CAMERA CAPTURE BASED ON AFORGE

PENGOLAHAN CITRA

MK401

POLITEKNIK NEGERI Batam

Disusun oleh :

Ricky Silitonga (4211901034)

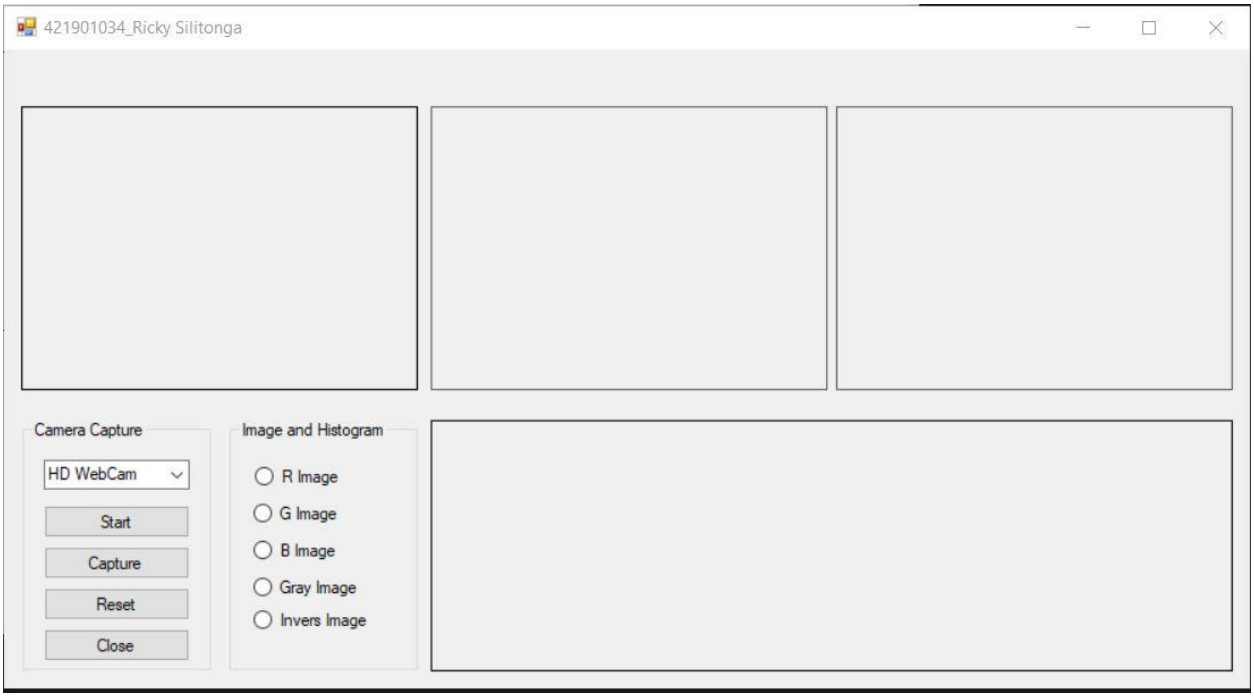
PROGRAM STUDI TEKNIK MEKATRONIKA

JURUSAN TEKNIK ELEKTRO

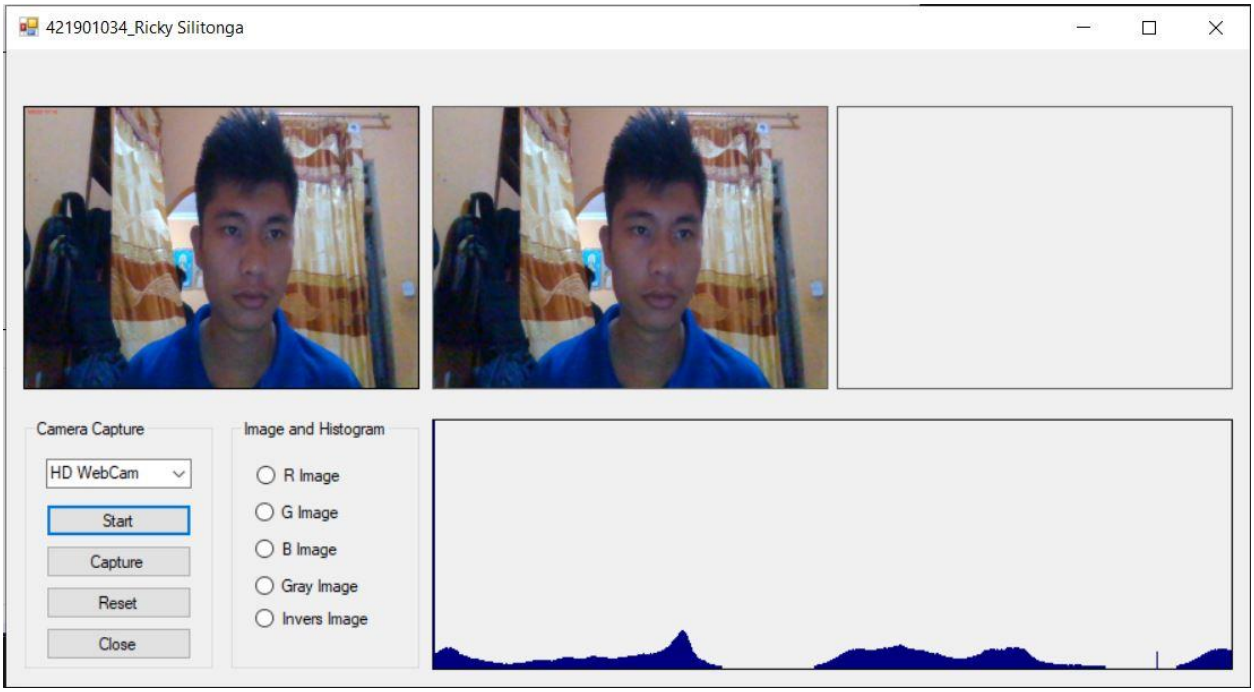
POLITEKNIK NEGERI BATAM

2021

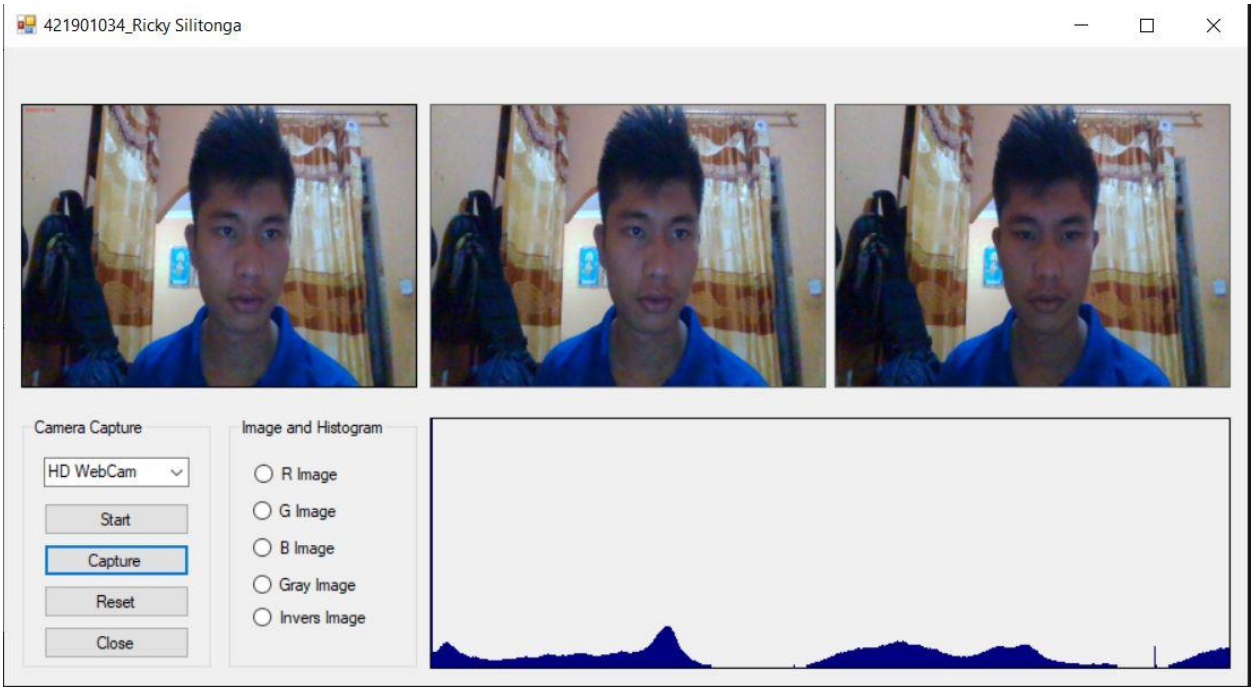
Introduction To Camera Capture Based On Aforge



