

APPENDIX

```
1.      import java.awt.Color;
2.      import java.awt.Graphics2D;
3.      import java.awt.Image;
4.      import java.awt.event.ActionEvent;
5.      import java.awt.event.ActionListener;
6.      import java.io.File;
7.      import javax.swing.*;
8.      import javax.swing.filechooser.FileNameExtensionFilter;
9.      import java.awt.image.BufferedImage;
10.     import java.io.*;
11.     import java.util.Scanner;
12.     import java.util.logging.Level;
13.     import java.util.logging.Logger;
14.     import javax.imageio.*;
15.     import java.lang.Math;
16.
17.     /**
18.      *
19.      * @author windows10
20.      */
21.     public class Form extends javax.swing.JFrame {
22.         /**
23.          * Creates new form Form
24.          */
25.         JLabel label;
26.         String path;
27.         int x = 300;
28.         int y = 300;
29.         int a;
30.         int w, h;
31.         int x0, y0;
32.         int avgR, avgG, avgB;
33.         FileReader inputFile;
34.         float H, S, V;
35.         float[][] tampilData = new float[40][40];
36.         float[][] simpanData = new float[40][40];
37.         float[] tampungkls0 = new float[4];
38.         float[] tampungkls1 = new float[4];;
39.         float kls0, kls1;
40.         int pangkat = 2;
41.         Boolean kls0kecil = true;
42.         float w1, w2, w3, w4;
43.
44.         Float learnRate = 0.05f;
45.         Float fungsi = 0.1f;
46.
```

```

47.         public Form() {
48.             initComponents();
49.             btProses.setEnabled(false);
50.             btBrowse.setEnabled(false);
51.             btLatih.setVisible(false);
52.             btLatih.doClick();
53.         }
54.
55.
56.         /**
57.          * This method is called from within the constructor to
initialize the form.
58.          * WARNING: Do NOT modify this code. The content of
this method is always
59.          * regenerated by the Form Editor.
60.          */
61.         @SuppressWarnings("unchecked")
62.         // <editor-fold defaultstate="collapsed"
desc="Generated Code">
63.         private void initComponents() {
64.
65.             jPanel1 = new javax.swing.JPanel();
66.             btBrowse = new javax.swing.JButton();
67.             pnlGambar = new javax.swing.JPanel();
68.             jLabel1 = new javax.swing.JLabel();
69.             btProses = new javax.swing.JButton();
70.             btLatih = new javax.swing.JButton();
71.             jPanel2 = new javax.swing.JPanel();
72.             jPanel3 = new javax.swing.JPanel();
73.             jLabelCrop = new javax.swing.JLabel();
74.             jLabel2 = new javax.swing.JLabel();
75.             jLabel3 = new javax.swing.JLabel();
76.             jLabel4 = new javax.swing.JLabel();
77.             Rvalue = new javax.swing.JTextField();
78.             Gvalue = new javax.swing.JTextField();
79.             Bvalue = new javax.swing.JTextField();
80.             jLabel5 = new javax.swing.JLabel();
81.             Hvalue = new javax.swing.JTextField();
82.             jLabel6 = new javax.swing.JLabel();
83.             jLabel7 = new javax.swing.JLabel();
84.             Svalue = new javax.swing.JTextField();
85.             Vvalue = new javax.swing.JTextField();
86.             jPanel4 = new javax.swing.JPanel();
87.             jLabel8 = new javax.swing.JLabel();
88.             jLabel9 = new javax.swing.JLabel();
89.             w1Value = new javax.swing.JTextField();
90.             w0Value = new javax.swing.JTextField();
91.             jLabel10 = new javax.swing.JLabel();
92.             lblKesimpulan = new javax.swing.JLabel();
93.
94.
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
95.
96.             jPanel1.setBackground(new java.awt.Color(51, 255,
255));
97.

