

9

INTRODUCTION TO CAMERA CAPTURE BASED ON AFORGE

Pada pokok bahasan ini, mahasiswa akan mengaplikasikan penggunaan camera untuk mencapture objek.

Pokok Bahasan :

1. Menampilkan objek
2. Menampilkan chanel RGB dengan kamera
3. Menampilkan histogram

Latihan :

1. Membuat aplikasi menggunakan kamera
2. Membuat aplikasi menampilkan RGB dan histogramnya dari kamera
3. Membuat aplikasi menampilkan tiap channel RGB dan histogram dari kamera

9.1 AForge.Video dan AForge.Video.DirectShow

Capaian pembelajaran: memahami dan mengaplikasikan library AForge.Video dan AForge.Video.DirectShow untuk mengcapture video dan menampilkannya.

Library AForge.Video dan AForge.Video.DirectShow adalah library yang diberikan oleh AForge untuk mengakses dan menampilkan video dengan menggunakan kamera.

Beberapa kelas yang diberikan diantaranya

1. **FilterInfoCollection** merupakan kelas yang berisi koleksi informasi dari filter untuk mencari informasi tentang kamera
2. **VideoCaptureDevice** merupakan kelas yang menyatakan sumber video dari mana video itu berasal misal USB kamera
3. **VideoCapabilities** merupakan kelas yang menunjukkan kemampuan video capture (USB kamera dalam menentukan property dari video seperti frame size, frame rate dll.

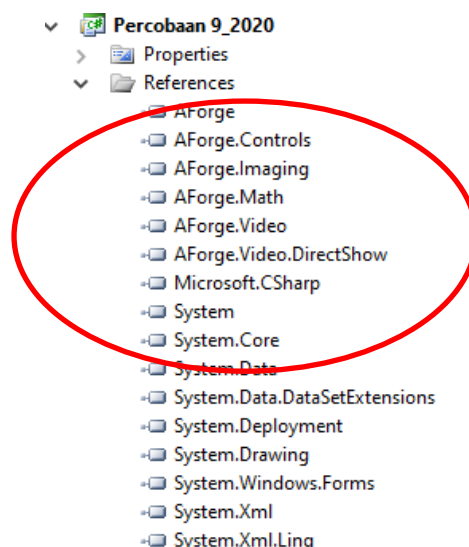
9.2 Latihan

Tujuan

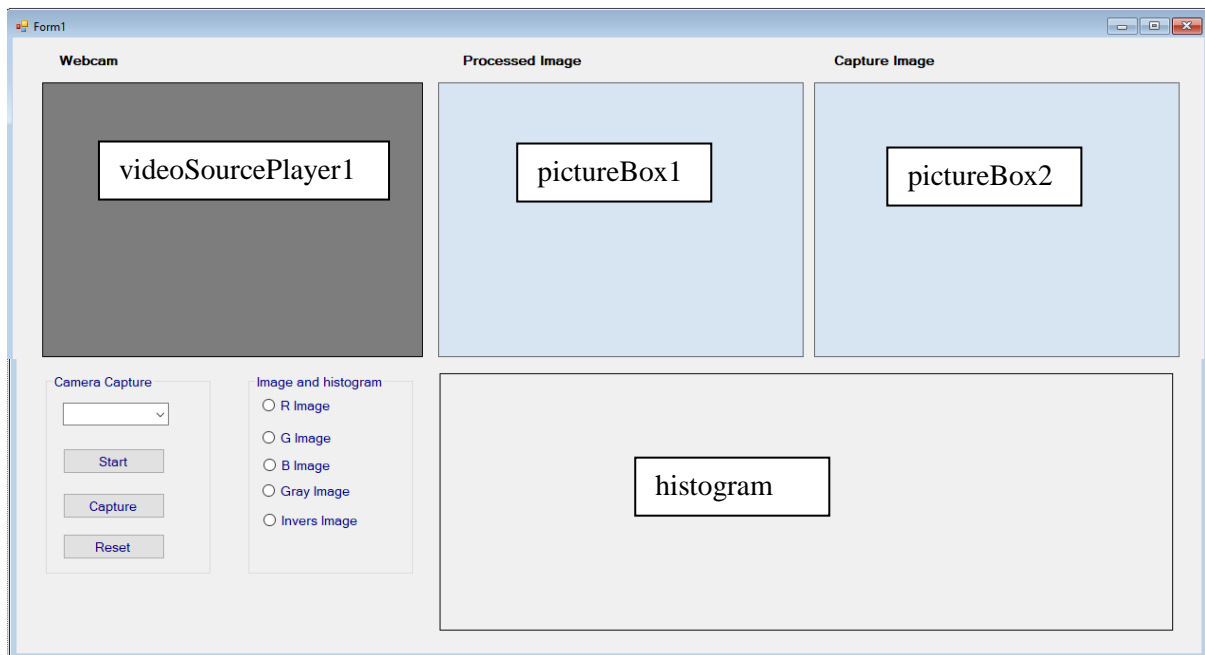
Memahami penggunaan AForge.NET dalam mengcapture image menggunakan kamera.

