

OBJECT TRACKING BASED ON RGB
PENGOLAHAN CITRA
MK401

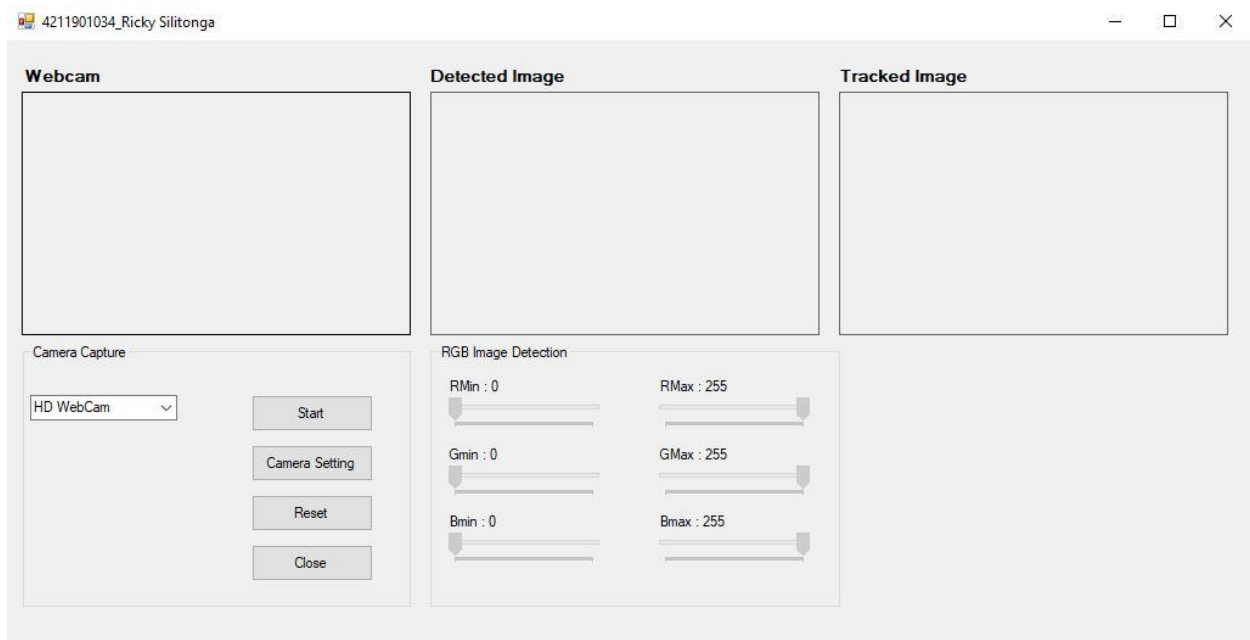


POLITEKNIK NEGERI Batam

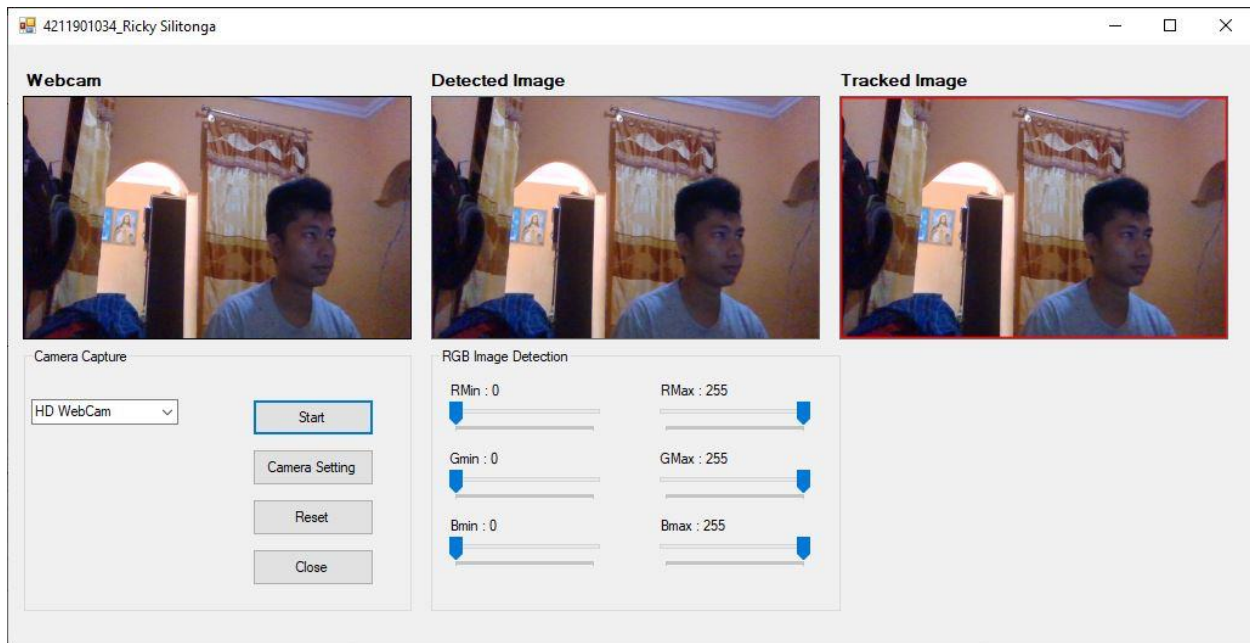
Disusun oleh :
Ricky Silitonga (4211901034)