```

```

98.         btBrowse.setBackground(new java.awt.Color(255, 51,
255));
99.         btBrowse.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
100.        btBrowse.setText("Input Gambar");
101.        btBrowse.addActionListener(new
java.awt.event.ActionListener() {
102.            public void
actionPerformed(java.awt.event.ActionEvent evt) {
103.                btBrowseActionPerformed(evt);
104.            }
105.        });
106.
107.        jLabel1.setBackground(new java.awt.Color(102, 0,
102));
108.
109.        javax.swing.GroupLayout pnlGambarLayout = new
javax.swing.GroupLayout(pnlGambar);
110.        pnlGambar.setLayout(pnlGambarLayout);
111.        pnlGambarLayout.setHorizontalGroup(
112.            pnlGambarLayout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
113.                .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
114.            );
115.        pnlGambarLayout.setVerticalGroup(
116.            pnlGambarLayout.createParallelGroup(javax.swing.GroupLayout.Alignmen
t.LEADING)
117.                .addGroup(pnlGambarLayout.createSequentialGroup()
118.                    .addContainerGap()
119.                    .addComponent(jLabel1,
javax.swing.GroupLayout.DEFAULT_SIZE,245, Short.MAX_VALUE)
120.                    .addContainerGap())
121.                );
122.
123.        btProses.setBackground(new java.awt.Color(255, 51,
255));
124.        btProses.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
125.        btProses.setText("Proses");
126.        btProses.addActionListener(new
java.awt.event.ActionListener() {
127.            public void
actionPerformed(java.awt.event.ActionEvent evt) {
128.                btProsesActionPerformed(evt);
129.            }
130.        });
131.
132.        btLatih.setBackground(new java.awt.Color(255, 51,
255));
133.        btLatih.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N

```

```

134.         btLatih.setText("Latih");
135.         btLatih.addActionListener(new
java.awt.event.ActionListener() {
136.             public void
actionPerformed(java.awt.event.ActionEvent evt) {
137.                 btLatihActionPerformed(evt);
138.             }
139.         });
140.
141.         javax.swing.GroupLayout jPanel1Layout = new
javax.swing.GroupLayout(jPanel1);
142.         jPanel1.setLayout(jPanel1Layout);
143.         jPanel1Layout.setHorizontalGroup(
144.             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
145.                 .addGroup(jPanel1Layout.createSequentialGroup()
146.                     .addGap(23, 23, 23)
147.                     .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING, false)
148.                         .addComponent(pnlGambar,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
149.                         .addComponent(btBrowse,
javax.swing.GroupLayout.DEFAULT_SIZE, 242, Short.MAX_VALUE)
150.                         .addComponent(btProses,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
151.                         .addComponent(btLatih,
javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
152.                     .addGap(28, Short.MAX_VALUE))
153.                 );
154.         jPanel1Layout.setVerticalGroup(
155.             jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
156.                 .addGroup(jPanel1Layout.createSequentialGroup()
157.                     .addGap()
158.                     .addComponent(pnlGambar,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
159.                     .addGap(18, 18, 18)
160.                     .addComponent(btBrowse,
javax.swing.GroupLayout.PREFERRED_SIZE, 47,
javax.swing.GroupLayout.PREFERRED_SIZE)
161.                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
162.                     .addComponent(btProses,
javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

163.
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
164.         .addComponent(btLatih,
javax.swing.GroupLayout.PREFERRED_SIZE, 44,
javax.swing.GroupLayout.PREFERRED_SIZE)
165.         .addGap(20, 20, 20))
166.     );
167.
168.     jPanel2.setBackground(new java.awt.Color(153, 153,
153));
169.
170.     javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
171.     jPanel3.setLayout(jPanel3Layout);
172.     jPanel3Layout.setHorizontalGroup(
173.
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
174.         .addGroup(jPanel3Layout.createSequentialGroup()
175.             .addContainerGap()
176.             .addComponent(jLabelCrop,
javax.swing.GroupLayout.PREFERRED_SIZE, 206,
javax.swing.GroupLayout.PREFERRED_SIZE)
177.             .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
178.         );
179.     jPanel3Layout.setVerticalGroup(
180.
jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
181.         .addGroup(jPanel3Layout.createSequentialGroup()
182.             .addContainerGap()
183.             .addComponent(jLabelCrop,
javax.swing.GroupLayout.PREFERRED_SIZE, 214,
javax.swing.GroupLayout.PREFERRED_SIZE)
184.             .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
185.         );
186.
187.     jLabel2.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
188.     jLabel2.setText("R");
189.
190.     jLabel3.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
191.     jLabel3.setText("G");
192.
193.     jLabel4.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
194.     jLabel4.setText("B");
195.
196.     Rvalue.setEditable(false);
197.

