APPENDIX

CODE MAIN PROGRAM INPUT DAN OUTPUT IMAGE

```
91.     public class Main {
92.       static BufferedImage joinedImg;
93.
94.
95.       public static void main(String args[]) throws Exception
  {
96.     BufferedImage            img1=            ImageIO.read(new
  File("foto/11.jpeg"));
97.     BufferedImage            img2=            ImageIO.read(new
  File("foto/13.jpeg"));
98.     BufferedImage            img3=            ImageIO.read(new
  File("foto/12.jpeg"));
99.     BufferedImage            img4=            ImageIO.read(new
  File("foto/14.jpeg"));
100.    BufferedImage            img5=            ImageIO.read(new
  File("foto/15.jpeg"));
101.
102.       int [] w1 = {2,3,1,4,1,1,1,2,1};
103.       int [] w2 = {2,1,2,1,3,1,2,2,1};
104.       int [] w3 = {2,2,2,2,3,2,3,2,1};
105.       int [] w4 = {2,2,2,1,2,2,3,1,4};
106.       int [] w5 = {2,2,3,2,1,4,1,3,3};
107.
108.       for(int i=0; i<w1.length; i++ )
109.       {
110.    joinedImg=JoinImage.joinBufferedImage(  img1,img2,img3,img
  4,img5,  w1[i], w2[i], w3[i],w4[i],w5[i]);
111.    ImageIO.write(joinedImg, "png", new File("fotohasil/2fto/
  f2"+i+".png"));
```

CODE METHOD PROGRAM

```
112.    public class JoinImage {
113.      public JoinImage(){
114.      }
115.       public  static  ImgARGB  PixelARGB(int  x,  int  y,  int
  pixel) {
116.         int alpha = (pixel >> 24) & 0xff;
117.         int red = (pixel >> 16) & 0xff;
118.         int green = (pixel >> 8) & 0xff;
119.         int blue = (pixel) & 0xff;
120.         return new ImgARGB(x, y, alpha, red, green, blue);
121.       }
122.    public                static                BufferedImage
  joinBufferedImage(BufferedImage              img1,BufferedImage
  img2,BufferedImage img3, BufferedImage img4,BufferedImage img5,
  int w1, int w2, int w3, int w4, int w5) {
```

```
123.     int width = img1.getWidth() ;
124.     int height = img1.getHeight() ;
125.     int min_r=255,min_g=255,min_b=255;
126.     int max_r=0,max_g=0,max_b=0;
127.     BufferedImage   newImage   =   new   BufferedImage(width,
  height,BufferedImage.TYPE_INT_ARGB);
128.      int s=0;
129.      for(int i=0; i<height; i++){
130.      for(int j=0; j<width; j++){
131.      int pixel  =  img1.getRGB(j, i);
132.      int pixel2 =  img2.getRGB(j, i);
133.      int pixel3 =  img3.getRGB(j, i);
134.      int pixel4 =  img4.getRGB(j, i);
135.      int pixel5 =  img5.getRGB(j, i);
136.      ImgARGB imgrg1 =  PixelARGB(j,i , pixel)  ;
137.      ImgARGB imgrg2 =  PixelARGB(j,i , pixel2) ;
138.      ImgARGB imgrg3 =  PixelARGB(j,i , pixel3) ;
139.      ImgARGB imgrg4 =  PixelARGB(j,i , pixel4) ;
140.      ImgARGB imgrg5 =  PixelARGB(j,i , pixel5) ;
141.      int r = (( w1*imgrg1.getRed()) + (w2* imgrg2.getRed()) +
  (w3      *      imgrg3.getRed())+      (w4      *imgrg4.getRed())+
  (w5*imgrg5.getRed()))/10;
142.      if(r>max_r)
143.       {
144.        max_r=r;
145.        }
146.      if(r<min_r)
147.       {
148.         min_r=r;
149.       }
150.     int g = ((w1*imgrg1.getGreen()) + (w2* imgrg2.getGreen())
  +   (w3   *imgrg3.getGreen())   +(w4   *imgrg4.getGreen())   +
  (w5*imgrg5.getGreen()))/10;
151.     if(g>max_g)
152.     {
153.      max_g=g;
154.     }
155.     if(g<min_g)
156.     {
157.     min_g=g;
158.     }
159.     int b = (( w1*imgrg1.getBlue())+ (w2* imgrg2.getBlue()) +
  (w3    *imgrg3.getBlue())    +    (w4    *imgrg4.getBlue())    +
  (w5*imgrg5.getBlue()))/10;
160.     if(b>max_b)
161.     {
162.     max_b=b;
163.     }
164.     if(b<min_b)
165.     {
166.       min_b=b;
167.     }
168.     s = s+r+g+b;
169.     Color rgb =   new Color((int) r, (int) g, (int) b, 255);
```

```
170.     newImage.setRGB(j, i, rgb.getRGB());
171.      }
172.
173.     }
174.                 int luas= height*width;
175.                 int brg = s/(3*luas);
176.                 int br = brg / 3;
177.                 int b = (br*100) / 255;
178.                 System.out.println("brightness= " +b);
179.
180.                 int maxpxl = max_r + max_g + max_b;
181.                 int minpxl = min_r + min_g + min_b;
182.                 int contras = maxpxl - minpxl;
183.                 int ct = contras / 3;
184.                 int ctrs = (ct*100)/255;
185.                 System.out.println("contras= "+ctrs);
186.
187.                 return newImage;
```

UNICHECK

Submission author:
15k10052 NICK ANTONA RICKY RISMA NUGRAH...

Check ID:
15997994

Check date:
17.01.2020 14:57:39 GMT+0

Check type:
Doc vs Internet + Library

Report date:
17.01.2020 23:30:10 GMT+0

User ID:
29120

Unika SOEGIJAPRANATA

File name: 15.k1.0052_Nickantona Ricky R N.docx

File ID: 20300319 Page count: 8 Word count: 2883 Character count: 16302 File size: 17.20 KB

0.87% Matches

Highest match: 0.31% with source https://www.oznium.com/blog/how-to-wire-4-pin-led-switch

0.87% Internet Matches 7 ..... Page 10

No Library Sources Found

0.42% Quotes

Quotes 1 ..... Page 11

No references found

0% Exclusions

No exclusions found

Replacement

No replaced characters found