PROGRAM STUDI TEKNIK MEKATRONIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI BATAM
2021

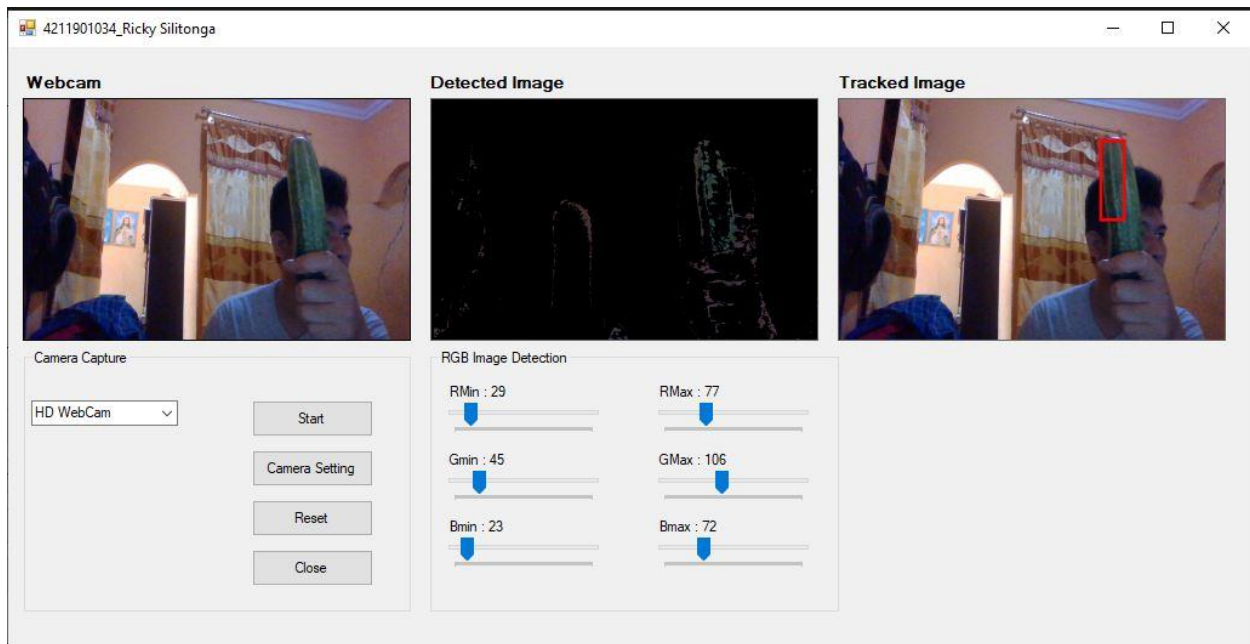