```

```
198.         Gvalue.setEditable(false);
199.
200.         Bvalue.setEditable(false);
201.
202.         jLabel5.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
203.         jLabel5.setText("H");
204.
205.         Hvalue.setEditable(false);
206.
207.         jLabel6.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
208.         jLabel6.setText("S");
209.
210.         jLabel7.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
211.         jLabel7.setText("V");
212.
213.         Svalue.setEditable(false);
214.
215.         Vvalue.setEditable(false);
216.
217.         javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
218.         jPanel2.setLayout(jPanel2Layout);
219.         jPanel2Layout.setHorizontalGroup(
220.     jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
221.         .addGroup(jPanel2Layout.createSequentialGroup()
222.             .addComponent(jLabel7)
223.             .addGap(10, 10, 10)
224.             .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
225.                 .addGroup(jPanel2Layout.createSequentialGroup()
226.                     .addComponent(jLabel5)
227.                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED,
228.     javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
229.                 .addGroup(jPanel2Layout.createSequentialGroup()
230.                     .addComponent(jLabel6)
231.                     .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.
RELATED,
232.     javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
```

```

232.             .addComponent(Svalue,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE))
233.
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup())
234.             .addComponent(jLabel5)
235.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
236.             .addComponent(Hvalue,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE))
237.
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup())
238.             .addComponent(jLabel4)
239.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
240.             .addComponent(Bvalue,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE))
241.
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup())
242.             .addComponent(jLabel3)
243.
    .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
244.             .addComponent(Gvalue,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE))
245.
    .addGroup(javax.swing.GroupLayout.Alignment.LEADING,
jPanel2Layout.createSequentialGroup())
246.             .addComponent(jLabel2)
247.             .addGap(27, 27, 27)
248.             .addComponent(Rvalue,
javax.swing.GroupLayout.PREFERRED_SIZE, 88,
javax.swing.GroupLayout.PREFERRED_SIZE))
249.             .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
250.
    .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
251.             );
252.             jPanel2Layout.setVerticalGroup(
253.
jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
254.             .addGroup(jPanel2Layout.createSequentialGroup())
255.             .addContainerGap()

```

```
256.             .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
257.             .addGap(18, 18, 18)
258.
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
259.             .addComponent(jLabel2)
260.             .addComponent(Rvalue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
261.
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
262.
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
263.             .addComponent(jLabel3)
264.             .addComponent(Gvalue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
265.
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
266.
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
267.             .addComponent(jLabel4)
268.             .addComponent(Bvalue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
269.             .addGap(31, 31, 31)
270.
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
271.             .addComponent(jLabel5)
272.             .addComponent(Hvalue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
273.
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
274.
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
275.             .addComponent(jLabel6)
276.             .addComponent(Svalue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
277.
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
```



```

278.
.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.BASELINE)
279.
.addComponent(vvalue,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
280.
.addComponent(jLabel7))
281.
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
282.
);
283.
284.
jPanel4.setBackground(new java.awt.Color(51, 255,
255));
285.
286.
jLabel8.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
287.
jLabel8.setText("Nilai Matang (W0)");
288.
289.
jLabel9.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
290.
jLabel9.setText("Nilai Mentah (W1)");
291.
292.
w1Value.setEditable(false);
293.
w1Value.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
294.
295.
w0Value.setEditable(false);
296.
w0Value.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
297.
298.
jLabel10.setFont(new java.awt.Font("Tahoma", 0,
18)); // NOI18N
299.
jLabel10.setText("Kesimpulan :");
300.
301.
lblKesimpulan.setFont(new java.awt.Font("Tahoma",
0, 36)); // NOI18N
302.
lblKesimpulan.setText("-");
303.
304.
javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
305.
jPanel4.setLayout(jPanel4Layout);
306.
jPanel4Layout.setHorizontalGroup(
307.
jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
308.
.addGroup(jPanel4Layout.createSequentialGroup())
309.
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
310.
.addGroup(jPanel4Layout.createSequentialGroup())
311.
.addContainerGap())

