

**LAPORAN PBO
PROGRES 2 PROJEK**



Dosen Pengampu:
Slamet Tryanto, S.ST

Disusun Oleh:
FERDI FEBRIAN
NIM: 202013001

**PROGRAM STUDI INFORMATIKA
POLITEKNIK KAMPAR**

2021

A. TAMPILAN PROGRES

1. Menu Utama

a. Tampilan



b. Sourcode

```
29 private void initComponents() {
30
31     jPanel1 = new javax.swing.JPanel();
32     jLabel1 = new javax.swing.JLabel();
33     jButton1 = new javax.swing.JButton();
34     jButton2 = new javax.swing.JButton();
35
36     setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
37
38     jPanel1.setBackground(new java.awt.Color(153, 0, 153));
39
40     jLabel1.setFont(new java.awt.Font("Verdana", 1, 24)); // NOI18N
41     jLabel1.setText("GAME PUZZEL HEWAN");
42
43     jButton1.setBackground(new java.awt.Color(0, 0, 153));
44     jButton1.setFont(new java.awt.Font("Verdana", 1, 14)); // NOI18N
45     jButton1.setText("KELUAR");
46     jButton1.addActionListener(new java.awt.event.ActionListener() {
47         public void actionPerformed(java.awt.event.ActionEvent evt) {
48             jButton1ActionPerformed(evt);
49         }
50     });
51
52     jButton2.setBackground(new java.awt.Color(0, 0, 153));
53     jButton2.setFont(new java.awt.Font("Verdana", 1, 14)); // NOI18N
54     jButton2.setText("MULAI");
55     jButton2.addActionListener(new java.awt.event.ActionListener() {
56         public void actionPerformed(java.awt.event.ActionEvent evt) {
57             jButton2ActionPerformed(evt);
58         }
59     });
60 }
```