Prosedur

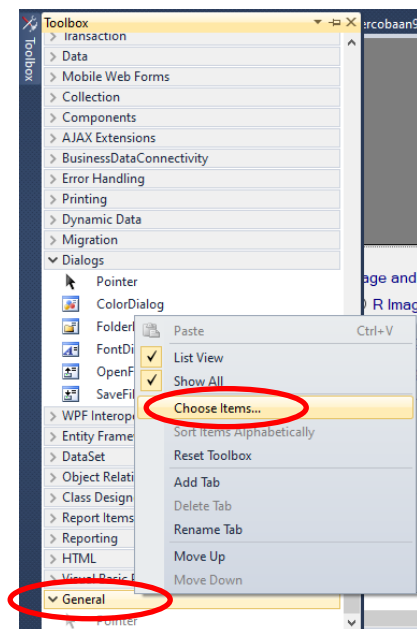
1. Tambahkah projek baru
2. Tambahkan library **AForge.dll**, **AForge.Controls.dll**, **AForge.Imaging.dll**, **AForge.Math.dll**, **AForge.Video.dll** dan **AForge.Video.DirectShow.dll** pada menu **References** seperti berikut :



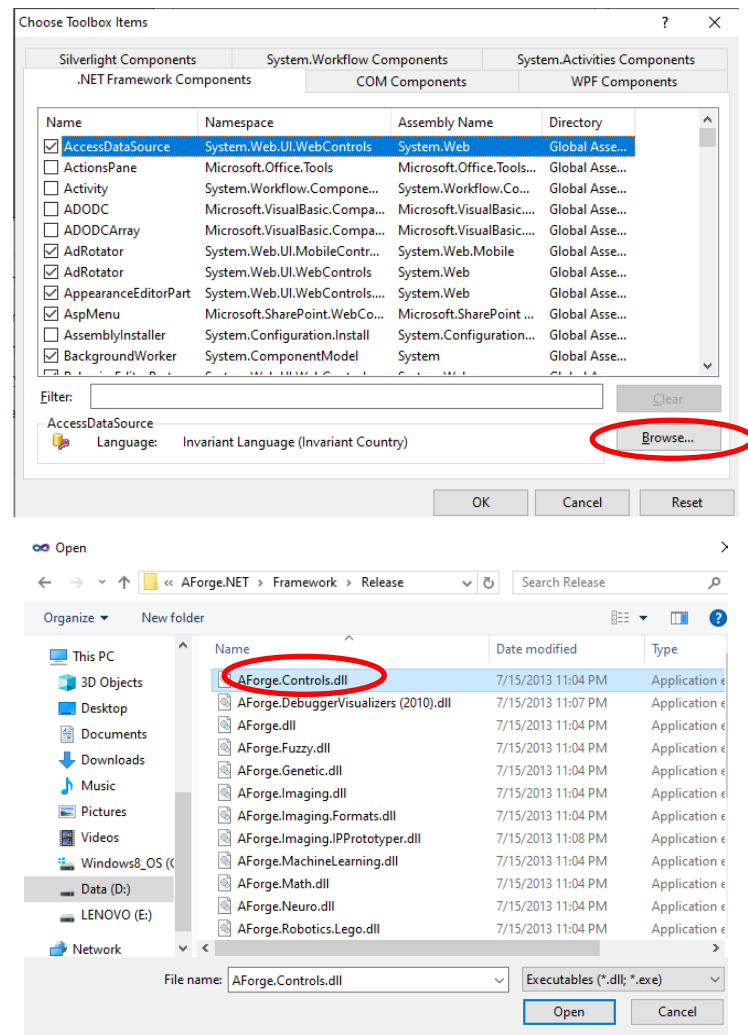
3. Tambahkan beberapa dengan **Toolbox** pada form anda sehingga menjadi seperti gambar berikut:



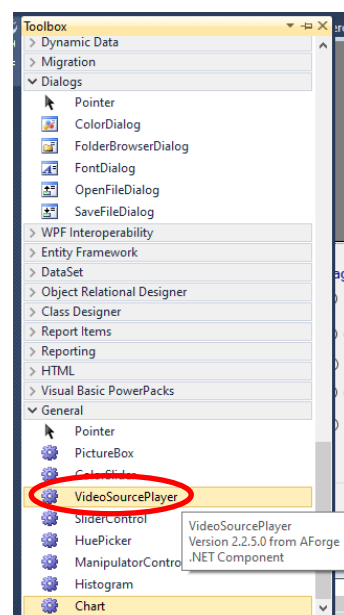
4. Untuk menambahkan **videoSourcePlayer**, klik **Toolbox** → pilih tab **General** → klik kanan → pilih **Choose Items** seperti gambar berikut :



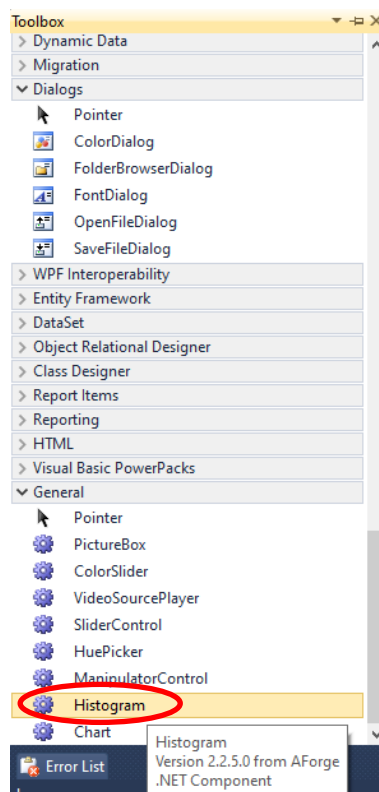
5. Kemudian klik Browse dan cari folder dimana **AForge** disimpan, dan pilih **AForge.Control.dll** dan klik **OK**



6. Pilihlah **VideoSourcePlayer** dari tab **General** pada **Toolbox**



7. Tambahkan histogram dengan cara yang sama :



8. Selanjutnya tambahkan **button**, **pictureBox**, **radioButton** dan lainnya seperti biasa.

9. Tambahkan library berikut pada program anda :

```
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;
```

10. Tambahkan global variable sebagai berikut :

```
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;
```

11. Tambahkan fungsi `getListCameraUSB()`; di bawah `InitializeComponent()`;

```
public Percobaan9()
{
    InitializeComponent();

    //list the available camera and add to comboBox
    getListCameraUSB();
}
```