```

```

312.
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
313.             .addComponent(jLabel8)
314.             .addComponent(jLabel9)
315.
.addGroup(jPanel4Layout.createSequentialGroup())
316.             .addGap(31, 31, 31)
317.             .addComponent(w1Value,
javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE))
318.             .addComponent(jLabel10)))
319.
.addGroup(jPanel4Layout.createSequentialGroup())
320.             .addGap(38, 38, 38)
321.             .addComponent(lblKesimpulan)))
322.             .addContainerGap(108, Short.MAX_VALUE))
323.
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
324.
.addGroup(jPanel4Layout.createSequentialGroup())
325.             .addGap(39, 39, 39)
326.             .addComponent(w0Value,
javax.swing.GroupLayout.PREFERRED_SIZE, 132,
javax.swing.GroupLayout.PREFERRED_SIZE)
327.             .addContainerGap(110,
Short.MAX_VALUE)))
328.             );
329.     jPanel4Layout.setVerticalGroup(
330.     jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.
LEADING)
331.         .addGroup(jPanel4Layout.createSequentialGroup())
332.             .addGap(23, 23, 23)
333.             .addComponent(jLabel8)
334.             .addGap(88, 88, 88)
335.             .addComponent(jLabel9)
336.             .addGap(18, 18, 18)
337.             .addComponent(w1Value,
javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE)
338.             .addGap(71, 71, 71)
339.             .addComponent(jLabel10)
340.             .addGap(18, 18, 18)
341.             .addComponent(lblKesimpulan)
342.
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
343.
.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.
Alignment.LEADING)
344.
.addGroup(jPanel4Layout.createSequentialGroup())
345.             .addGap(66, 66, 66)

```

```

346.                .addComponent(w0Value,
javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE)
347.                .addContainerGap(363,
Short.MAX_VALUE))
348.                );
349.
350.
jLabel10.getAccessibleContext().setAccessibleName("Kesimpulan");
351.
352.                javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
353.                getContentPane().setLayout(layout);
354.                layout.setHorizontalGroup(
355.
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
356.                .addGroup(layout.createSequentialGroup()
357.                .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
358.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATE
D)
359.                .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
360.                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
361.                .addComponent(jPanel4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
362.                .addContainerGap())
363.                );
364.                layout.setVerticalGroup(
365.
layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING
)
366.                .addComponent(jPanel2,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
367.                .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
368.                .addComponent(jPanel4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
369.                );
370.
371.                pack();
372.                }// </editor-fold>
373.
374.                private void
btBrowseActionPerformed(java.awt.event.ActionEvent evt) {

```

```

375.         // TODO add your handling code here:
376.         lblKesimpulan.setText("-");
377.         JFileChooser file = new JFileChooser();
378.         file.setCurrentDirectory(new
File(System.getProperty("user.home")));
379.         //filter the files
380.         FileNameExtensionFilter filter = new
FileNameExtensionFilter("*.Images", "jpg","gif","png");
381.         file.addChoosableFileFilter(filter);
382.         int result = file.showSaveDialog(null);
383.         //if the user click on save in Jfilechooser
384.         if(result == JFileChooser.APPROVE_OPTION){
385.             File selectedFile = file.getSelectedFile();
386.             path = selectedFile.getAbsolutePath();
387.             jLabel1.setIcon(ResizeImage(path));
388.             infoBox("Sukses Upload Foto", "Success");
389.         }
390.
391.         else if(result == JFileChooser.CANCEL_OPTION){
392.             System.out.println("No File Select");
393.             infoBox("Gagal Upload Foto", "Error");
394.         }
395.     }
396.
397.     private void
btProsesActionPerformed(java.awt.event.ActionEvent evt) {
398.         try {
399.             // TODO add your handling code here:
400.             BufferedImage originalImage = ImageIO.read(new
File(path));
401.             BufferedImage subImage =
resize(originalImage,300,300);
402.
403.             // Proses z Crop Gambar (Mengambil sisi Tengah
Citra)
404.             x = 20;
405.             y = 20;
406.             int w = subImage.getWidth();
407.             int h = subImage.getHeight();
408.
409.             subImage = subImage.getSubimage(x, y, (260),
(260));
410.             File outputfile = new
File("C:/Users/Asus/Pictures/Data Latihan/croppedImage2.jpg");
411.             ImageIO.write(subImage, "jpg", outputfile);
412.             ImageIcon img = new ImageIcon(subImage);
413.
414.             path = "C:/Users/Asus/Pictures/Data
Latihan/croppedImage2.jpg";
415.             jLabelCrop.setIcon(ResizeImage(path));
416.
417.
418.             // Proses Menentukan RGB Jambu
419.             originalImage = ImageIO.read(new File(path));
420.             w = originalImage.getWidth();
421.             h = originalImage.getHeight();