2. Pemilihan Gambar

a. Tampilan

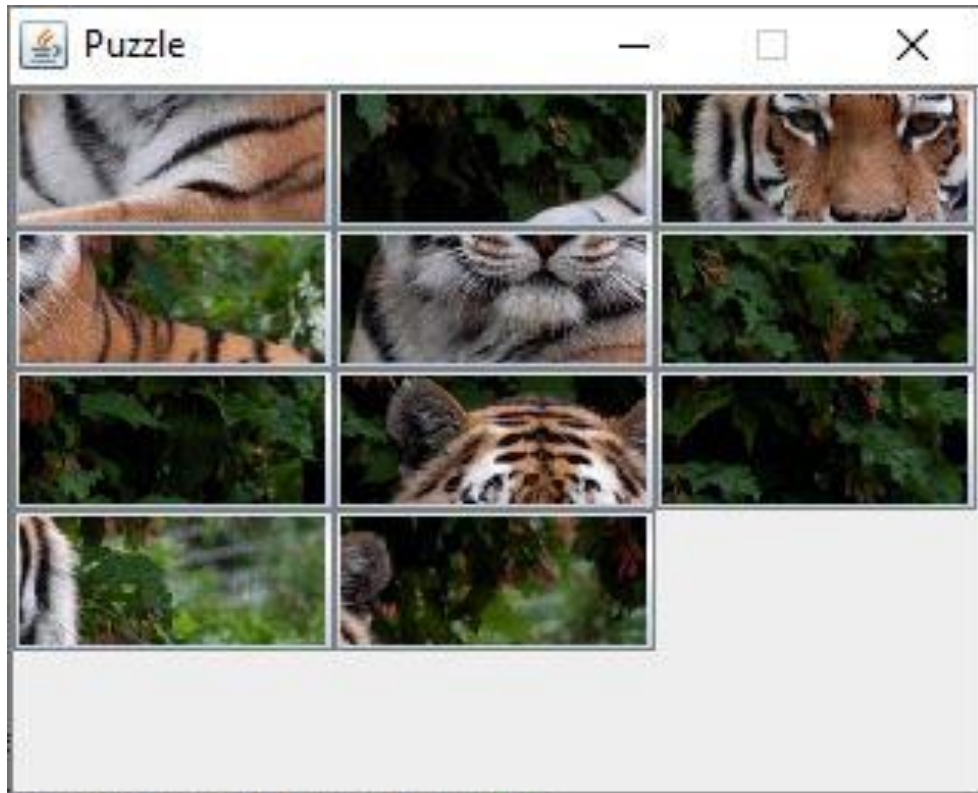


b. Sourcode

```
134 private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
135     new Rusa().show();  
136     this.dispose();  
137 }  
138  
139 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
140     new Harimau().show();  
141     this.dispose();  
142 }  
143  
144 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {  
145     new Zebra().show();  
146     this.dispose();  
147 }  
148  
149 private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {  
150     new Jerapah().show();  
151     this.dispose();  
152 }  
153  
154 /**  
155  * @param args the command line arguments  
156  */  
157 public static void main(String args[]) {  
158     /* Set the Nimbus look and feel */  
159     Look and feel setting code (optional)  
160  
161     /* Create and display the form */  
162     java.awt.EventQueue.invokeLater(new Runnable() {  
163         public void run() {  
164             new PilihGambar().setVisible(true);  
165         }  
166     });  
167 }
```

3. Puzzel Harimau

a. Tampilan

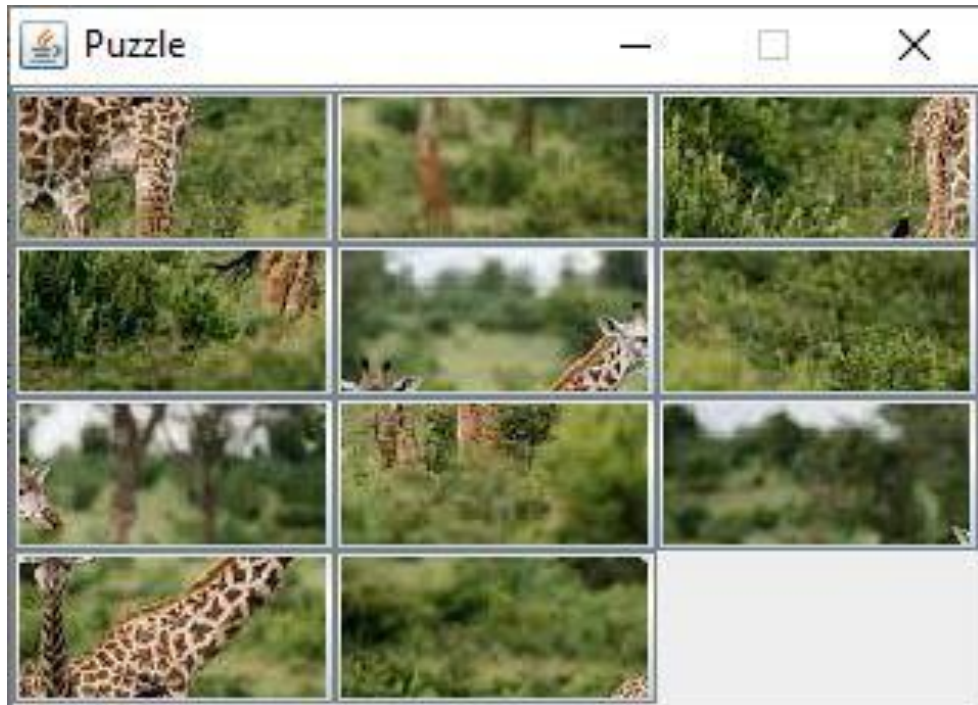


b. Sourcode

```
93 public class Harimau extends JFrame {
94
95     private JPanel panel;
96     private BufferedImage source;
97     private BufferedImage resized;
98     private Image image;
99     private JButton lastButton;
100     private int width, height;
101
102     private List<JButton> buttons;
103     private List<Point> solution;
104
105     private final int NUMBER_OF_BUTTONS = 12;
106     private final int DESIRED_WIDTH = 300;
107
108     public Harimau() {
109         initUI();
110     }
111
112     private void initUI() {
113         solution = new ArrayList<>();
114         solution.add(new Point(0, 0));
115         solution.add(new Point(0, 1));
116         solution.add(new Point(0, 2));
117         solution.add(new Point(1, 0));
118         solution.add(new Point(1, 1));
119         solution.add(new Point(1, 2));
120         solution.add(new Point(2, 0));
```