OBJECT TRACKING BASED ON RGB COLOR IMAGE FILTERING USING CAMERA



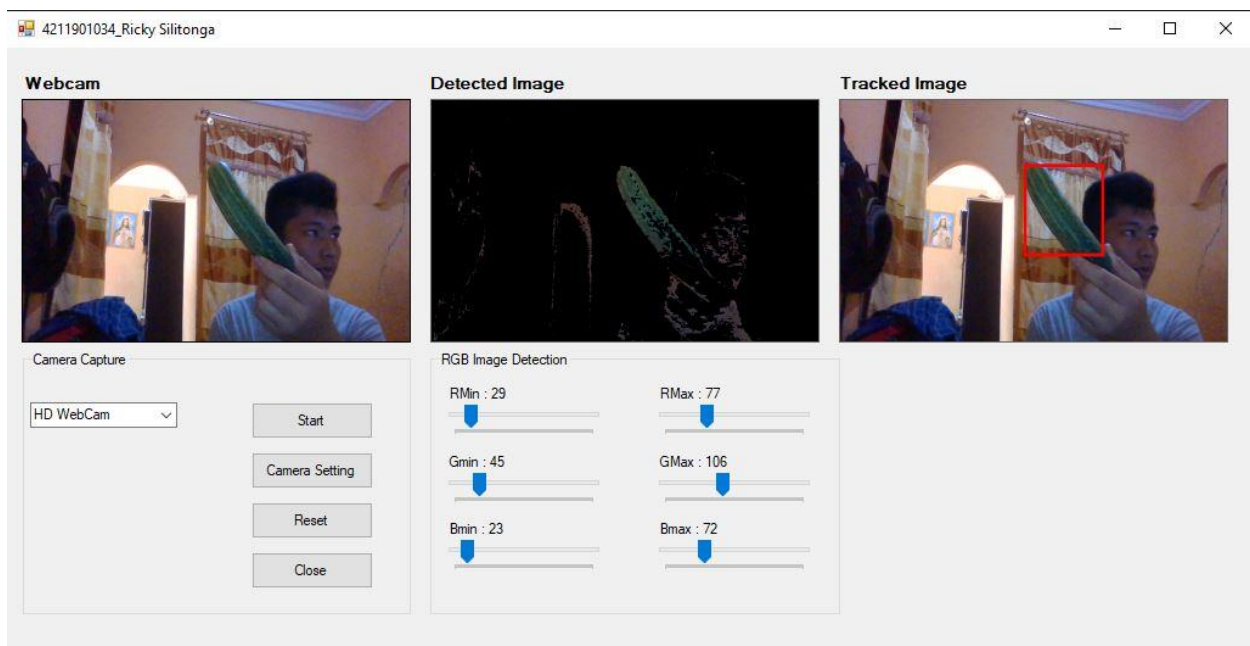
Gambar 1. Tampilan awal program



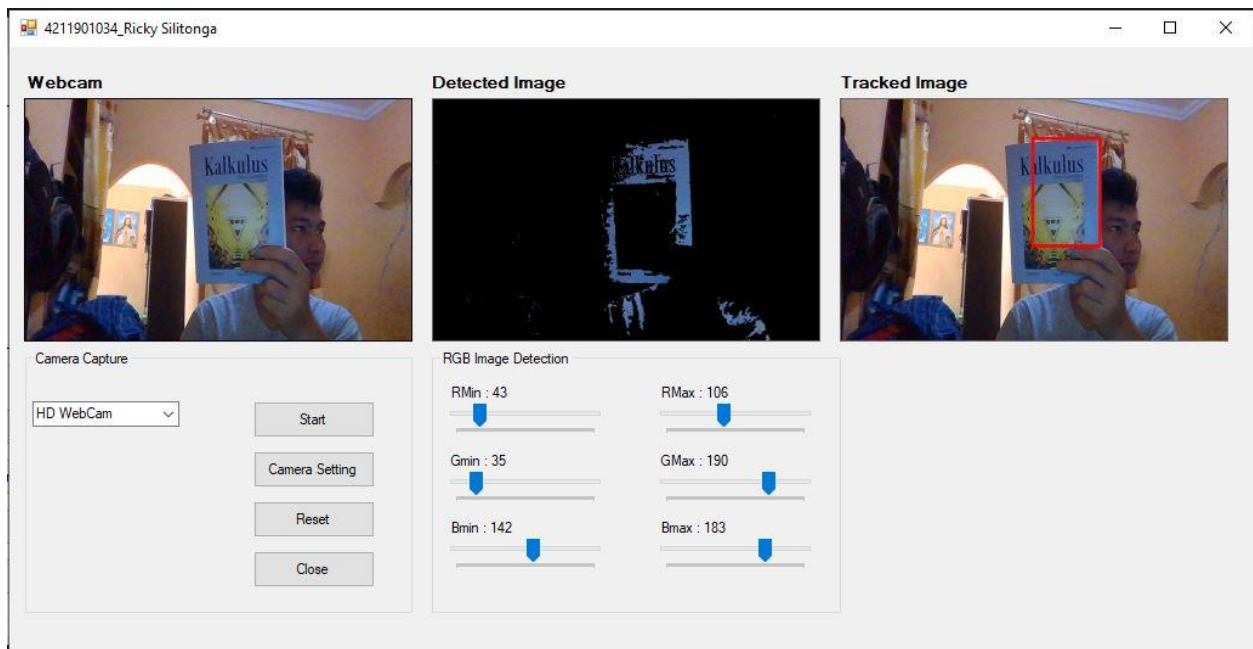