```

```

422.
423.         int x1 = x0 + w;
424.         int y1 = y0 + h;
425.         long sumr = 0, sumg = 0, sumb = 0;
426.         for (int x = x0; x < x1; x++)
427.         {
428.             for (int y = y0; y < y1; y++)
429.             {
430.                 Color pixel = new
Color(originalImage.getRGB(x, y));
431.                 sumr += pixel.getRed();
432.                 sumg += pixel.getGreen();
433.                 sumb += pixel.getBlue();
434.             }
435.         }
436.         int num = w * h;
437.         avgR = (int)sumr/num;
438.         avgG = (int)sumg/num;
439.         avgB = (int)sumb/num;
440.
441.         System.out.println("R-" + avgR + "   G-"+avgG+"
B-"+avgB);
442.         Rvalue.setText(String.valueOf(avgR));
443.         Gvalue.setText(String.valueOf(avgG));
444.         Bvalue.setText(String.valueOf(avgB));
445.
446.         // Proses Mendapatkan HSV
447.         float[] comp = new float[3];
448.         Color.RGBtoHSB(avgR, avgG, avgB, comp);
449.         comp[0]*= 360;
450.         comp[1]*= 100;
451.         comp[2]*= 100;
452.
453.         Hvalue.setText(Float.toString(Math.round(comp[0])));
454.         Svalue.setText(Float.toString(Math.round(comp[1])));
455.         Vvalue.setText(Float.toString(Math.round(comp[2])));
456.
457.         H = comp[0];
458.         S = comp[1];
459.         V = comp[2];
460.
461.         hitungData(H, S, V);
462.
463.     } catch (IOException ex) {
464.
465.         Logger.getLogger(Form.class.getName()).log(Level.SEVERE, null, ex);
466.     }
467.
468.     private void hitungData(float no1, float no2, float
no3)
469.     {
470.         kls0 = 0;

```

```

471.         kls1 = 0;
472.
473.         tampilData[0][0] = no1;
474.         tampilData[0][1] = no2;
475.         tampilData[0][2] = no3;
476.
477.         for (int x = 0; x < 3; x++)
478.         {
479.             float temp1, temp2;
480.             temp1 = tampilData[0][x];
481.             temp2 = simpanData[0][x];
482.             tampungkls0[x] = temp1 - temp2;
483.             kls0 += (float) Math.pow((temp1 -
temp2) ,pangkat);
484.             //System.out.println(kls0);
485.         }
486.
487.         for (int x = 0; x < 3; x++)
488.         {
489.             float temp1, temp2;
490.             temp1 = tampilData[0][x];
491.             temp2 = simpanData[1][x];
492.             tampungkls1[x] = temp1 - temp2;
493.             kls1 += (float) Math.pow((temp1 -
temp2) ,pangkat);
494.             //System.out.println(kls0);
495.         }
496.
497.         kls0 = (float) Math.sqrt(kls0);
498.         kls1 = (float) Math.sqrt(kls1);
499.
500.         w0Value.setText(Float.toString(kls0));
501.         w1Value.setText(Float.toString(kls1));
502.
503.
504.         if(kls0 <= kls1)
505.         {
506.             System.out.println("bener kls 0");
507.             infoBox("Termasuk Kelas Matang", "Success");
508.             lblKesimpulan.setText("Matang!!");
509.         }
510.
511.         else
512.         {
513.             System.out.println("bener kls 1");
514.             infoBox("Termasuk kelas Mentah", "Success");
515.             lblKesimpulan.setText("Mentah!!");
516.         }
517.     }
518.
519.     private void
btLatihActionPerformed(java.awt.event.ActionEvent evt) {
520.         btProses.setEnabled(true);
521.         btBrowse.setEnabled(true);
522.         btLatih.setEnabled(false);
523.         try {

