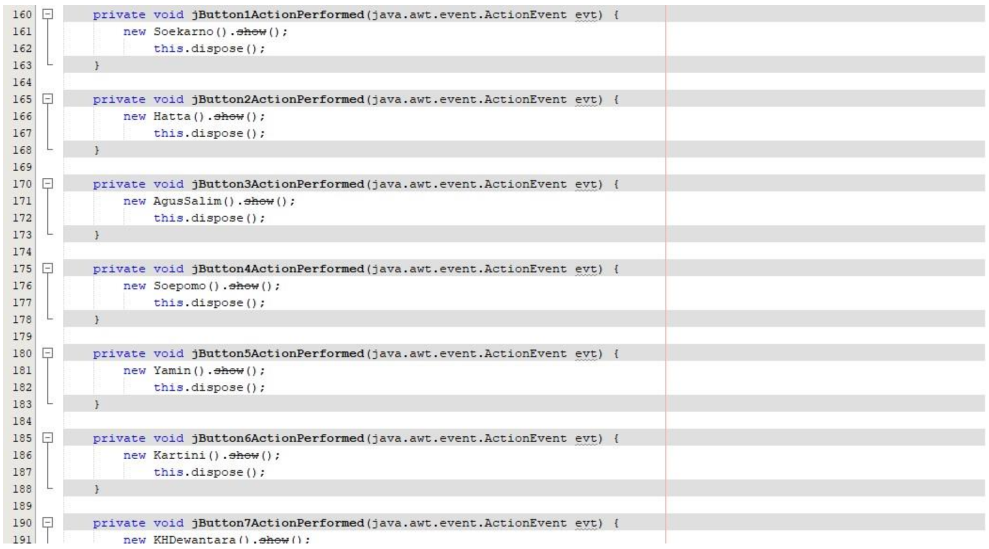
Gambar Tampilan Utama



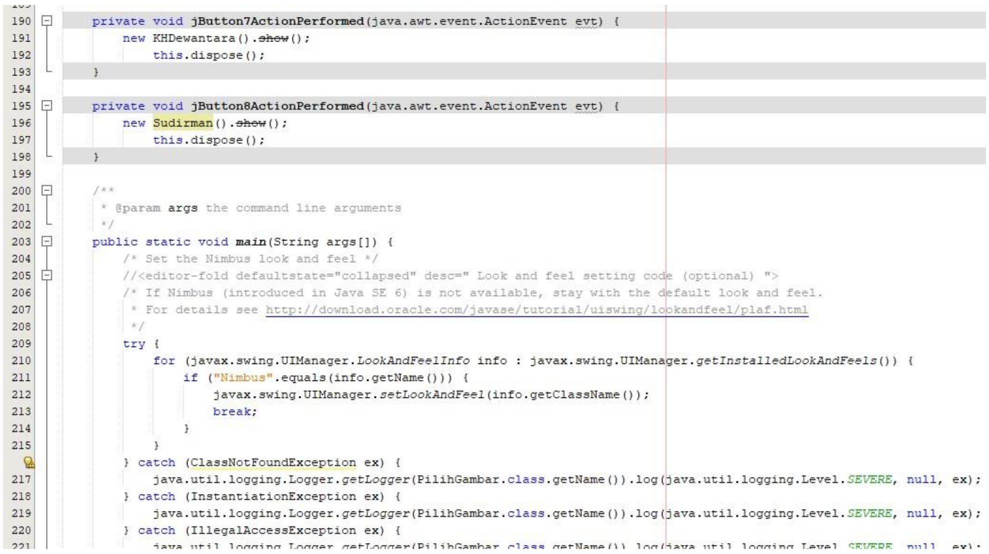
b. Source Code Tampilan Pilihan Gambar



```java
/*
 * 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 project.neatbeans;

/**
 *
 * @author HP
 */
public class PilihGambar extends javax.swing.JFrame {

    /**
     * Creates new form PilihGambar
     */
    public PilihGambar() {
        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")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        jButton2 = new javax.swing.JButton();
        jButton3 = new javax.swing.JButton();
```