4. Puzzel Jerapah

a. Tampilan

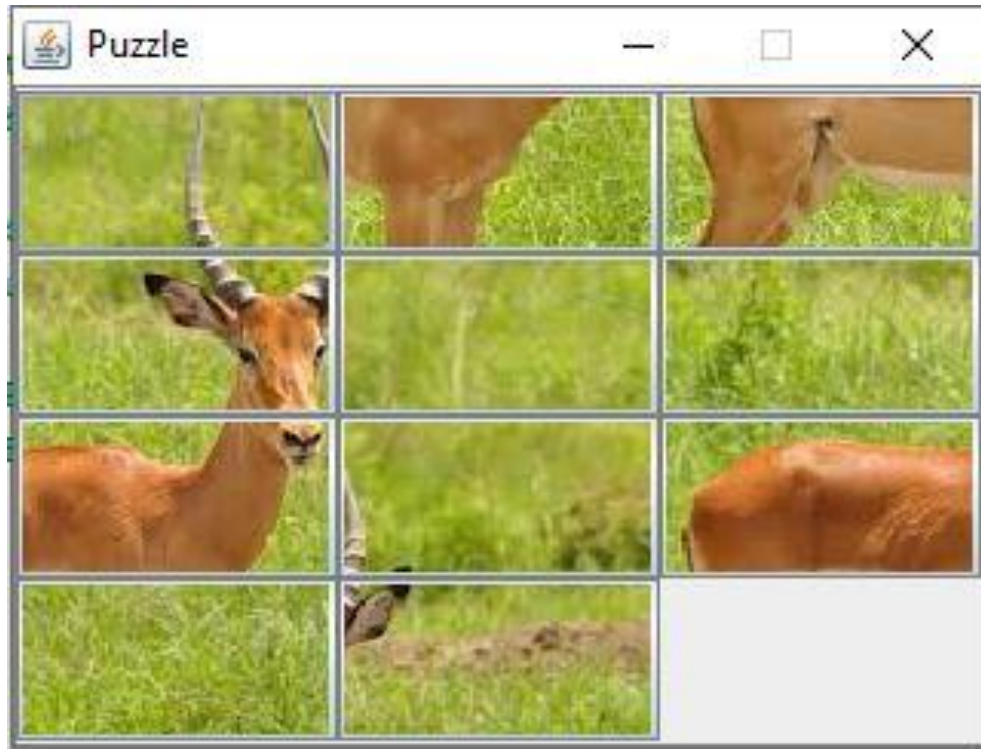


b. Sourcode

```
84 public class Jerapah extends JFrame {
85
86     private JPanel panel;
87     private BufferedImage source;
88     private BufferedImage resized;
89     private Image image;
90     private JButton lastButton;
91     private int width, height;
92
93     private List<JButton> buttons;
94     private List<Point> solution;
95
96     private final int NUMBER_OF_BUTTONS = 12;
97     private final int DESIRED_WIDTH = 300;
98
99     public Jerapah() {
100
101         initUI();
102     }
103
104     private void initUI() {
105
106         solution = new ArrayList<>();
107
108         solution.add(new Point(0, 0));
109         solution.add(new Point(0, 1));
110         solution.add(new Point(0, 2));
111         solution.add(new Point(1, 0));
112         solution.add(new Point(1, 1));
113         solution.add(new Point(1, 2));
114         solution.add(new Point(2, 0));
115         solution.add(new Point(2, 1));
```