Gambar 2. Tombol start di klik



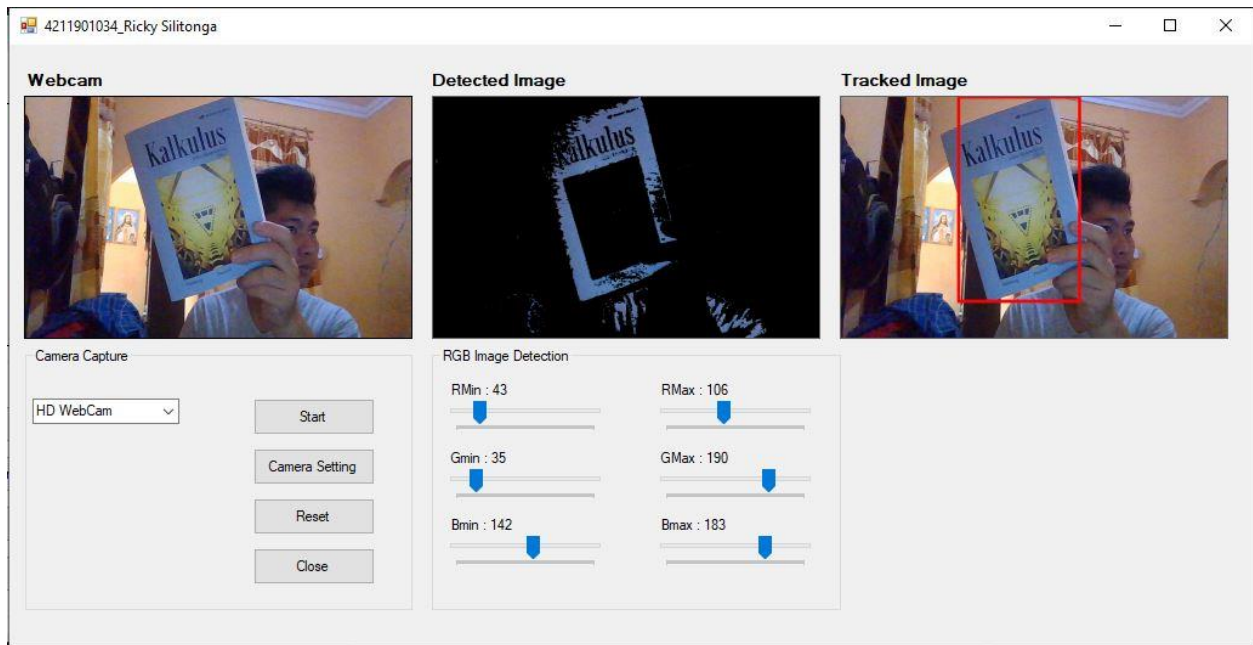
Gambar 3.1. Melakukan tracking pada objek yaitu buah timun