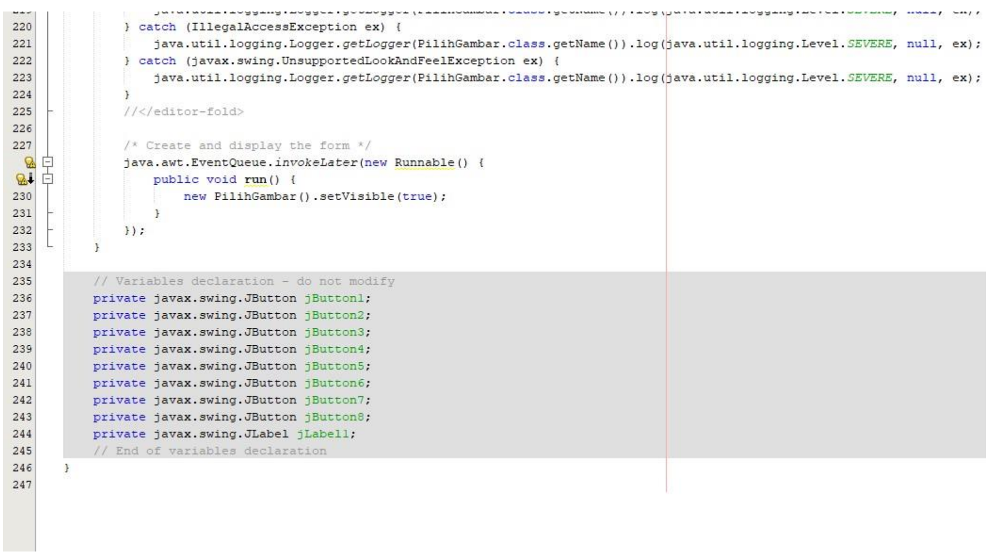


```
    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        new Soekarno().~~show~~();
            this.dispose();
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        new Hatta().~~show~~();
            this.dispose();
    }

    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        new AgusSalim().~~show~~();
            this.dispose();
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        new Soepomo().~~show~~();
            this.dispose();
    }

    private void jButton5ActionPerformed(java.awt.event.ActionEvent evt) {
        new Yamin().~~show~~();
            this.dispose();
    }

    private void jButton6ActionPerformed(java.awt.event.ActionEvent evt) {
        new Kartini().~~show~~();
            this.dispose();
    }

    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
        new KHDewantara().~~show~~();
```



```
    private void jButton7ActionPerformed(java.awt.event.ActionEvent evt) {
        new KHDewantara().~~show~~();
```

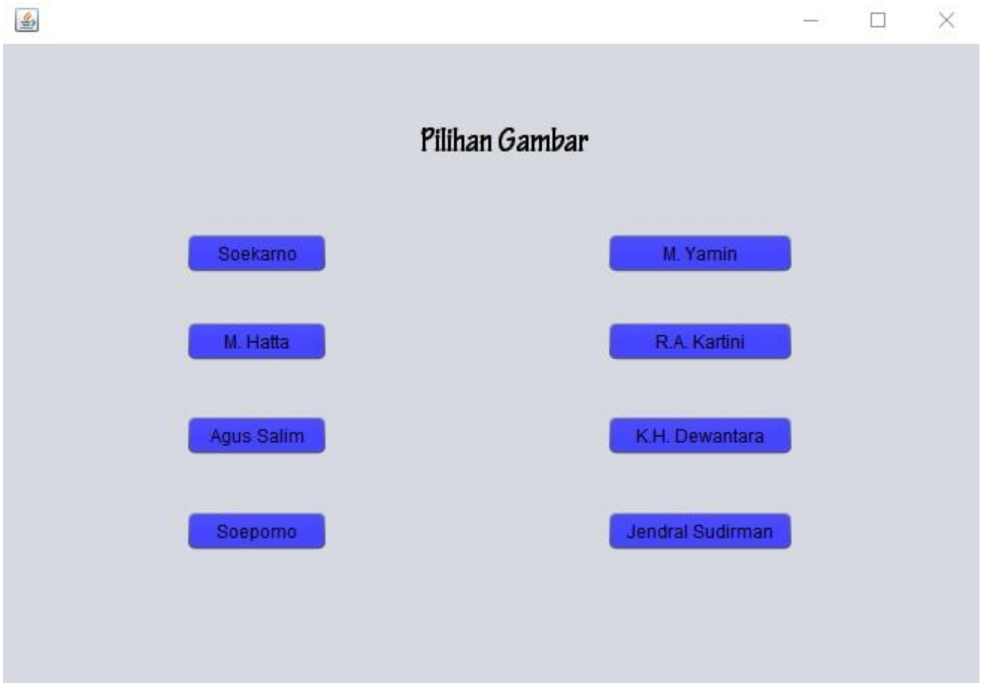
```
            this.dispose();
    }

    private void jButton8ActionPerformed(java.awt.event.ActionEvent evt) {
        new Sudirman().~~show~~();
            this.dispose();
    }

    /**
     * @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(PilihGambar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(PilihGambar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(PilihGambar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```



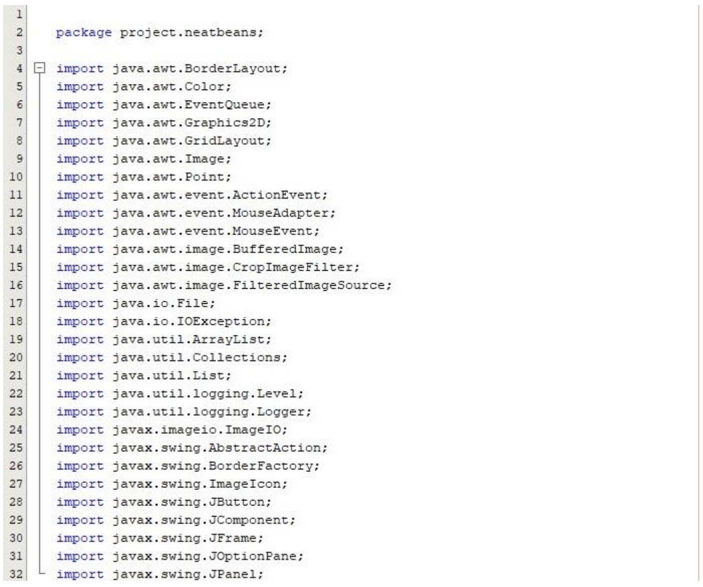
```
            java.util.logging.Logger.getLogger(PilihGambar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(PilihGambar.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);
```