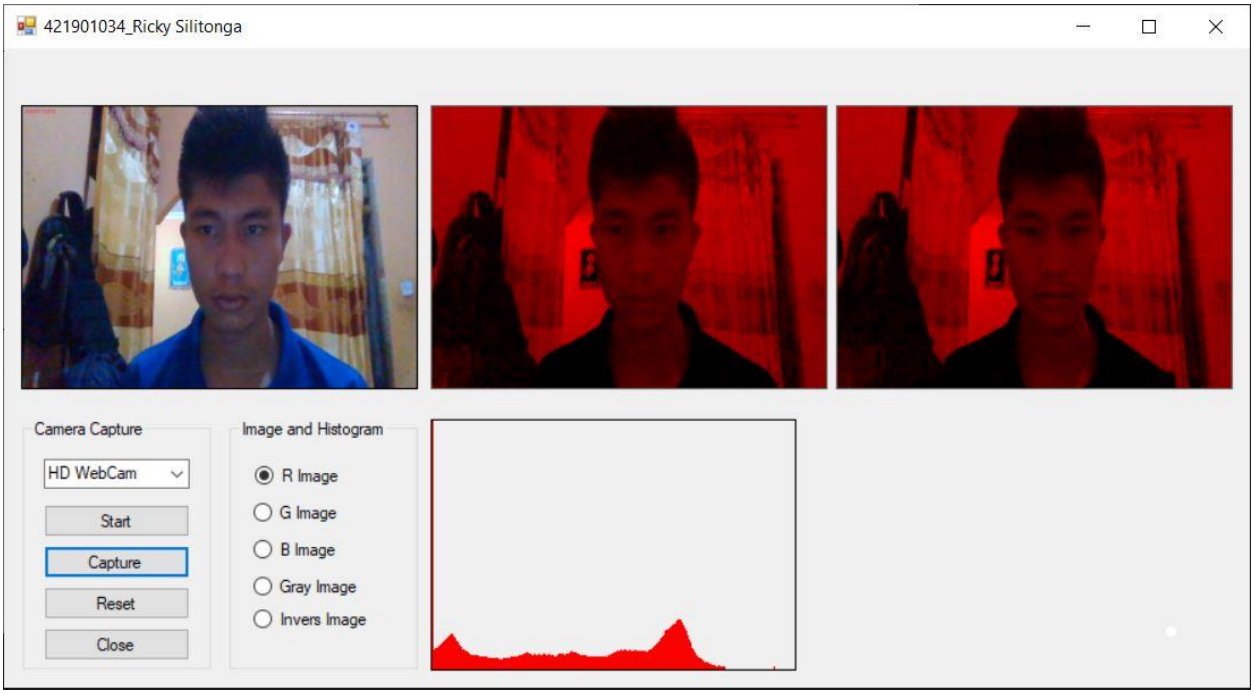
Gambar 1. Tampilan Awal



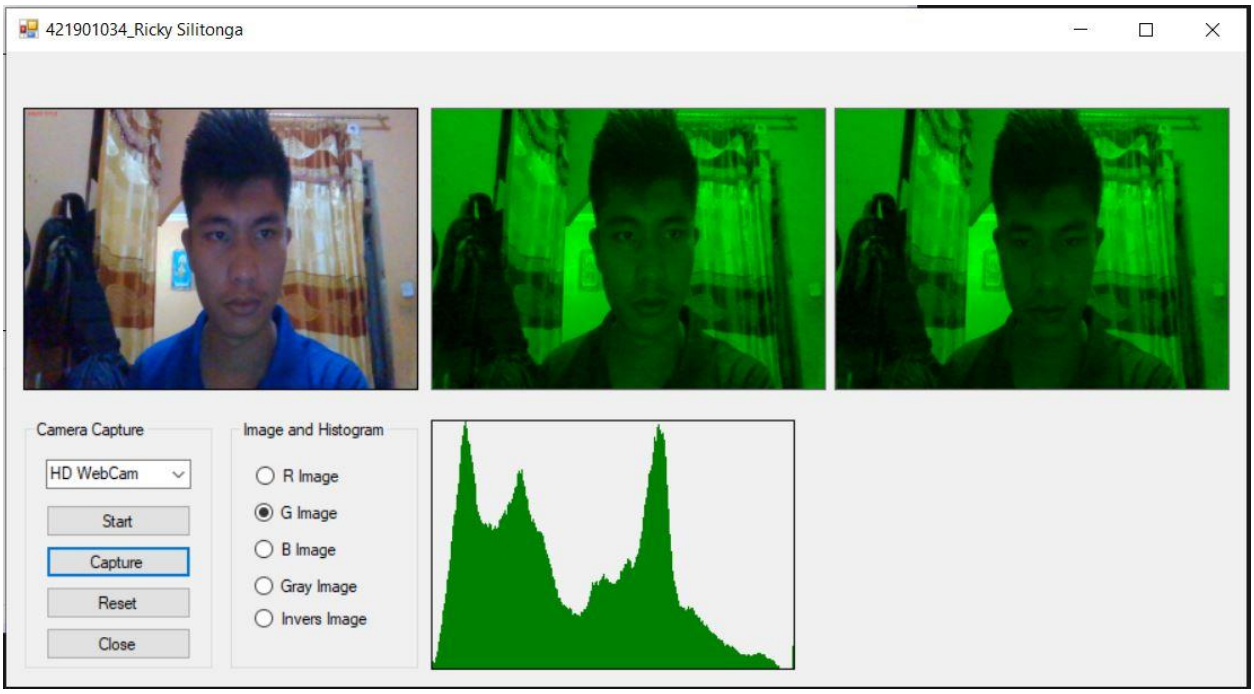
Gambar 2. Start Webcam



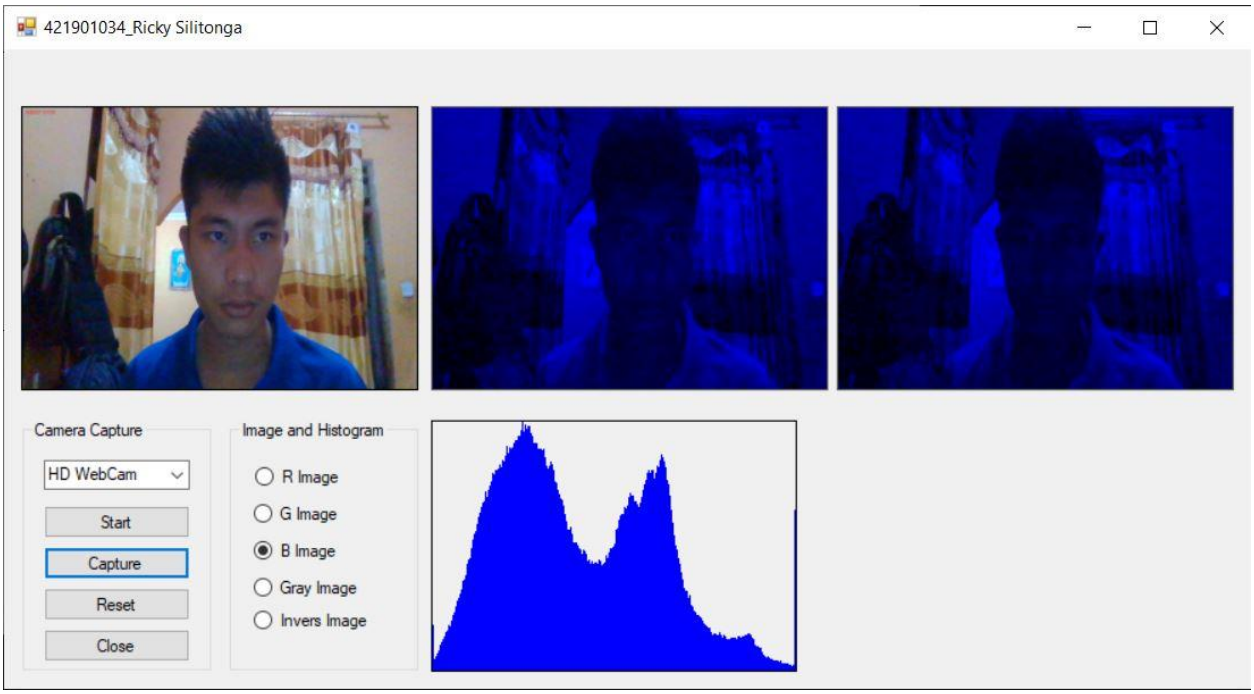
Gambar 3. Tombol Capture di klik



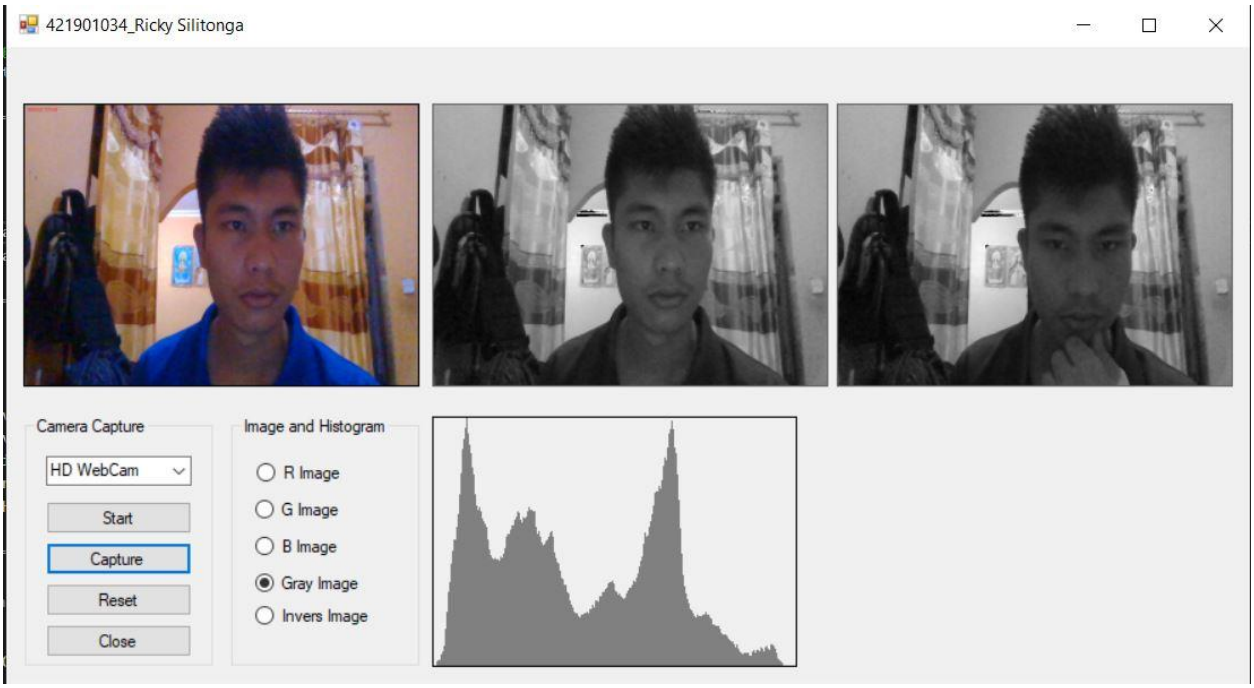
Gambar 4. R Image + Capture



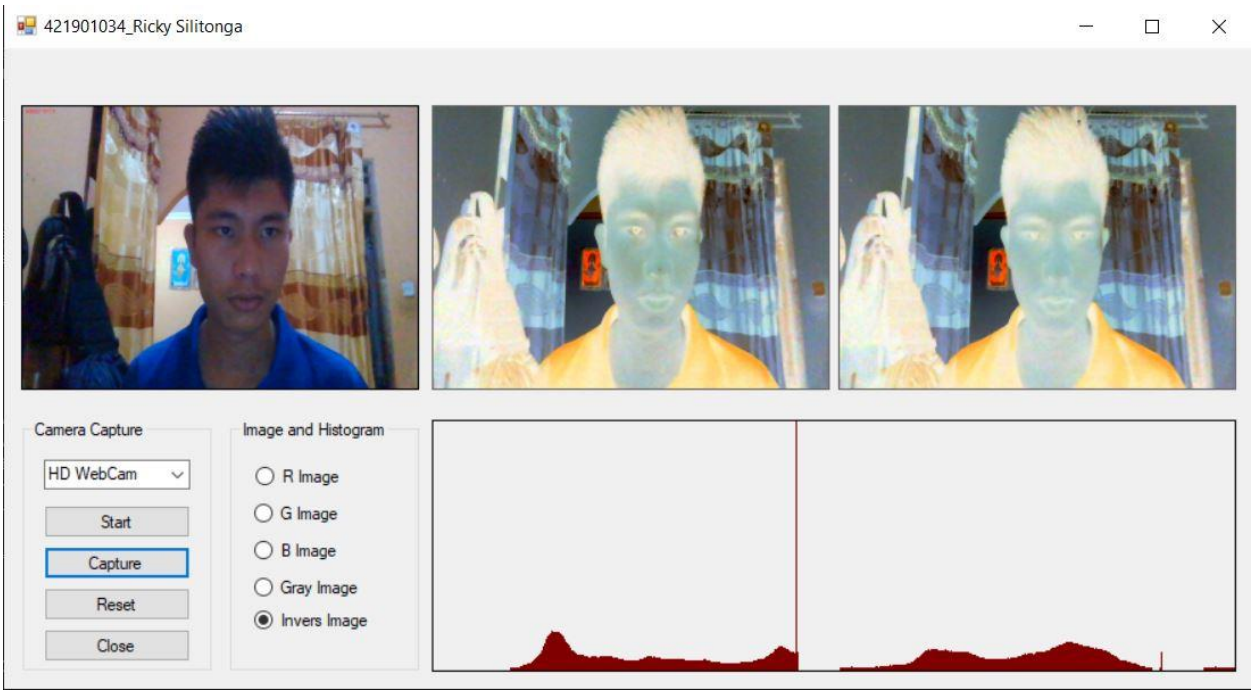
Gambar 5. G Image + Capture



Gambar 6. B Image + Capture

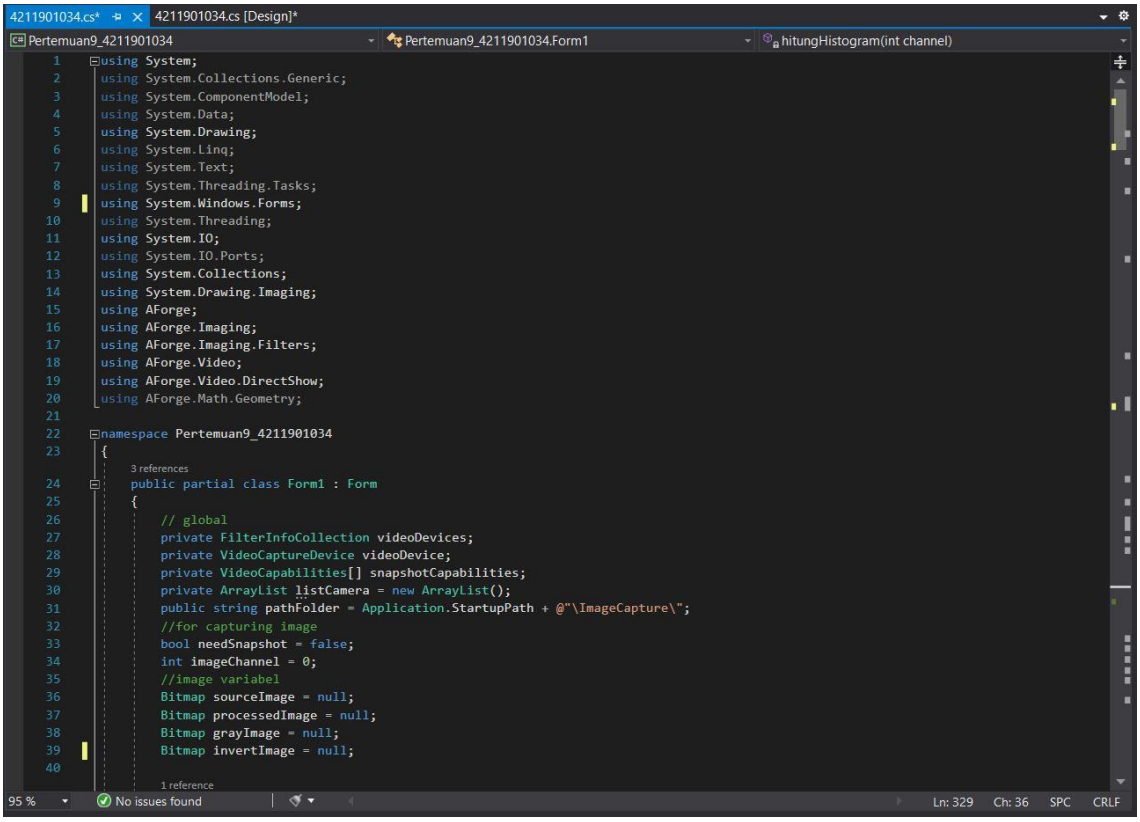


Gambar 7. Gray Image + Capture

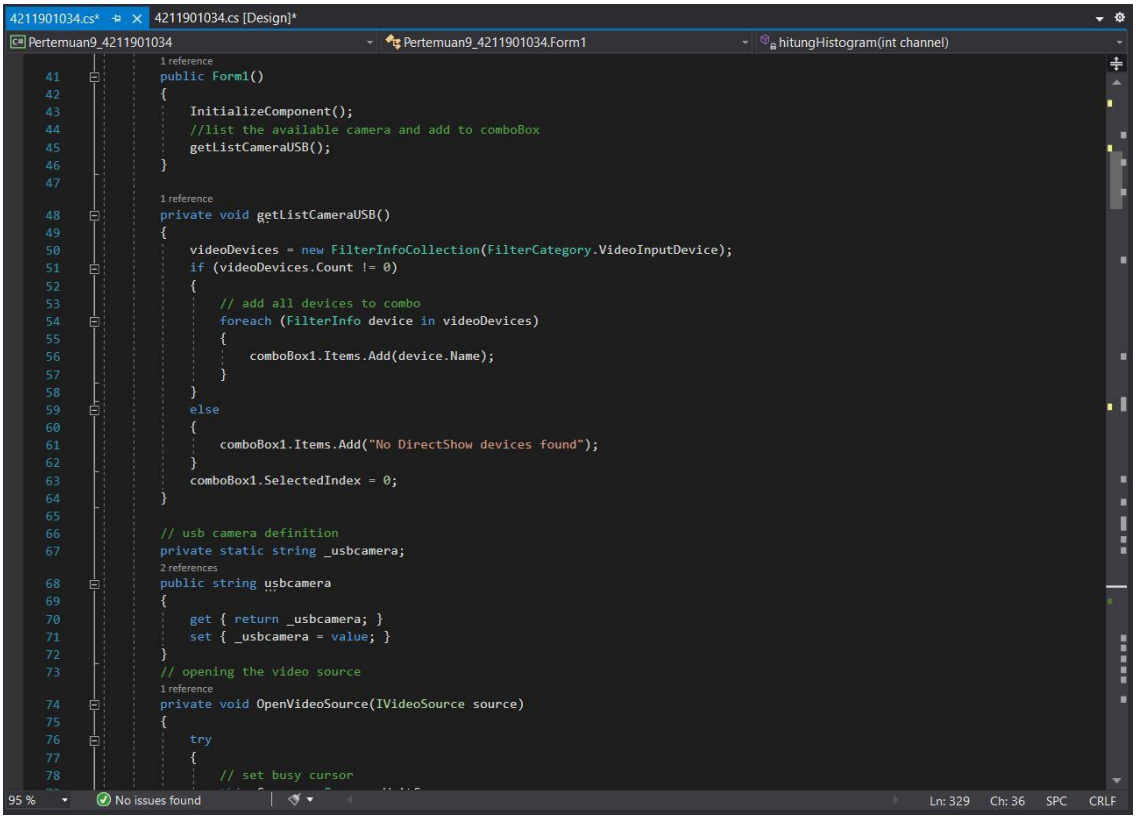


Gambar 8. Invers Image + Capture