12. Tambahkan juga fungsi berikut ini :

```
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;
}
```

13. Tambahkan fungsi-fungsi berikut untuk membuka dan menutup video source:

```
// usb camera definition
private static string _usbcamera;
public string usbcamera
{
    get { return _usbcamera; }
    set { _usbcamera = value; }
}

// opening the video source
private void OpenVideoSource(IVideoSource source)
{
    try
    {
        // set busy cursor
        this.Cursor = Cursors.WaitCursor;

        // stop current video source
        CloseCurrentVideoSource();

        // start new video source
        videoSourcePlayer1.VideoSource = source;
        videoSourcePlayer1.Start();

        this.Cursor = Cursors.Default;
    }
    catch { }
}
```

```

// closing the video source
public void CloseCurrentVideoSource()
{
    try
    {
        if (videoSourcePlayer1.VideoSource != null)
        {
            videoSourcePlayer1.SignalToStop();

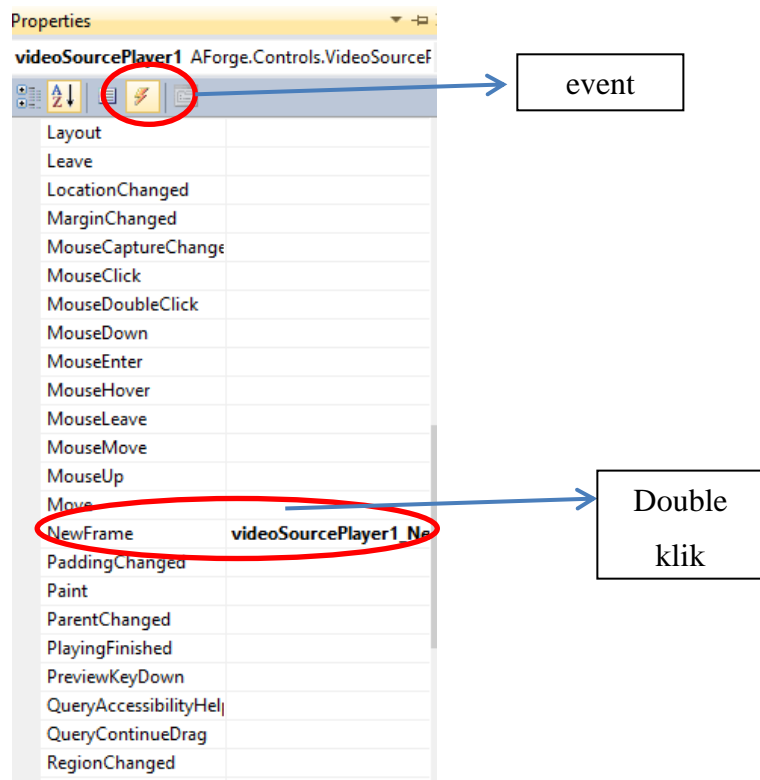
            // wait ~ 3 seconds
            for (int i = 0; i < 30; i++)
            {
                if (!videoSourcePlayer1.IsRunning)
                    break;
                System.Threading.Thread.Sleep(100);
            }

            if (videoSourcePlayer1.IsRunning)
            {
                videoSourcePlayer1.Stop();
            }

            videoSourcePlayer1.VideoSource = null;
        }
    }
    catch { }
}

```

14. Untuk mengcapture frame menggunakan kamera, tambahkan program berikut dengan **meng-klik kanan** videoSourcePlayer1 → properties → event. Double klik **NewFrame**. Dan tambahkan program berikut :



```

private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap image)
{
    try
    {
        DateTime now = DateTime.Now;
        Graphics g = Graphics.FromImage(image);

        sourceImage = image.Clone() as Bitmap;

        //process the image
        processedImage= channelFiltering(imageChannel);
        hitungHistogram(imageChannel);

        //display the processed image
        pictureBox1.Image = processedImage;
        // paint current time
        SolidBrush brush = new SolidBrush(Color.Red);
        g.DrawString(now.ToString(), this.Font, brush, new PointF(5, 5));
        brush.Dispose();
        if (needSnapshot)
        {
            this.Invoke(new
CaptureSnapshotManifest(UpdateCaptureSnapshotManifest), processedImage);
        }
        g.Dispose();
    }
    catch
    {
    }
}

```

15. Tambahkan fungsi berikut untuk open camera:

```

private void OpenCamera()
{
    try
    {
        usbcamera = comboBox1.SelectedIndex.ToString();
        videoDevices = new FilterInfoCollection(FilterCategory.VideoInputDevice);

        if (videoDevices.Count != 0)
        {
            // add all devices to combo
            foreach (FilterInfo device in videoDevices)
            {
                listCamera.Add(device.Name);
            }
        }
        else
        {
            MessageBox.Show("Camera devices found");
        }

        videoDevice = new
VideoCaptureDevice(videoDevices[Convert.ToInt32(usbcamera)].MonikerString);
        snapshotCapabilities = videoDevice.SnapshotCapabilities;
        if (snapshotCapabilities.Length == 0)
        {
            MessageBox.Show("Camera Capture Not supported");
        }
    }
}

```



```

        OpenVideoSource(videoDevice);
    }
    catch (Exception err)
    {
        MessageBox.Show(err.ToString());
    }
}