```
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(PilihGambar.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 PilihGambar().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JButton jButton4;
    private javax.swing.JButton jButton5;
    private javax.swing.JButton jButton6;
    private javax.swing.JButton jButton7;
    private javax.swing.JButton jButton8;
    private javax.swing.JLabel jLabel1;
    // End of variables declaration
}
```

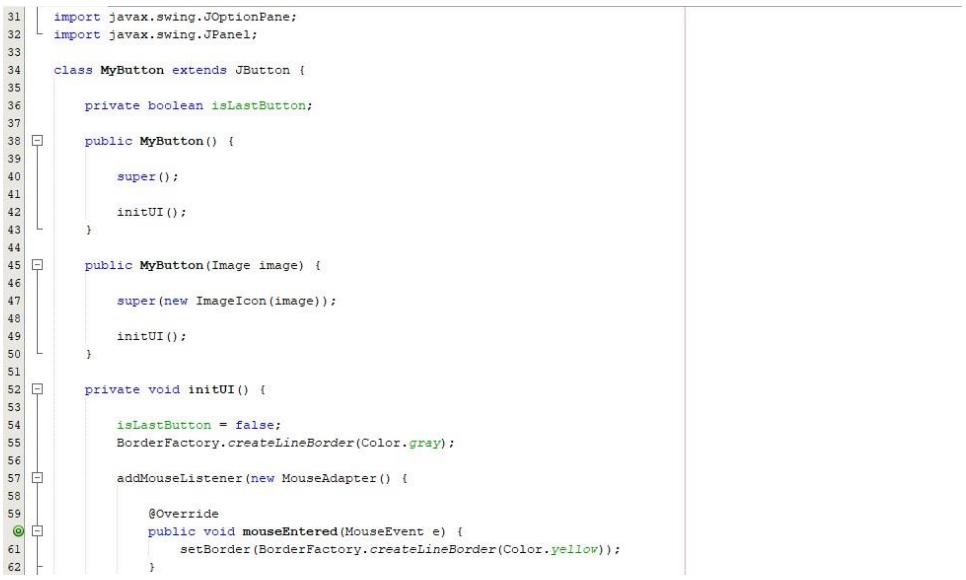
Gambar Tampilan Pilihan Gambar



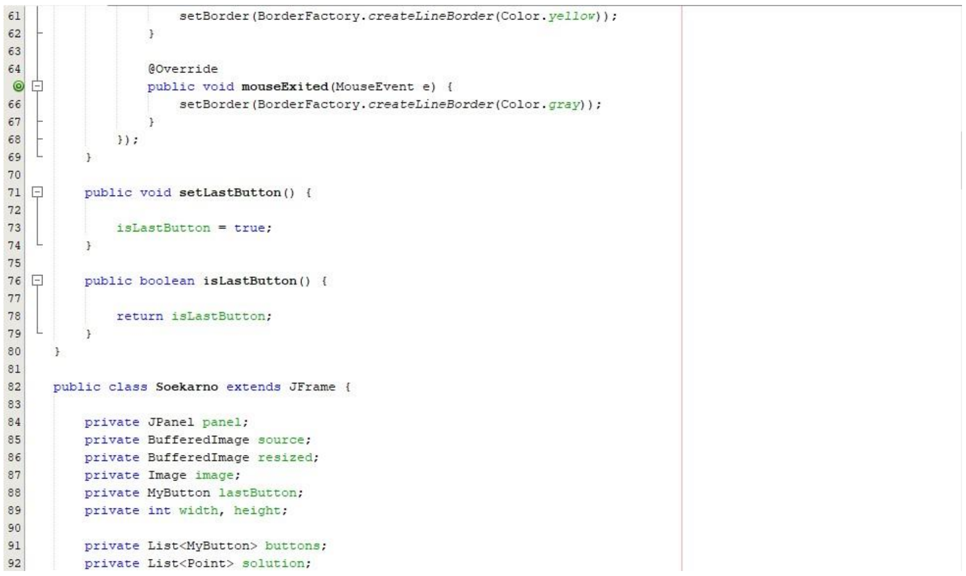
c. Source Code Game Puzzle



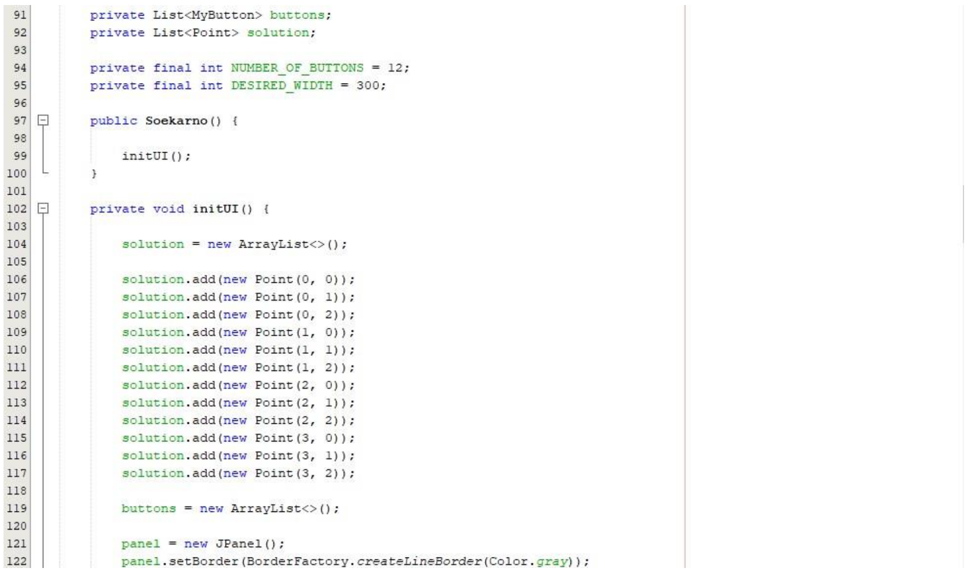
```java