Gambar 3.2. Melakukan tracking pada objek yaitu buah timun



Gambar 4.1. Melakukan tracking pada objek yaitu buku



Gambar 4.2. Melakukan tracking pada objek yaitu buku

Sourcecode

```
4211901034.cs 4211901034 [Design]
Pertemuan10_4211901034
Form1_Load(object sender, EventArgs e)

1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 // libraries
11 using AForge;
12 using AForge.Imaging;
13 using AForge.Imaging.Filters;
14 using AForge.Video;
15 using AForge.Video.DirectShow;
16 using System.Collections;
17
18 namespace Pertemuan10_4211901034
19 {
20     3 references
21     public partial class Form1 : Form
22     {
23         private FilterInfoCollection videoDevices;
24         private VideoCaptureDevice videoDevice;
25         private ArrayList listCamera = new ArrayList();
26         //image variabel
27         Bitmap sourceImage = null;
28         Bitmap detectedImage = null;
29         //trackbar variabel
30         int Rmin, Rmax, Gmin, Gmax, Bmin, Bmax;
31         int TRACK_SPACE = 2; //space between trackBar
32
33         1 reference
34         public Form1()
35         {
36             InitializeComponent();
37             trackBarEnable(false);
38             trackBarReset();
39             labelReset();
40
41             //RGB trackbar Enable
42             2 references
43             private void trackBarEnable(bool enable = true)
44             {
45                 //RGB trackbar Enable
46                 trackBarRmax.Enabled = enable;
47                 trackBarRmin.Enabled = enable;
48                 trackBarGmax.Enabled = enable;
49                 trackBarGmin.Enabled = enable;
50                 trackBarBmax.Enabled = enable;
51                 trackBarBmin.Enabled = enable;
52             }
53         }
54     }
55 }
```

```
4211901034.cs 4211901034 [Design]
Pertemuan10_4211901034
Form1_Load(object sender, EventArgs e)

49 trackBarBmin.Enabled = enable;
50 } private void trackBarEnable(bool enable = true)
51 2 references
52 private void trackBarReset()
53 {
54     //RGB trackbar init
55     trackBarRmax.Maximum = 255;
56     trackBarRmin.Maximum = 255;
57     trackBarGmax.Maximum = 255;
58     trackBarGmin.Maximum = 255;
59     trackBarBmax.Maximum = 255;
60     trackBarBmin.Maximum = 255;
61     //RGB trackbar reset
62     trackBarRmax.Value = 255;
63     trackBarRmin.Value = 0;
64     trackBarGmax.Value = 255;
65     trackBarGmin.Value = 0;
66     trackBarBmax.Value = 255;
67     trackBarBmin.Value = 0;
68     Rmin = trackBarRmin.Value;
69     Rmax = trackBarRmax.Value;
70     Gmin = trackBarGmin.Value;
71     Gmax = trackBarGmax.Value;
72     Bmin = trackBarBmin.Value;
73     Bmax = trackBarBmax.Value;
74 }
75 //RGB label reset
76 2 references
77 private void labelReset()
78 {
79     labelRmax.Text = string.Format("Rmax : {0}", trackBarRmax.Value);
80     labelRmin.Text = string.Format("Rmin : {0}", trackBarRmin.Value);
81     labelGmax.Text = string.Format("Gmax : {0}", trackBarGmax.Value);
82     labelGmin.Text = string.Format("Gmin : {0}", trackBarGmin.Value);
83     labelBmax.Text = string.Format("Bmax : {0}", trackBarBmax.Value);
84     labelBmin.Text = string.Format("Bmin : {0}", trackBarBmin.Value);
85 }
86
87 1 reference
88 private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap image)
89 {
90     try
91     {
92         sourceImage = image.Clone() as Bitmap;
93         //detect the image
94         colorFiltering(sourceImage);
95         //traking the image
96         objectTracking(sourceImage);
97     }
98     catch
99     {
100     }
101 }
```



```
4211901034.cs 4211901034 [Design]
Pertemuan10_4211901034
Form1_Load(object sender, EventArgs e)

96 {
97     } catch
98 } % private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap image)
99
100 1 reference
101 private void colorFiltering(Bitmap srcImage)
102 {
103     // create RGB color filter
104     ColorFiltering filter = new ColorFiltering();
105     // set color ranges to keep
106     filter.Red = new InRange(Rmin, Rmax);
107     filter.Green = new InRange(Gmin, Gmax);