```

```

524.          // TODO add your handling code here:
525.          // proses baca txt
526.          inputFile = new
FileReader("C:/Users/Asus/Pictures/Data Latihan/data.txt");
527.          Scanner parser = new Scanner(inputFile);
528.          a = 0;
529.          while (parser.hasNext())
530.          {
531.              String data = parser.nextLine();
532.              //          System.out.println(data);
533.              String[] temp = data.split(";");
534.              //          System.out.println(temp[0]);
535.              //          System.out.println(temp[1]);
536.              //          System.out.println(temp[2]);
537.              //          System.out.println(temp[3]);
538.              //          System.out.println(temp[4]);
539.
540.              // menyimpan data ke array
541.              for (int i = 0; i < 3; i++)
542.              {
543.                  simpanData[a][i] =
Float.parseFloat(temp[i]);
544.              }
545.              a = 1 + a;
546.          }
547.          //          System.out.println(simpanData[0][0]);
548.          //          System.out.println(simpanData[0][1]);
549.          //          System.out.println(simpanData[0][2]);
550.          //          System.out.println(simpanData[0][3]);
551.          //          System.out.println(simpanData[0][4]);
552.
553.          ProsesLVQ();
554.      }
555.      catch (FileNotFoundException ex) {
556.
557.          Logger.getLogger(Form.class.getName()).log(Level.SEVERE, null, ex);
558.
559.          finally
560.          {
561.              try {
562.                  inputFile.close();
563.              } catch (IOException ex) {
564.
565.                  Logger.getLogger(Form.class.getName()).log(Level.SEVERE, null, ex);
566.              }
567.          }
568.
569.          private void ProsesLVQ()
570.          {
571.              // menentukan awal bobot dari 2 data pertama
572.
573.              for (int i = 0; i < a; i++)
574.              {
575.                  kls0 = 0.0f;

```

```

576.         kls1 = 0.0f;
577.
578.         if(i >= 2)
579.         {
580.             // proses menghitung kls 0
581.             for (int x = 0; x < 3; x++)
582.             {
583.                 float temp1, temp2;
584.                 temp1 = simpanData[i][x];
585.                 temp2 = simpanData[0][x];
586.                 tampungkls0[x] = temp1 - temp2;
587.                 kls0 += (float) Math.pow((temp1 -
temp2) , pangkat) ;
588.                 //System.out.println(kls0) ;
589.             }
590.             //proses menghitung kls 1
591.             System.out.println("-----
-----");
592.             for (int x = 0; x < 3; x++)
593.             {
594.                 float temp1, temp2;
595.                 temp1 = simpanData[i][x];
596.                 temp2 = simpanData[1][x];
597.                 tampungkls1[x] = temp1 - temp2;
598.                 kls1 += (float) Math.pow((temp1 -
temp2) , pangkat) ;
599.                 //System.out.println(kls1) ;
600.             }
601.             System.out.println("-----
-----");
602.
603.             kls0 = (float) Math.sqrt(kls0) ;
604.             kls1 = (float) Math.sqrt(kls1) ;
605.
606.             // proses mencari nilai terkecil
607.             updateBobot(i) ;
608.         //         b = b + 1;
609.         }
610.     }
611.     //         System.out.println(kls0) ;
612.     //         System.out.println(kls1) ;
613.
614. }
615.
616. private void updateBobot(int i)
617. {
618.     // mencari nilai terkecil dan update bobot baru
619.     if(kls0 <= kls1)
620.     {
621.         kls0kecil = true;
622.         w1 = simpanData[0][0] + (
learnRate*tampungkls0[0] );
623.         w2 = simpanData[0][1] + (
learnRate*tampungkls0[1] );
624.         w3 = simpanData[0][2] + (
learnRate*tampungkls0[2] );