package project.neatbeans;

import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.EventQueue;
import java.awt.Graphics2D;
import java.awt.GridLayout;
import java.awt.Image;
import java.awt.Point;
import java.awt.event.ActionEvent;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.awt.image.BufferedImage;
import java.awt.image.CropImageFilter;
import java.awt.image.FilteredImageSource;
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.imageio.ImageIO;
import javax.swing.AbstractAction;
import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import javax.swing.JComponent;
import javax.swing.JFrame;
import javax.swing.JOptionPane;
import javax.swing.JPanel;
```



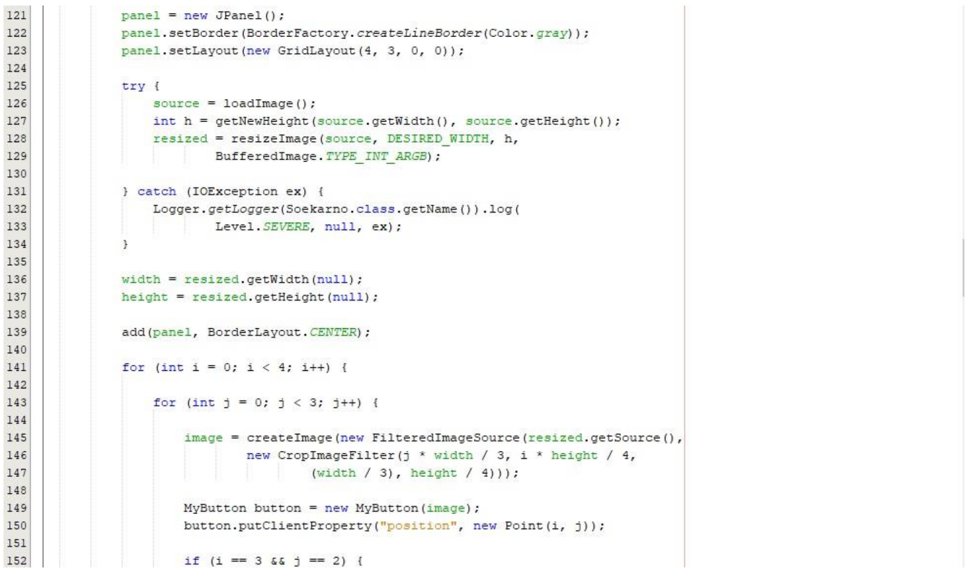
```java
import javax.swing.JOptionPane;
import javax.swing.JPanel;