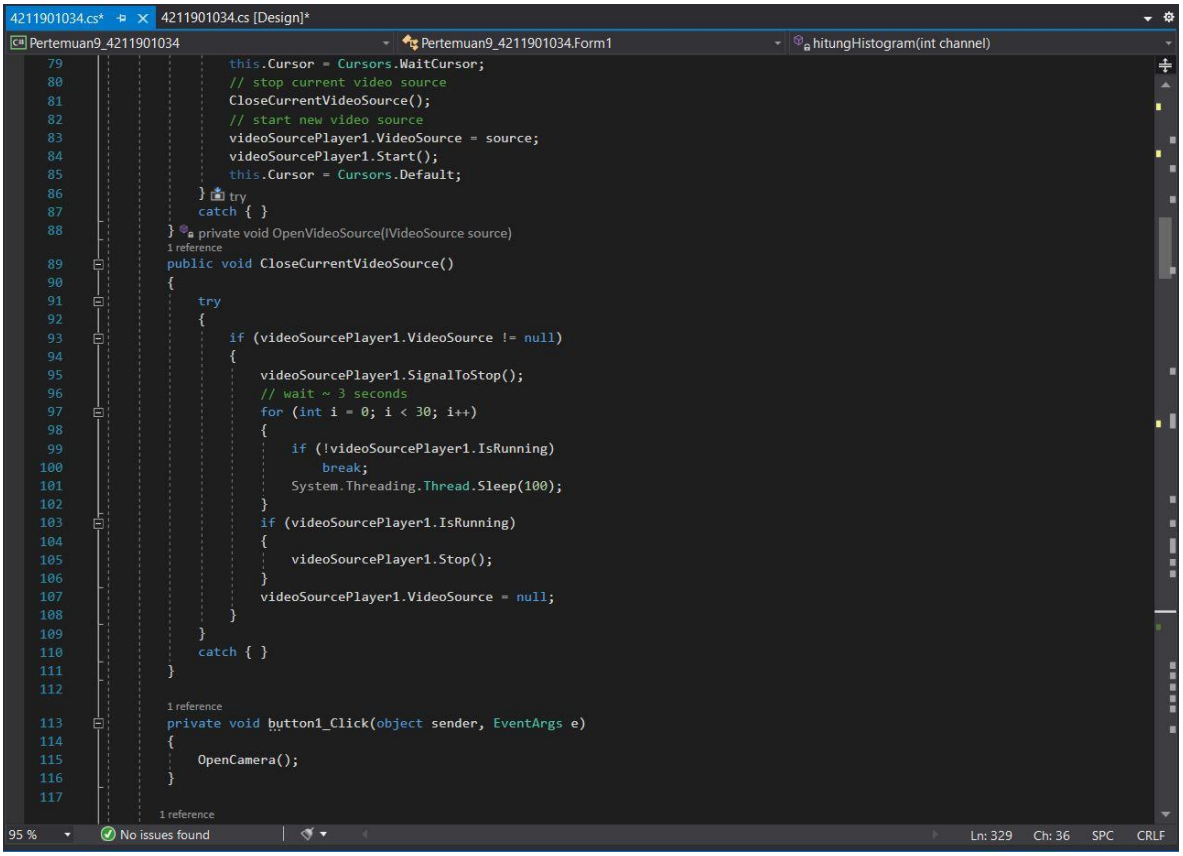
Source Code



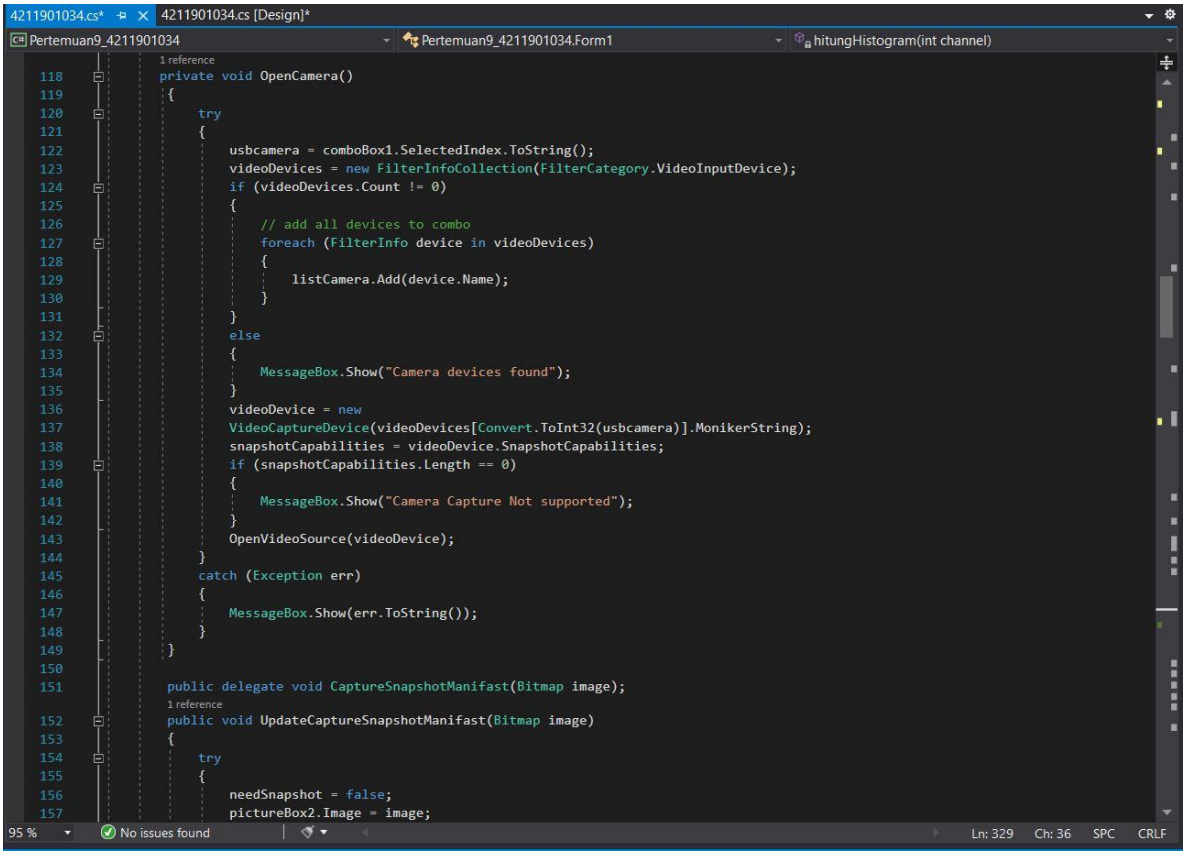
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Threading;
using System.IO;
using System.IO.Ports;
using System.Collections;
using System.Drawing.Imaging;
using AForge;
using AForge.Imaging;
using AForge.Imaging.Filters;
using AForge.Video;
using AForge.Video.DirectShow;
using AForge.Math.Geometry;

namespace Pertemuan9_4211901034
{
    3 references
    public partial class Form1 : Form
    {
        // global
        private FilterInfoCollection videoDevices;
        private VideoCaptureDevice videoDevice;
        private VideoCapabilities[] snapshotCapabilities;
        private ArrayList listCamera = new ArrayList();
        public string pathFolder = Application.StartupPath + @"\ImageCapture\";
        //for capturing image
        bool needSnapshot = false;
        int imageChannel = 0;
        //image variabel
        Bitmap sourceImage = null;
        Bitmap processedImage = null;
        Bitmap grayImage = null;
        Bitmap invertImage = null;