5. Puzzle Rusa

a. Tampilan

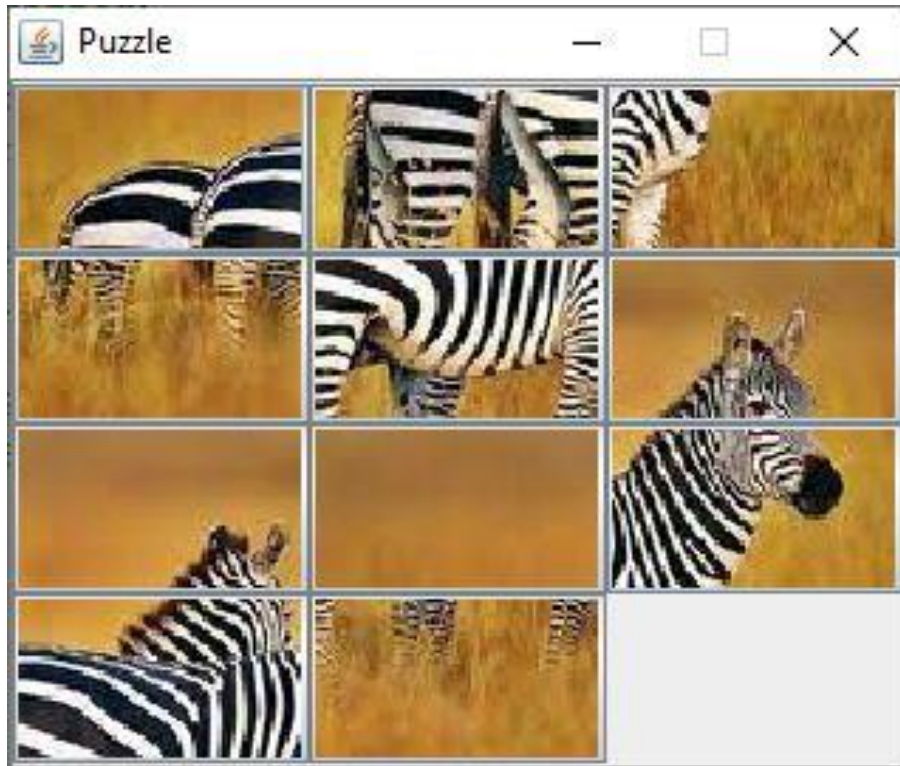


b. Sourcode

```
84 public class Rusa extends JFrame {
85
86     private JPanel panel;
87     private BufferedImage source;
88     private BufferedImage resized;
89     private Image image;
90     private JButton lastButton;
91     private int width, height;
92
93     private List<JButton> buttons;
94     private List<Point> solution;
95
96     private final int NUMBER_OF_BUTTONS = 12;
97     private final int DESIRED_WIDTH = 300;
98
99     public Rusa() {
100
101         initUI();
102     }
103
104     private void initUI() {
105
106         solution = new ArrayList<>();
107
108         solution.add(new Point(0, 0));
109         solution.add(new Point(0, 1));
110         solution.add(new Point(0, 2));
111         solution.add(new Point(1, 0));
112         solution.add(new Point(1, 1));
113         solution.add(new Point(1, 2));
114         solution.add(new Point(2, 0));
115         solution.add(new Point(2, 1));
```

6. Puzzle Zebra

a. Tampilan



b. Sourcode

```
84 public class Zebra extends JFrame {
85
86     private JPanel panel;
87     private BufferedImage source;
88     private BufferedImage resized;
89     private Image image;
90     private JButton lastButton;
91     private int width, height;
92
93     private List<JButton> buttons;
94     private List<Point> solution;
95
96     private final int NUMBER_OF_BUTTONS = 12;
97     private final int DESIRED_WIDTH = 300;
98
99     public Zebra() {
100
101         initUI();
102     }
103
104     private void initUI() {
105
106         solution = new ArrayList<>();
107
108         solution.add(new Point(0, 0));
109         solution.add(new Point(0, 1));
110         solution.add(new Point(0, 2));
111         solution.add(new Point(1, 0));
112         solution.add(new Point(1, 1));
113         solution.add(new Point(1, 2));
114         solution.add(new Point(2, 0));
115         solution.add(new Point(2, 1));
```


B. KESULITAN

Dalam mengerjakan projek ini saya cukup banyak mengalami kesulitan terutama pada sourcode yang tidak valid. Pada saat ini saya mengalami kesulitan dimana tidak bisanya puzzle yang saya buat di besarkan lagi.