class MyButton extends JButton {

    private boolean isLastButton;

    public MyButton() {

        super();

        initUI();
    }

    public MyButton(Image image) {

        super(new ImageIcon(image));

        initUI();
    }

    private void initUI() {

        isLastButton = false;
        BorderFactory.createLineBorder(Color.gray);

        addMouseListener(new MouseAdapter() {

            @Override
            public void mouseEntered(MouseEvent e) {
                setBorder(BorderFactory.createLineBorder(Color.yellow));
            }
```



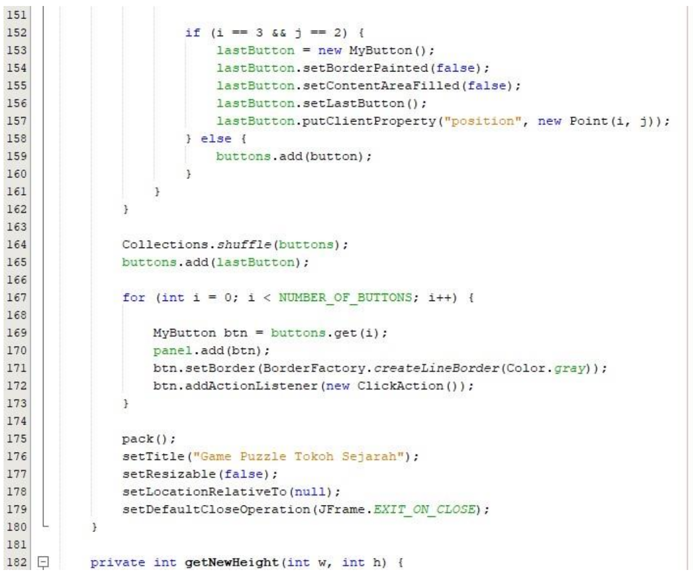
```java
                setBorder(BorderFactory.createLineBorder(Color.yellow));
            }

            @Override
            public void mouseExited(MouseEvent e) {
                setBorder(BorderFactory.createLineBorder(Color.gray));
            }
        });
    }

    public void setLastButton() {

        isLastButton = true;
    }

    public boolean isLastButton() {

        return isLastButton;
    }
}

public class Soekarno extends JFrame {

    private JPanel panel;
    private BufferedImage source;
    private BufferedImage resized;
    private Image image;
    private MyButton lastButton;
    private int width, height;

    private List<MyButton> buttons;
    private List<Point> solution;
```



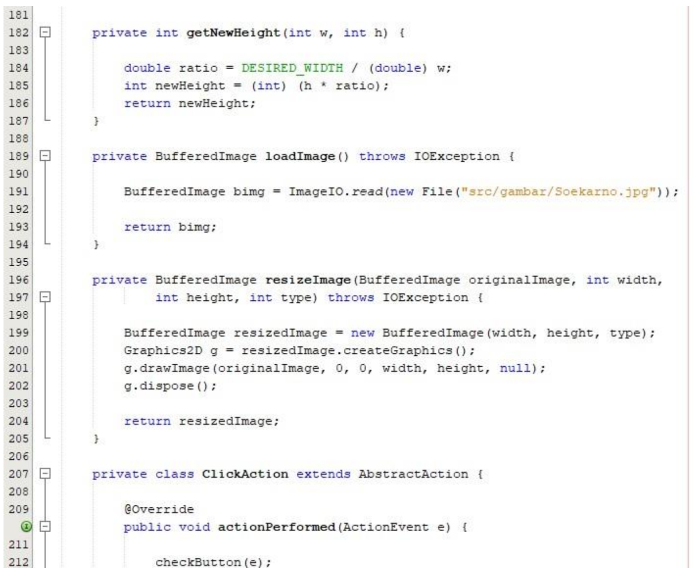
```java
    private List<MyButton> buttons;
    private List<Point> solution;