108     filter.Blue = new InRange(Bmin, Bmax);
109     // apply the filter
110     detectedImage = filter.Apply(srcImage);
111     //draw the picture
112     pictureBox1.Image = detectedImage;
113 }
114 1 reference
115 private void objectTracking(Bitmap srcImage)
116 {
117     if (srcImage == null || detectedImage == null) return;
118     //copy detected image to the new one
119     Bitmap newImage = (Bitmap)detectedImage.Clone();
120     //blob counter on the detected image
121     BlobCounter bc = new BlobCounter();
122     bc.MinHeight = 20;
123     bc.MinWidth = 20;
124     bc.FilterBlobs = true;
125     bc.ObjectsOrder = ObjectsOrder.Area;
126     bc.ProcessImage(newImage);
127     Rectangle[] rects = bc.GetObjectsRectangles();
128     foreach (Rectangle recs in rects)
129     {
130         if (rects.Length > 0)
131         {
132             Rectangle objectRect = rects[0];
133             Graphics graph = Graphics.FromImage(srcImage);
134             using (Pen pen = new Pen(Color.FromArgb(255, 0, 0), 10))
135             {
136                 graph.DrawRectangle(pen, objectRect);
137             }
138             graph.Dispose();
139         }
140     }
141     //draw tracked object on picture box
142     pictureBox2.Image = srcImage;
143 }
144 private static string _usbcamera;
```

```
4211901034.cs 4211901034 [Design]
Pertemuan10_4211901034
Form1_Load(object sender, EventArgs e)

145 1 reference
146 private void Form1_Load(object sender, EventArgs e)
147 {
148     videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
149     if (videoDevices.Count != 0)
150     {
151         // add all devices to combo
152         foreach (FilterInfo device in videoDevices)
153         {
154             comboBox1.Items.Add(device.Name);
155         }
156     }
157     else
158     {
159         comboBox1.Items.Add("No DirectShow devices found");
160     }
161     comboBox1.SelectedIndex = 0;
162 }
163
164 2 references
165 public string _usbcamera
166 {
167     get { return _usbcamera; }
168     set { _usbcamera = value; }
169 }
170
171 // opening video source
172 1 reference
173 private void OpenVideoSource(TVideoSource source)
174 {
175     // set busy cursor
176     try
177     {
178         this.Cursor = Cursors.WaitCursor;
179         // stop current video source
180         CloseCurrentVideoSource();
181         // start new video source
182         videoSourcePlayer1.VideoSource = source;
183         videoSourcePlayer1.Start();
184         this.Cursor = Cursors.Default;
185     }
186     catch { }
187 }
188
189 // closing the video source
190 1 reference
191 public void CloseCurrentVideoSource()
192 {
193 }
```

```
4211901034.cs x 4211901034 [Design]
Pertemuan10_4211901034
Pertemuan10_4211901034.Form1
Form1_Load(object sender, EventArgs e)

192 try
193 {
194     if (videoSourcePlayer1.VideoSource != null)
195     {
196         videoSourcePlayer1.SignalToStop();
197         // wait ~ 3 seconds
198         for (int i = 0; i < 30; i++)
199         {
200             if (!videoSourcePlayer1.IsRunning)
201                 break;
202             System.Threading.Thread.Sleep(100);
203         }
204         if (videoSourcePlayer1.IsRunning)
205         {
206             videoSourcePlayer1.Stop();
207         }
208         videoSourcePlayer1.VideoSource = null;
209     }
210 } catch { }
211 }
212 # public void CloseCurrentVideoSource()
213 {
214     1 reference
215     private void OpenCamera()
216     {
217         try
218         {
219             usbCamera = comboBox1.SelectedIndex.ToString();
220             VideoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
221             if (VideoDevices.Count != 0)
222             {
223                 // add all devices to combo
224                 foreach (FilterInfo device in VideoDevices)
225                 {
226                     listCamera.Add(device.Name);
227                 }
228             }
229             else
230             {
231                 MessageBox.Show("Camera devices found");