        1 reference
```



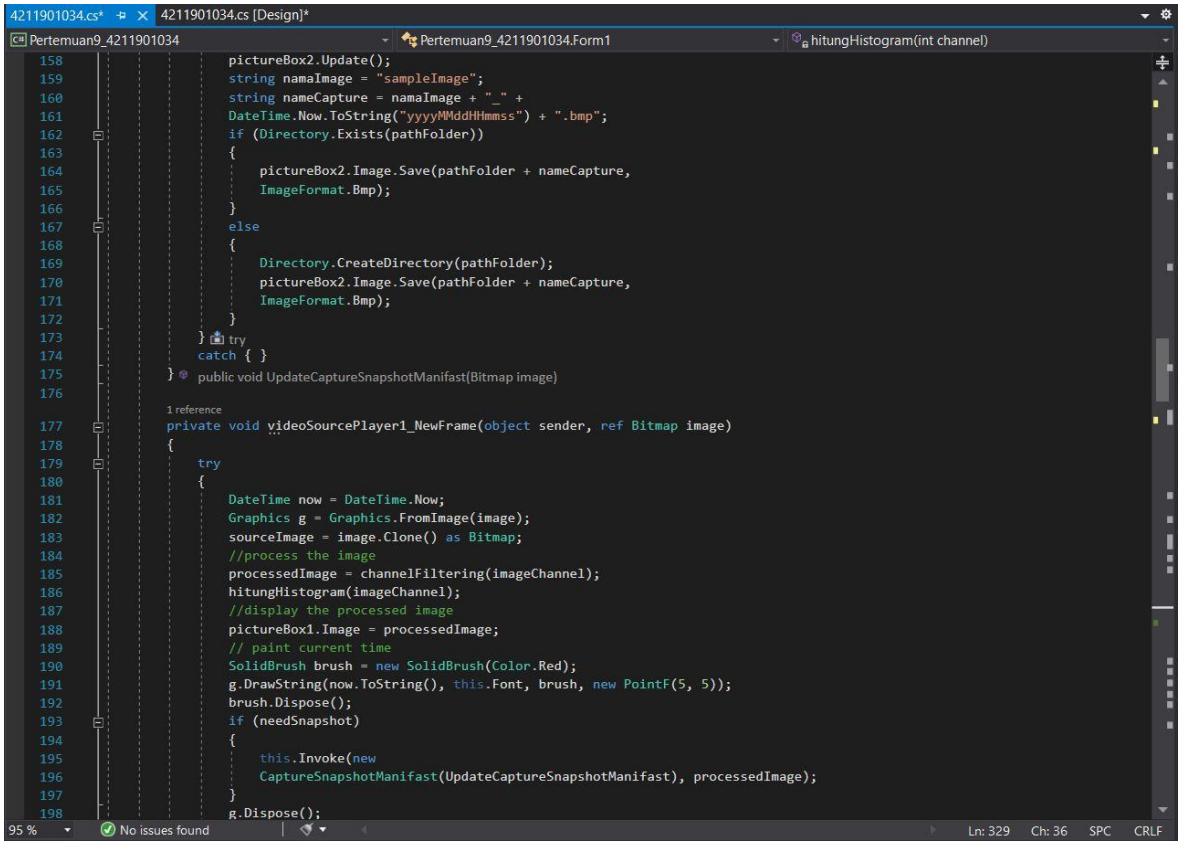
```
        1 reference
        public Form1()
        {
            InitializeComponent();
            //list the available camera and add to comboBox
            getListCameraUSB();
        }

        1 reference
        private void getListCameraUSB()
        {
            videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
            if (videoDevices.Count != 0)
            {
                // add all devices to combo
                foreach (FilterInfo device in videoDevices)
                {
                    comboBox1.Items.Add(device.Name);
                }
            }
            else
            {
                comboBox1.Items.Add("No DirectShow devices found");
            }
            comboBox1.SelectedIndex = 0;
        }