    private final int NUMBER_OF_BUTTONS = 12;
    private final int DESIRED_WIDTH = 300;

    public Soekarno() {

        initUI();
    }

    private void initUI() {

        solution = new ArrayList<>();

        solution.add(new Point(0, 0));
        solution.add(new Point(0, 1));
        solution.add(new Point(0, 2));
        solution.add(new Point(1, 0));
        solution.add(new Point(1, 1));
        solution.add(new Point(1, 2));
        solution.add(new Point(2, 0));
        solution.add(new Point(2, 1));
        solution.add(new Point(2, 2));
        solution.add(new Point(3, 0));
        solution.add(new Point(3, 1));
        solution.add(new Point(3, 2));

        buttons = new ArrayList<>();

        panel = new JPanel();
        panel.setBorder(BorderFactory.createLineBorder(Color.gray));
```



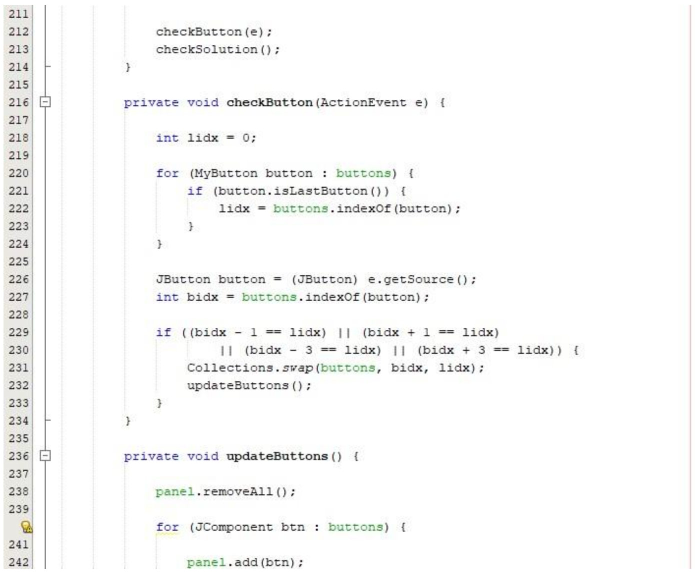
```java
        panel = new JPanel();
        panel.setBorder(BorderFactory.createLineBorder(Color.gray));
        panel.setLayout(new GridLayout(4, 3, 0, 0));

        try {
            source = loadImage();
            int h = getNewHeight(source.getWidth(), source.getHeight());
            resized = resizeImage(source, DESIRED_WIDTH, h,
                    BufferedImage.TYPE_INT_ARGB);

        } catch (IOException ex) {
            Logger.getLogger(Soekarno.class.getName()).log(
                    Level.SEVERE, null, ex);
        }

        width = resized.getWidth(null);
        height = resized.getHeight(null);

        add(panel, BorderLayout.CENTER);

        for (int i = 0; i < 4; i++) {

            for (int j = 0; j < 3; j++) {

                image = createImage(new FilteredImageSource(resized.getSource(),
                        new CropImageFilter(j * width / 3, i * height / 4,
                                (width / 3), height / 4)));

                MyButton button = new MyButton(image);
                button.putClientProperty("position", new Point(i, j));