```



```

625.      //          w4 = simpanData[0][3] + (
learnRate*tampungkls0[3] );
626.
627.          simpanData[0][0] = w1;
628.          simpanData[0][1] = w2;
629.          simpanData[0][2] = w3;
630.      //          if(simpanData[i][3] == 0)
631.      //          {
632.      //              simpanData[0][0] = w1;
633.      //              simpanData[0][1] = w2;
634.      //              simpanData[0][2] = w3;
635.      //          //          simpanData[0][3] = w4;
636.      //
637.      //          System.out.println("bener kls 0");
638.      //          }
639.      //      }
640.      //      else
641.      //      {
642.          w1 = simpanData[1][0] + (
learnRate*tampungkls1[0] );
643.          w2 = simpanData[1][1] + (
learnRate*tampungkls1[1] );
644.          w3 = simpanData[1][2] + (
learnRate*tampungkls1[2] );
645.      //          w4 = simpanData[1][3] + (
learnRate*tampungkls1[3] );
646.
647.          simpanData[1][0] = w1;
648.          simpanData[1][1] = w2;
649.          simpanData[1][2] = w3;
650.      //          if(simpanData[i][3] == 1)
651.      //          {
652.      //              simpanData[1][0] = w1;
653.      //              simpanData[1][1] = w2;
654.      //              simpanData[1][2] = w3;
655.      //          ///          simpanData[1][3] = w4;
656.      //          System.out.println("bener kls 1");
657.      //          }
658.      //      }
659.
660.      //          System.out.println("0 == "+kls0 + " 1 == " +
kls1);
661.      //          System.out.println("kls0 " + tampungkls0[0] + "
"+ tampungkls0[1] + "  "+ tampungkls0[2] + "  " + tampungkls0[3]);
662.      //          System.out.println("==" + kls0);
663.      //          System.out.println("kls1 " + tampungkls1[0] + "
"+ tampungkls1[1] + "  "+ tampungkls1[2] + "  " + tampungkls1[3]);
664.      //          System.out.println("==" + kls1);
665.      //          System.out.println("");
666.      //          System.out.println("bobot 0");
667.      //          System.out.println(simpanData[0][0]);
668.      //          System.out.println(simpanData[0][1]);
669.      //          System.out.println(simpanData[0][2]);
670.      //          ///          System.out.println(simpanData[0][3]);
671.      //          System.out.println("");
672.      //          System.out.println("bobot 1");

```

```

673. //      System.out.println(simpanData[1][0]);
674. //      System.out.println(simpanData[1][1]);
675. //      System.out.println(simpanData[1][2]);
676. ////      System.out.println(simpanData[1][3]);
677. //      System.out.println("");
678.     }
679.
680.     private String getColorHSV(Color color)
681.     {
682.         float[] comp = new float[3];
683.         Color.RGBtoHSB(color.getRed(), color.getGreen(),
color.getBlue(), comp);
684.         comp[0]*= 360;
685.         comp[1]*= 100;
686.         comp[2]*= 100;
687.         return comp[0]+" "+comp[1]+" "+comp[2];
688.     }
689.
690.     public static BufferedImage resize(BufferedImage img,
int newW, int newH)
691.     {
692.         Image tmp = img.getScaledInstance(newW, newH,
Image.SCALE_SMOOTH);
693.         BufferedImage dimg = new BufferedImage(newW, newH,
BufferedImage.TYPE_INT_RGB);
694.
695.         Graphics2D g2d = dimg.createGraphics();
696.         g2d.drawImage(tmp, 0, 0, null);
697.         g2d.dispose();
698.
699.         return dimg;
700.     }
701.
702.     public static Color averageColor(BufferedImage bi, int
x0, int y0, int w, int h)
703.     {
704.         int x1 = x0 + w;
705.         int y1 = y0 + h;
706.         long sumr = 0, sumg = 0, sumb = 0;
707.         for (int x = x0; x < x1; x++) {
708.             for (int y = y0; y < y1; y++) {
709.                 Color pixel = new Color(bi.getRGB(x, y));
710.                 sumr += pixel.getRed();
711.                 sumg += pixel.getGreen();
712.                 sumb += pixel.getBlue();
713.             }
714.         }
715.         int num = w * h;
716.         return new Color(sumr / num, sumg / num, sumb /
num);
717.     }
718.
719.     public ImageIcon ResizeImage(String ImagePath)
720.     {
721.         System.out.println(ImagePath);
722.         ImageIcon MyImage = new ImageIcon(ImagePath);