        // usb camera definition
        private static string _usbcamera;
        2 references
        public string usbcamera
        {
            get { return _usbcamera; }
            set { _usbcamera = value; }
        }
        // opening the video source
        1 reference
        private void OpenVideoSource(IVideoSource source)
        {
            try
            {
                // set busy cursor
```



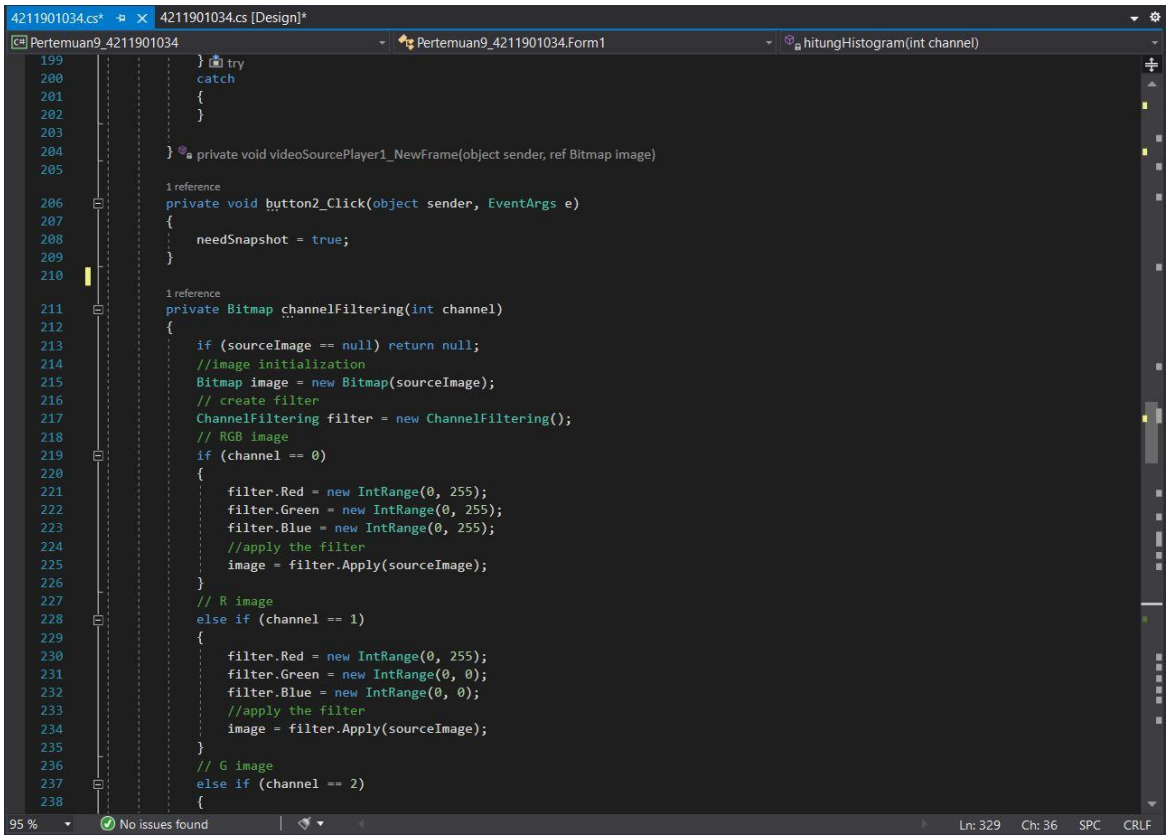
```
4211901034.cs*  4211901034.cs [Design]*
Pertemuan9_4211901034   Pertemuan9_4211901034.Form1   hitungHistogram(int channel)
79              this.Cursor = Cursors.WaitCursor;
80              // stop current video source
81              CloseCurrentVideoSource();
82              // start new video source
83              videoSourcePlayer1.VideoSource = source;
84              videoSourcePlayer1.Start();
85              this.Cursor = Cursors.Default;
86          } try
87          catch { }
88      } private void OpenVideoSource(IVideoSource source)
        1 reference
89      public void CloseCurrentVideoSource()
90      {
91          try
92          {
93              if (videoSourcePlayer1.VideoSource != null)
94              {
95                  videoSourcePlayer1.SignalToStop();
96                  // wait ~ 3 seconds
97                  for (int i = 0; i < 30; i++)
98                  {
99                      if (!videoSourcePlayer1.IsRunning)
100                         break;
101                     System.Threading.Thread.Sleep(100);
102                 }
103                 if (videoSourcePlayer1.IsRunning)
104                 {
105                     videoSourcePlayer1.Stop();
106                 }
107                 videoSourcePlayer1.VideoSource = null;
108             }
109         }
110         catch { }
111     }
112
        1 reference
113     private void button1_Click(object sender, EventArgs e)
114     {
115         OpenCamera();
116     }
117
        1 reference
95 %   No issues found   Ln: 329   Ch: 36   SPC   CRLF
```



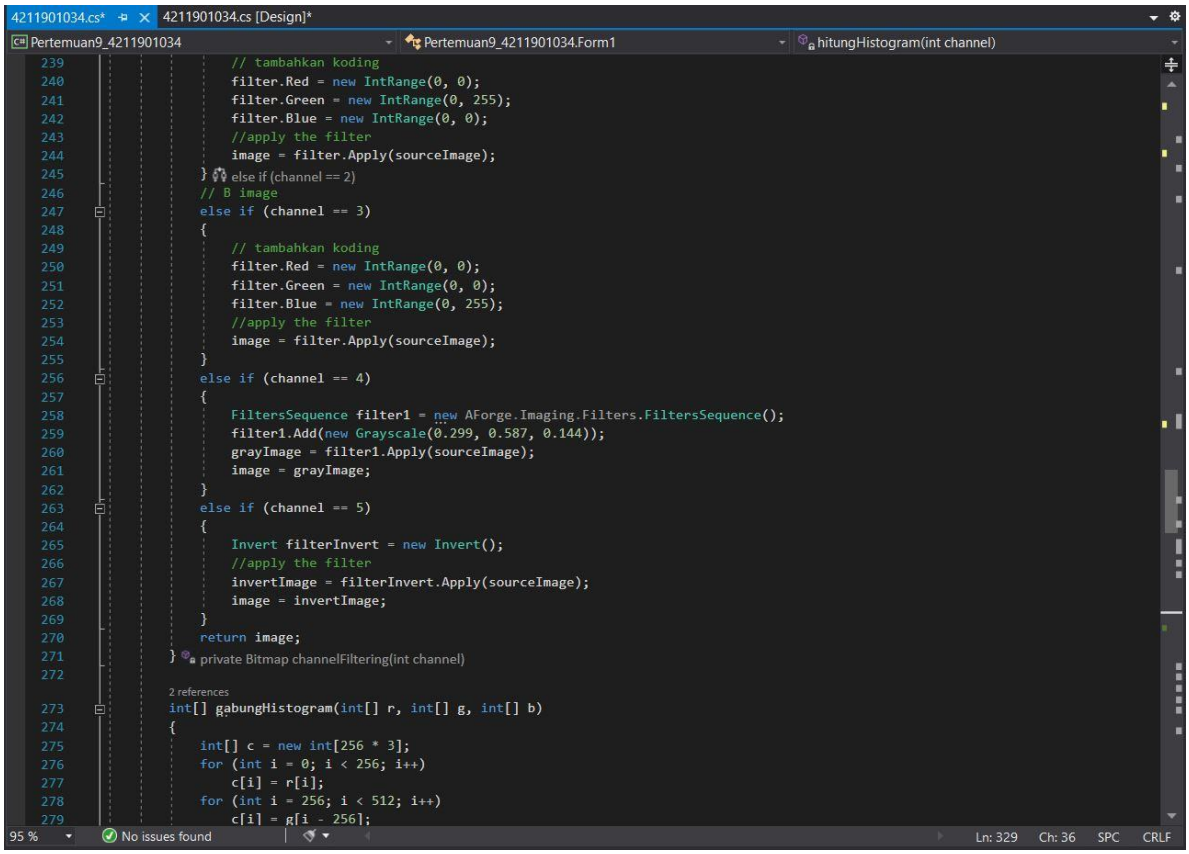
```
4211901034.cs*  4211901034.cs [Design]*
Pertemuan9_4211901034   Pertemuan9_4211901034.Form1   hitungHistogram(int channel)
        1 reference
118     private void OpenCamera()
119     {
120         try
121         {
122             usbcamera = comboBox1.SelectedIndex.ToString();
123             videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);
124             if (videoDevices.Count != 0)
125             {
126                 // add all devices to combo
127                 foreach (FilterInfo device in videoDevices)
128                 {
129                     listCamera.Add(device.Name);
130                 }
131             }
132             else
133             {
134                 MessageBox.Show("Camera devices found");
135             }
136             videoDevice = new
137             VideoCaptureDevice(videoDevices[Convert.ToInt32(usbcamera)].MonikerString);
138             snapshotCapabilities = videoDevice.SnapshotCapabilities;
139             if (snapshotCapabilities.Length == 0)
140             {
141                 MessageBox.Show("Camera Capture Not supported");
142             }
143             OpenVideoSource(videoDevice);
144         }
145         catch (Exception err)
146         {
147             MessageBox.Show(err.ToString());
148         }
149     }
150
151     public delegate void CaptureSnapshotManifast(Bitmap image);
        1 reference
152     public void UpdateCaptureSnapshotManifast(Bitmap image)
153     {
154         try
155         {
156             needSnapshot = false;
157             pictureBox2.Image = image;
95 %   No issues found   Ln: 329   Ch: 36   SPC   CRLF
```



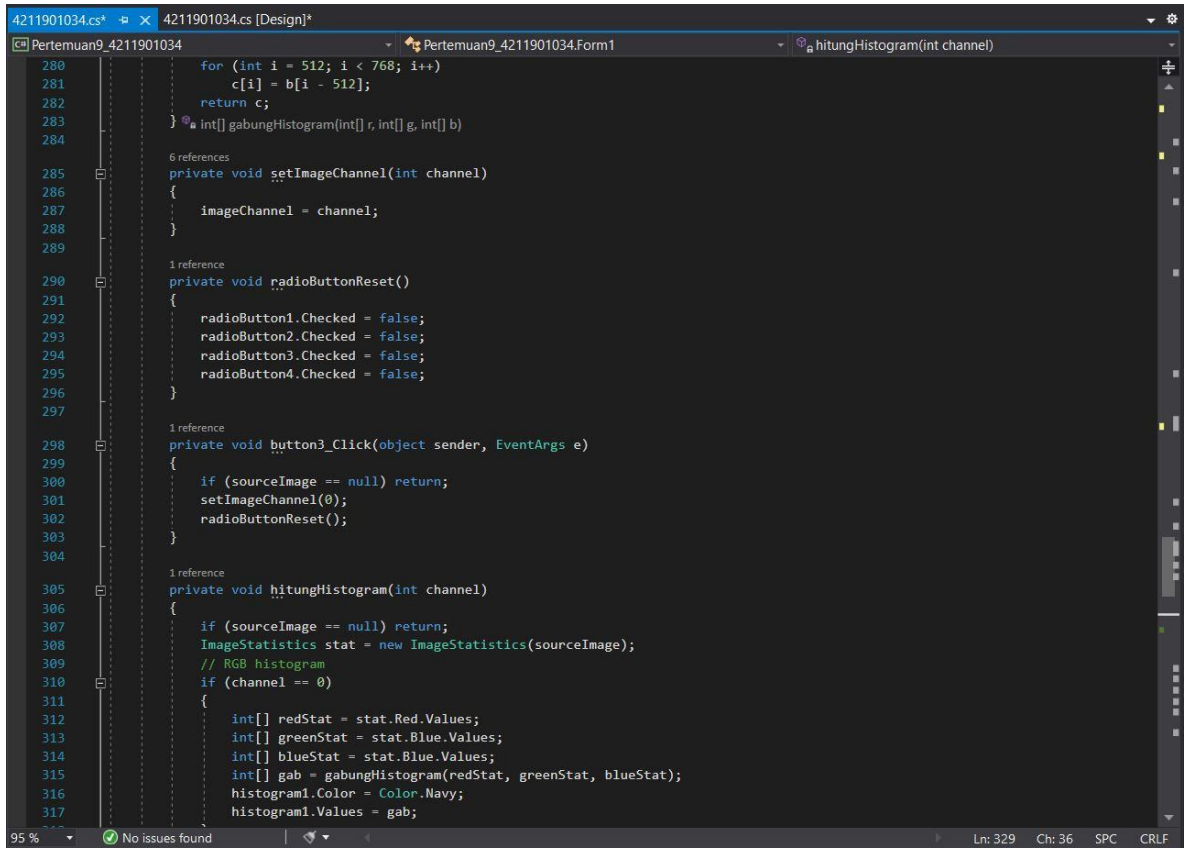
```
4211901034.cs*  4211901034.cs [Design]*
Pertemuan9_4211901034  Pertemuan9_4211901034.Form1  hitungHistogram(int channel)
158                     pictureBox2.Update();
159                     string namaImage = "sampleImage";
160                     string nameCapture = namaImage + "_" +
161                     DateTime.Now.ToString("yyyyMMddHHmmss") + ".bmp";
162                     if (Directory.Exists(pathFolder))
163                     {
164                         pictureBox2.Image.Save(pathFolder + nameCapture,
165                         ImageFormat.Bmp);
166                     }
167                     else
168                     {
169                         Directory.CreateDirectory(pathFolder);
170                         pictureBox2.Image.Save(pathFolder + nameCapture,
171                         ImageFormat.Bmp);
172                     }
173                 } try
174                 catch { }
175             } public void UpdateCaptureSnapshotManifast(Bitmap image)
176
                 1 reference
177             private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap image)