                if (i == 3 && j == 2) {
```



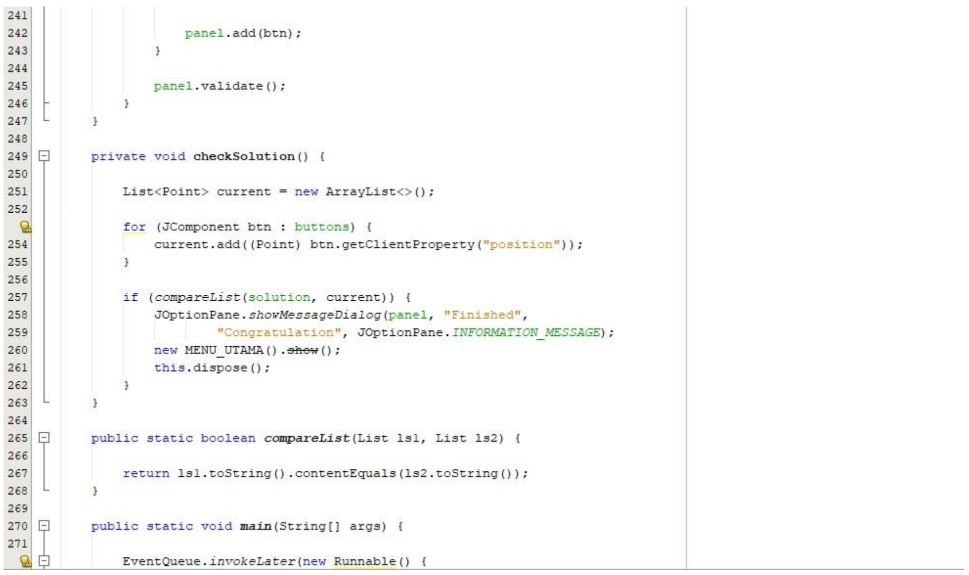
```java

                if (i == 3 && j == 2) {
                    lastButton = new MyButton();
                    lastButton.setBorderPainted(false);
                    lastButton.setContentAreaFilled(false);
                    lastButton.setLastButton();
                    lastButton.putClientProperty("position", new Point(i, j));
                } else {
                    buttons.add(button);
                }
            }
        }

        Collections.shuffle(buttons);
        buttons.add(lastButton);

        for (int i = 0; i < NUMBER_OF_BUTTONS; i++) {

            MyButton btn = buttons.get(i);
            panel.add(btn);
            btn.setBorder(BorderFactory.createLineBorder(Color.gray));
            btn.addActionListener(new ClickAction());
        }

        pack();
        setTitle("Game Puzzle Tokoh Sejarah");
        setResizable(false);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }

    private int getNewHeight(int w, int h) {
```



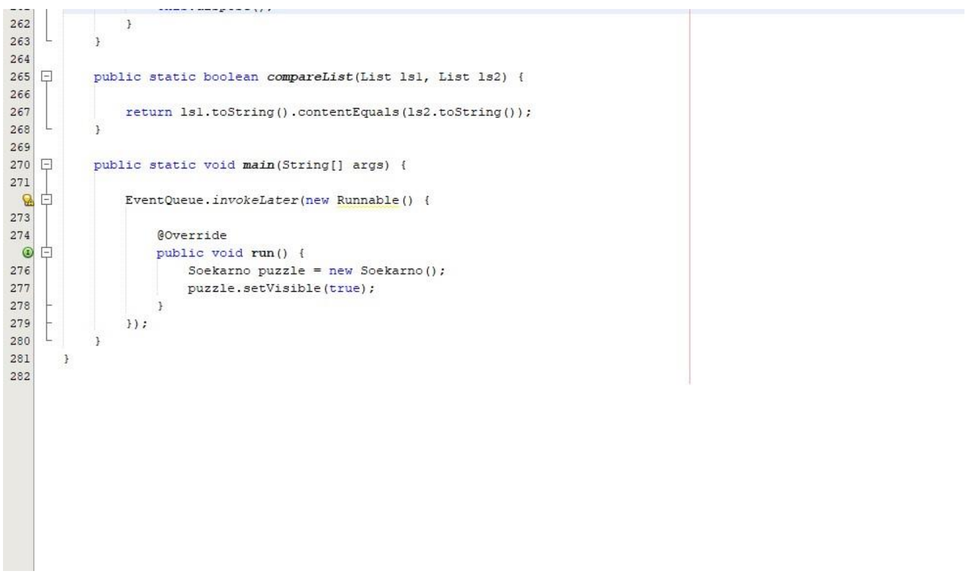
```

    private int getNewHeight(int w, int h) {

        double ratio = DESIRED_WIDTH / (double) w;
        int newHeight = (int) (h * ratio);
        return newHeight;
    }

    private BufferedImage loadImage() throws IOException {

        BufferedImage bimg = ImageIO.read(new File("src/gambar/Soekarno.jpg"));

        return bimg;
    }

    private BufferedImage resizeImage(BufferedImage originalImage, int width,
            int height, int type) throws IOException {

        BufferedImage resizedImage = new BufferedImage(width, height, type);
        Graphics2D g = resizedImage.createGraphics();
        g.drawImage(originalImage, 0, 0, width, height, null);
        g.dispose();

        return resizedImage;
    }

    private class ClickAction extends AbstractAction {

        @Override
        public void actionPerformed(ActionEvent e) {

            checkButton(e);
```



```

            checkButton(e);
            checkSolution();
        }

        private void checkButton(ActionEvent e) {

            int lidx = 0;

            for (MyButton button : buttons) {
                if (button.isLastButton()) {
                    lidx = buttons.indexOf(button);
                }
            }