```

```

723.         Image img = MyImage.getImage();
724.         Image newImg =
img.getScaledInstance(jLabel1.getWidth(), JLabel1.getHeight(),
Image.SCALE_SMOOTH);
725.         ImageIcon image = new ImageIcon(newImg);
726.
727.         return image;
728.     }
729.
730.     public static void infoBox(String infoMessage, String
titleLabel)
731.     {
732.         JOptionPane.showMessageDialog(null, infoMessage,
titleLabel, JOptionPane.INFORMATION_MESSAGE);
733.     }
734.
735.     /**
736.      * @param args the command line arguments
737.      */
738.     public static void main(String args[]) {
739.         /* Set the Nimbus look and feel */
740.         //<editor-fold defaultstate="collapsed" desc=" Look
and feel setting code (optional) ">
741.         /* If Nimbus (introduced in Java SE 6) is not
available, stay with the default look and feel.
742.          * For details see
http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
743.          */
744.         try {
745.             for (javax.swing.UIManager.LookAndFeelInfo info
: javax.swing.UIManager.getInstalledLookAndFeels()) {
746.                 if ("Nimbus".equals(info.getName())) {
747.                     javax.swing.UIManager.setLookAndFeel(info.getClassName());
748.                     break;
749.                 }
750.             }
751.         } catch (ClassNotFoundException ex) {
752.             java.util.logging.Logger.getLogger(Form.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
753.         } catch (InstantiationException ex) {
754.             java.util.logging.Logger.getLogger(Form.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
755.         } catch (IllegalAccessException ex) {
756.             java.util.logging.Logger.getLogger(Form.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
757.         } catch
(javax.swing.UnsupportedLookAndFeelException ex) {
758.             java.util.logging.Logger.getLogger(Form.class.getName()).log(java.ut
il.logging.Level.SEVERE, null, ex);
759.         }

```

```
760.         //</editor-fold>
761.
762.         /* Create and display the form */
763.         java.awt.EventQueue.invokeLater(new Runnable() {
764.             public void run() {
765.                 new Form().setVisible(true);
766.             }
767.         });
768.     }
769.
770.     // Variables declaration - do not modify
771.     private javax.swing.JTextField Bvalue;
772.     private javax.swing.JTextField Gvalue;
773.     private javax.swing.JTextField Hvalue;
774.     private javax.swing.JTextField Rvalue;
775.     private javax.swing.JTextField Svalue;
776.     private javax.swing.JTextField Vvalue;
777.     private javax.swing.JButton btBrowse;
778.     private javax.swing.JButton btLatih;
779.     private javax.swing.JButton btProses;
780.     private javax.swing.JLabel jLabel1;
781.     private javax.swing.JLabel jLabel10;
782.     private javax.swing.JLabel jLabel2;
783.     private javax.swing.JLabel jLabel3;
784.     private javax.swing.JLabel jLabel4;
785.     private javax.swing.JLabel jLabel5;
786.     private javax.swing.JLabel jLabel6;
787.     private javax.swing.JLabel jLabel7;
788.     private javax.swing.JLabel jLabel8;
789.     private javax.swing.JLabel jLabel9;
790.     private javax.swing.JLabel jLabelCrop;
791.     private javax.swing.JPanel jPanel1;
792.     private javax.swing.JPanel jPanel2;
793.     private javax.swing.JPanel jPanel3;
794.     private javax.swing.JPanel jPanel4;
795.     private javax.swing.JLabel lblKesimpulan;
796.     private javax.swing.JPanel pnlGambar;
797.     private javax.swing.JTextField w0Value;
798.     private javax.swing.JTextField w1Value;
```



2.82% PLAGIARISM
APPROXIMATELY

Report #12352083

CHAPTER I INTRODUCTION 1.1 Background Digital Image processing is the process of processing an image, the results will be issued according to the wishes of the image processing itself. The background started when I was born, where it used to be is the public work of the people farming. From the guava garden to the corn field, the problem that always occurs is that the harvest results are wasted because the farmers do not know the texture and color of the ripe guava fruit. With this problem, it is hoped that this research will be able to help farmers classify and classify guava fruit that are ready to be picked based on the color and texture grade based on the research results. HSV defines color in terms of Hue, Saturation and Value. The advantage of HSV is that it contains the same colors that are captured by the human senses. Meanwhile, the colors formed by other models such as RGB are the result of a mixture of primary colors. The results of color processing will be data sets that are processed using the Learning Vector Quantization classification method in order to identify the ripeness of the guava fruit. Digital image processing and the Learning Vector Quantization (LVQ) classification method are implemented in the form of display data. This research is expected to provide solutions to the problems that exist in the processing of plantation products. Especially in the process of selecting