178             {
179                 try
180                 {
181                     DateTime now = DateTime.Now;
182                     Graphics g = Graphics.FromImage(image);
183                     sourceImage = image.Clone() as Bitmap;
184                     //process the image
185                     processedImage = channelFiltering(imageChannel);
186                     hitungHistogram(imageChannel);
187                     //display the processed image
188                     pictureBox1.Image = processedImage;
189                     // paint current time
190                     SolidBrush brush = new SolidBrush(Color.Red);
191                     g.DrawString(now.ToString(), this.Font, brush, new PointF(5, 5));
192                     brush.Dispose();
193                     if (needSnapshot)
194                     {
195                         this.Invoke(new
196                         CaptureSnapshotManifast(UpdateCaptureSnapshotManifast), processedImage);
197                     }
198                     g.Dispose();
95 %  No issues found  Ln: 329  Ch: 36  SPC  CRLF
```



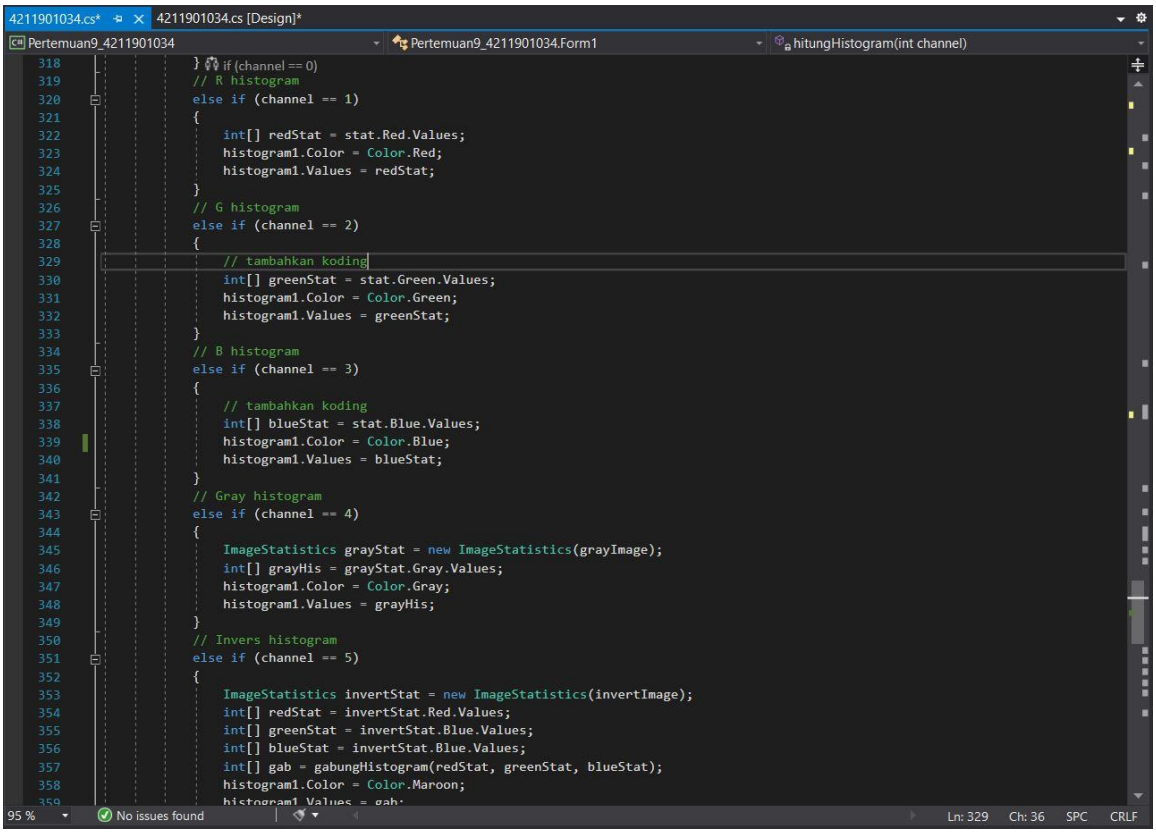
```
4211901034.cs*  4211901034.cs [Design]*
Pertemuan9_4211901034  Pertemuan9_4211901034.Form1  hitungHistogram(int channel)
199                 } try
200                 catch
201                 {
202                 }
203
204             } private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap image)
205
                 1 reference
206             private void button2_Click(object sender, EventArgs e)
207             {
208                 needSnapshot = true;
209             }
210
                 1 reference
211             private Bitmap channelFiltering(int channel)
212             {
213                 if (sourceImage == null) return null;
214                 //image initialization
215                 Bitmap image = new Bitmap(sourceImage);
216                 // create filter
217                 ChannelFiltering filter = new ChannelFiltering();
218                 // RGB image
219                 if (channel == 0)
220                 {
221                     filter.Red = new IntRange(0, 255);
222                     filter.Green = new IntRange(0, 255);
223                     filter.Blue = new IntRange(0, 255);
224                     //apply the filter
225                     image = filter.Apply(sourceImage);
226                 }
227                 // R image
228                 else if (channel == 1)
229                 {
230                     filter.Red = new IntRange(0, 255);
231                     filter.Green = new IntRange(0, 0);
232                     filter.Blue = new IntRange(0, 0);
233                     //apply the filter
234                     image = filter.Apply(sourceImage);
235                 }
236                 // G image
237                 else if (channel == 2)
238                 {
95 %  No issues found  Ln: 329  Ch: 36  SPC  CRLF
```

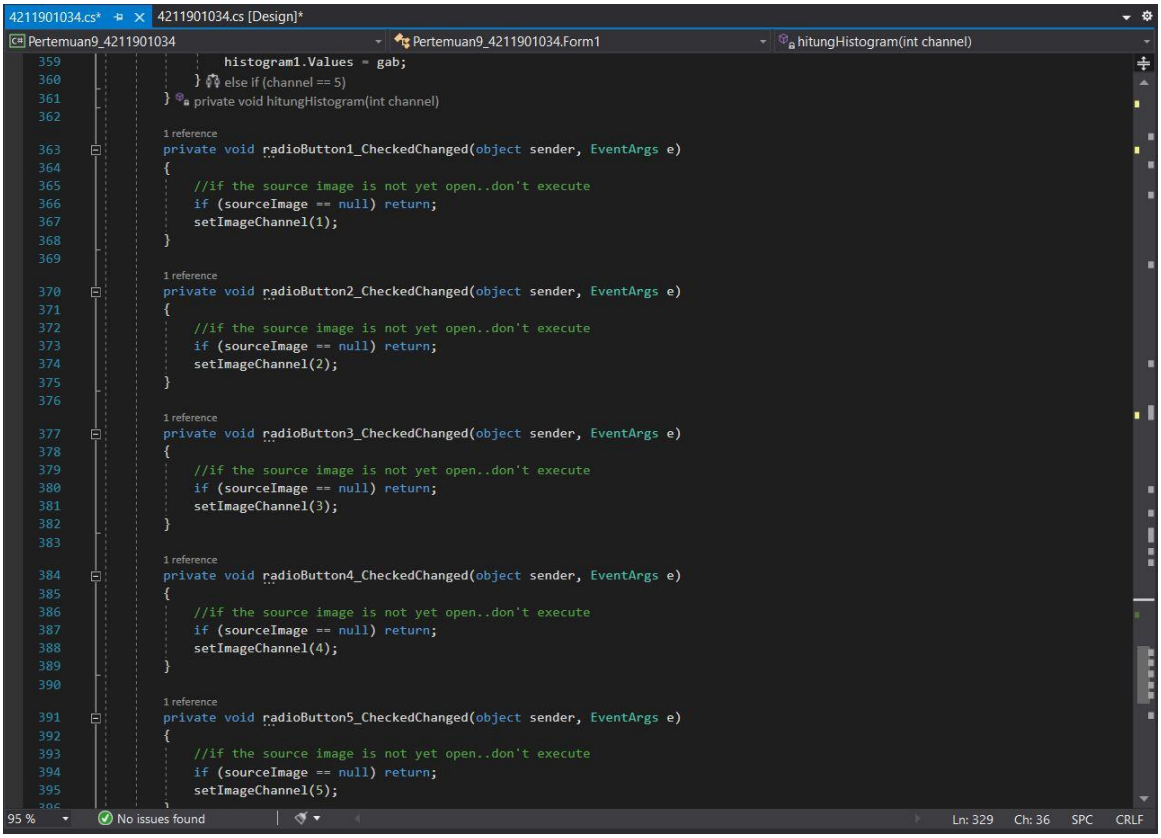


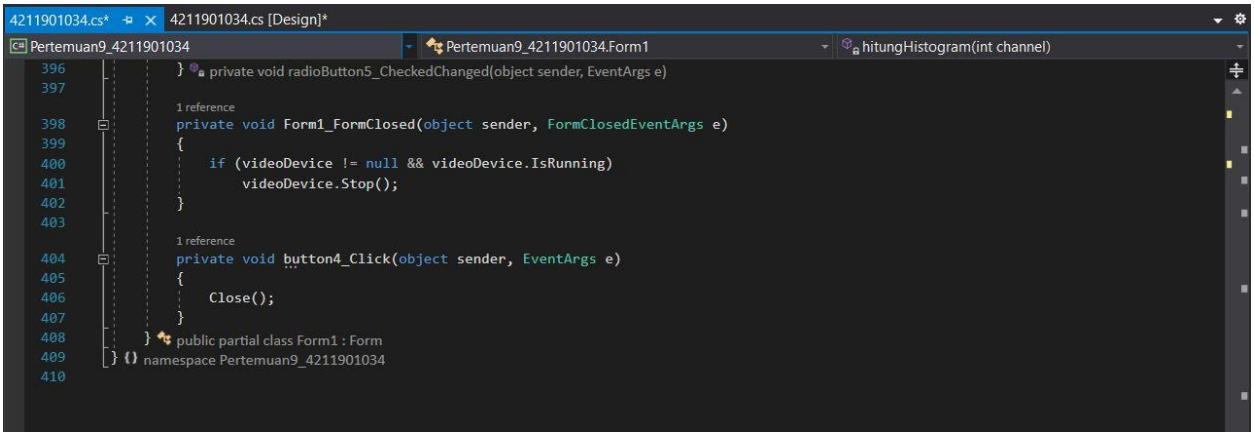
```
4211901034.cs*  4211901034.cs [Design]*
Pertemuan9_4211901034  Pertemuan9_4211901034.Form1  hitungHistogram(int channel)
239             // tambahkan koding
240             filter.Red = new IntRange(0, 0);
241             filter.Green = new IntRange(0, 255);
242             filter.Blue = new IntRange(0, 0);
243             //apply the filter
244             image = filter.Apply(sourceImage);
245         } else if (channel == 2)
246         // B image
247         else if (channel == 3)
248         {
249             // tambahkan koding
250             filter.Red = new IntRange(0, 0);
251             filter.Green = new IntRange(0, 0);
252             filter.Blue = new IntRange(0, 255);
253             //apply the filter
254             image = filter.Apply(sourceImage);
255         }
256         else if (channel == 4)
257         {
258             FiltersSequence filter1 = new AForge.Imaging.Filters.FiltersSequence();
259             filter1.Add(new Grayscale(0.299, 0.587, 0.144));
260             grayImage = filter1.Apply(sourceImage);
261             image = grayImage;
262         }
263         else if (channel == 5)
264         {
265             Invert filterInvert = new Invert();
266             //apply the filter
267             invertImage = filterInvert.Apply(sourceImage);
268             image = invertImage;
269         }
270         return image;
271     } private Bitmap channelFiltering(int channel)
272
            2 references
273     int[] gabungHistogram(int[] r, int[] g, int[] b)
274     {
275         int[] c = new int[256 * 3];
276         for (int i = 0; i < 256; i++)
277             c[i] = r[i];
278         for (int i = 256; i < 512; i++)
279             c[i] = g[i - 256];
95 %  No issues found  Ln: 329  Ch: 36  SPC  CRLF
```