            JButton button = (JButton) e.getSource();
            int bidx = buttons.indexOf(button);

            if ((bidx - 1 == lidx) || (bidx + 1 == lidx)
                    || (bidx - 3 == lidx) || (bidx + 3 == lidx)) {
                Collections.swap(buttons, bidx, lidx);
                updateButtons();
            }
        }

        private void updateButtons() {

            panel.removeAll();

            for (JComponent btn : buttons) {

                panel.add(btn);
```

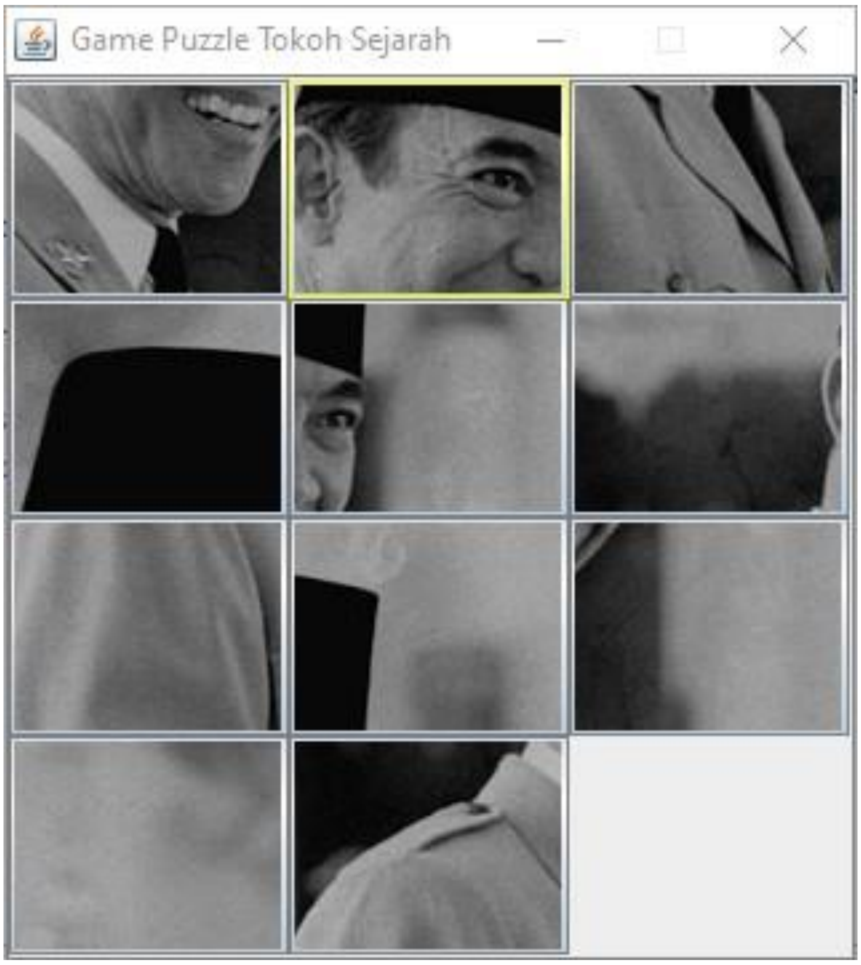


```

                panel.add(btn);
            }

            panel.validate();
        }
    }

    private void checkSolution() {

        List<Point> current = new ArrayList<>();

        for (JComponent btn : buttons) {
            current.add((Point) btn.getClientProperty("position"));
        }

        if (compareList(solution, current)) {
            JOptionPane.showMessageDialog(panel, "Finished",
                    "Congratulation", JOptionPane.INFORMATION_MESSAGE);
            new MENU_UTAMA().~~show~~();
            this.dispose();
        }
    }

    public static boolean compareList(List ls1, List ls2) {

        return ls1.toString().contentEquals(ls2.toString());
    }

    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {
```



```
        }
    }

    public static boolean compareList(List ls1, List ls2) {

        return ls1.toString().contentEquals(ls2.toString());
    }

    public static void main(String[] args) {

        EventQueue.invokeLater(new Runnable() {

            @Override
            public void run() {
                Soekarno puzzle = new Soekarno();
                puzzle.setVisible(true);
            }
        });
    }
}
```

Gambar Tampilan Game Puzzle


Game Puzzle Tokoh Sejarah