232             }
233             videoDevice = new VideoCaptureDevice(VideoDevices[Convert.ToInt32(usbCamera)].MonikerString);
234             OpenVideoSource(videoDevice);
235         } catch (Exception err)
236         {
237             MessageBox.Show(err.ToString());
238         }
239     }
240 }
79 % No issues found Ln: 147 Ch: 13 SPC CRLF
```

```
4211901034.cs x 4211901034 [Design]
Pertemuan10_4211901034
Pertemuan10_4211901034.Form1
Form1_Load(object sender, EventArgs e)

242 } # private void OpenCamera()
243 {
244     1 reference
245     private void button1_Click(object sender, EventArgs e)
246     {
247         OpenCamera();
248         trackBar1.Enabled();
249     }
250     1 reference
251     private void Form1_FormClosed(object sender, FormClosedEventArgs e)
252     {
253         if (videoDevice != null && videoDevice.IsRunning)
254             videoDevice.Stop();
255     }
256     1 reference
257     private void trackBarRmin_Scroll(object sender, EventArgs e)
258     {
259         if (trackBarRmax.Value - trackBarRmin.Value <= TRACK_SPACE)
260             trackBarRmin.Value = trackBarRmax.Value - TRACK_SPACE;
261         Rmin = trackBarRmin.Value;
262         labelRmin.Text = string.Format("RMin : {0}", Rmin);
263     }
264     1 reference
265     private void trackBarRmax_Scroll(object sender, EventArgs e)
266     {
267         if (trackBarRmax.Value - trackBarRmin.Value <= TRACK_SPACE)
268             trackBarRmax.Value = trackBarRmin.Value + TRACK_SPACE;
269         Rmax = trackBarRmax.Value;
270         labelRmax.Text = string.Format("RMax : {0}", Rmax);
271     }
272     1 reference
273     private void trackBarGmin_Scroll(object sender, EventArgs e)
274     {
275         if (trackBarGmax.Value - trackBarGmin.Value <= TRACK_SPACE)
276             trackBarGmin.Value = trackBarGmax.Value - TRACK_SPACE;
277         Gmin = trackBarGmin.Value;
278         labelGmin.Text = string.Format("Gmin : {0}", Gmin);
279     }
280     1 reference
281     private void trackBarGmax_Scroll(object sender, EventArgs e)
282     {
283         if (trackBarGmax.Value - trackBarGmin.Value <= TRACK_SPACE)
284             trackBarGmax.Value = trackBarGmin.Value + TRACK_SPACE;
285         Gmax = trackBarGmax.Value;
286         labelGmax.Text = string.Format("GMax : {0}", Gmax);
287     }
79 % No issues found Ln: 147 Ch: 13 SPC CRLF
```

```
4211901034.cs X 4211901034.cs [Design]
Pertemuan10_4211901034 Pertemuan10_4211901034.Form1 Form1_Load(object sender, EventArgs e)

287
288 1 reference
289 private void trackBarBmin_Scroll(object sender, EventArgs e)
290 {
291     if (trackBarBmax.Value - trackBarBmin.Value <= TRACK_SPACE)
292     {
293         trackBarBmin.Value = trackBarBmax.Value - TRACK_SPACE;
294         Bmin = trackBarBmin.Value;
295         labelBmin.Text = string.Format("Bmin : {0}", Bmin);
296     }
297
298 1 reference
299 private void trackBarBmax_Scroll(object sender, EventArgs e)
300 {
301     if (trackBarBmax.Value - trackBarBmin.Value <= TRACK_SPACE)
302     {
303         trackBarBmax.Value = trackBarBmin.Value + TRACK_SPACE;
304         Bmax = trackBarBmax.Value;
305         labelBmax.Text = string.Format("Bmax : {0}", Bmax);
306     }
307
308 1 reference
309 private void button2_Click(object sender, EventArgs e)
310 {
311     if ((videoDevice != null))
312     {
313         try
314         {
315             ((VideoCaptureDevice)videoDevice).DisplayPropertyPage(this.Handle);
316         }
317         catch (NotSupportedException ex)
318         {
319             MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
320                             MessageBoxIcon.Error);
321         }
322     }
323
324 1 reference
325 private void button3_Click(object sender, EventArgs e)
326 {
327     trackBarReset();
328     labelReset();
329 }
330
331 1 reference
332 private void button4_Click(object sender, EventArgs e)
333 {
334     Close();
335 }
336
337 }
338 public partial class Form1 : Form
339 {
340     namespace Pertemuan10_4211901034
341     {
342     }
```