```
4211901034.cs*  4211901034.cs [Design]*
Pertemuan9_4211901034  Pertemuan9_4211901034.Form1  hitungHistogram(int channel)
280             for (int i = 512; i < 768; i++)
281                 c[i] = b[i - 512];
282             return c;
283         } int[] gabungHistogram(int[] r, int[] g, int[] b)
284
            6 references
285         private void setImageChannel(int channel)
286         {
287             imageChannel = channel;
288         }
289
            1 reference
290         private void radioButtonReset()
291         {
292             radioButton1.Checked = false;
293             radioButton2.Checked = false;
294             radioButton3.Checked = false;
295             radioButton4.Checked = false;
296         }
297
            1 reference
298         private void button3_Click(object sender, EventArgs e)
299         {
300             if (sourceImage == null) return;
301             setImageChannel(0);
302             radioButtonReset();
303         }
304
            1 reference
305         private void hitungHistogram(int channel)
306         {
307             if (sourceImage == null) return;
308             ImageStatistics stat = new ImageStatistics(sourceImage);
309             // RGB histogram
310             if (channel == 0)
311             {
312                 int[] redStat = stat.Red.Values;
313                 int[] greenStat = stat.Blue.Values;
314                 int[] blueStat = stat.Blue.Values;
315                 int[] gab = gabungHistogram(redStat, greenStat, blueStat);
316                 histogram1.Color = Color.Navy;
317                 histogram1.Values = gab;
95 %  No issues found  Ln: 329  Ch: 36  SPC  CRLF
```


4211901034.cs*
4211901034.cs [Design]*
Pertemuan9_4211901034
Pertemuan9_4211901034.Form1
hitungHistogram(int channel)
} if (channel == 0)
// R histogram
else if (channel == 1)
{
int[] redStat = stat.Red.Values;
histogram1.Color = Color.Red;
histogram1.Values = redStat;
}
// G histogram
else if (channel == 2)
{
// tambahkan koding
int[] greenStat = stat.Green.Values;
histogram1.Color = Color.Green;
histogram1.Values = greenStat;
}
// B histogram
else if (channel == 3)
{
// tambahkan koding
int[] blueStat = stat.Blue.Values;
histogram1.Color = Color.Blue;
histogram1.Values = blueStat;
}
// Gray histogram
else if (channel == 4)
{
ImageStatistics grayStat = new ImageStatistics(grayImage);
int[] grayHis = grayStat.Gray.Values;
histogram1.Color = Color.Gray;
histogram1.Values = grayHis;
}
// Invers histogram
else if (channel == 5)
{
ImageStatistics invertStat = new ImageStatistics(invertImage);
int[] redStat = invertStat.Red.Values;
int[] greenStat = invertStat.Blue.Values;
int[] blueStat = invertStat.Blue.Values;
int[] gab = gabungHistogram(redStat, greenStat, blueStat);
histogram1.Color = Color.Maroon;
95 %
No issues found
Ln: 329
Ch: 36
SPC
CRLF


4211901034.cs*
4211901034.cs [Design]*
Pertemuan9_4211901034
Pertemuan9_4211901034.Form1
hitungHistogram(int channel)
histogram1.Values = gab;
} else if (channel == 5)
} private void hitungHistogram(int channel)
1 reference
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
//if the source image is not yet open..don't execute
if (sourceImage == null) return;
setImageChannel(1);
}
1 reference
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
//if the source image is not yet open..don't execute
if (sourceImage == null) return;
setImageChannel(2);
}
1 reference
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
//if the source image is not yet open..don't execute
if (sourceImage == null) return;
setImageChannel(3);
}
1 reference
private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
//if the source image is not yet open..don't execute
if (sourceImage == null) return;
setImageChannel(4);
}
1 reference
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
//if the source image is not yet open..don't execute
if (sourceImage == null) return;
setImageChannel(5);
95 %
No issues found
Ln: 329
Ch: 36
SPC
CRLF


4211901034.cs*
4211901034.cs [Design]*
Pertemuan9_4211901034
Pertemuan9_4211901034.Form1
hitungHistogram(int channel)
} private void radioButton5_CheckedChanged(object sender, EventArgs e)
1 reference
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
if (videoDevice != null && videoDevice.IsRunning)
videoDevice.Stop();
}
1 reference
private void button4_Click(object sender, EventArgs e)
{
Close();
}
} public partial class Form1 : Form
} {} namespace Pertemuan9_4211901034