```

16. Double klik button Start dan tambahkan program berikut :

```

private void button1_Click(object sender, EventArgs e)
{
    OpenCamera();
}

```

17. Tambahkan fungsi-fungsi berikut untuk meng-capture video per frame

```

public delegate void CaptureSnapshotManifest(Bitmap image);

public void UpdateCaptureSnapshotManifest(Bitmap image)
{
    try
    {
        needSnapshot = false;
        pictureBox2.Image = image;
        pictureBox2.Update();

        string namaImage = "sampleImage";
        string nameCapture = namaImage + "_" +
        DateTime.Now.ToString("yyyyMMddHHmmss") + ".bmp";

        if (Directory.Exists(pathFolder))
        {
            pictureBox2.Image.Save(pathFolder + nameCapture,
            ImageFormat.Bmp);
        }
        else
        {
            Directory.CreateDirectory(pathFolder);
            pictureBox2.Image.Save(pathFolder + nameCapture,
            ImageFormat.Bmp);
        }

    }
    catch { }
}

```

18. Double klik tombol capture dan tambahkan fungsi berikut :

```

private void button2_Click(object sender, EventArgs e)
{
    needSnapshot = true;
}

```

19. Tambahkan fungsi untuk mengubah citra RGB ke masing-masing channel

```
private Bitmap channelFiltering(int channel)
{
    if (sourceImage == null) return null;

    //image initialization
    Bitmap image = new Bitmap(sourceImage);

    // create filter
    ChannelFiltering filter = new ChannelFiltering();

    // RGB image
    if (channel == 0)
    {
        filter.Red = new IntRange(0, 255);
        filter.Green = new IntRange(0, 255);
        filter.Blue = new IntRange(0, 255);

        //apply the filter
        image = filter.Apply(sourceImage);
    }
    // R image
    else if (channel == 1)
    {
        filter.Red = new IntRange(0, 255);
        filter.Green = new IntRange(0, 0);
        filter.Blue = new IntRange(0, 0);

        //apply the filter
        image = filter.Apply(sourceImage);
    }
    // G image
    else if (channel == 2)
    {
        // tambahkan koding
    }
    // B image
    else if (channel == 3)
    {
        // tambahkan koding
    }
    else if (channel == 4)
    {
        FiltersSequence filter1 = new AForge.Imaging.Filters.FiltersSequence();
        filter1.Add(new Grayscale(0.299, 0.587, 0.144));
        grayImage = filter1.Apply(sourceImage);
        image = grayImage;
    }
    else if (channel == 5)
    {
        Invert filterInvert = new Invert();

        //apply the filter
        invertImage = filterInvert.Apply(sourceImage);
        image = invertImage;
    }

    return image;
}
```

20. Tambahkan fungsi untuk menghitung histogram masing-masing channel

```
private void hitungHistogram(int channel)
{
    if (sourceImage == null) return;

    ImageStatistics stat = new ImageStatistics(sourceImage);

    // RGB histogram
    if (channel == 0)
    {
        int[] redStat = stat.Red.Values;
        int[] greenStat = stat.Blue.Values;
        int[] blueStat = stat.Blue.Values;
        int [] gab = gabungHistogram(redStat, greenStat, blueStat);
        histogram1.Color = Color.Navy;
        histogram1.Values = gab;
    }
    // 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
    }
    // B histogram
    else if (channel == 3)
    {
        // tambahkan koding
    }

    // 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;
        histogram1.Values = gab;
    }
}
```

21. Tambahkan fungsi untuk menggabungkan histogram masing-masing channel ke dalam satu histogram

```
int[] gabungHistogram(int[] r, int[] g, int[] b)
{
    int[] c = new int[256*3];
    for (int i = 0; i < 256; i++)
        c[i] = r[i];
    for (int i = 256; i < 512; i++)
        c[i] = g[i - 256];
    for (int i = 512; i < 768; i++)
        c[i] = b[i - 512];
    return c;
}
```

22. Tambahkan fungsi memilih channel ;

```
private void setImageChannel(int channel)
{
    imageChannel = channel;
}
```

23. Double klik semua radioButton dan tambahkan program berikut:

```
//R Image radioButton
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);
}

//G Image radioButton
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);
}

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);
}

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);
}

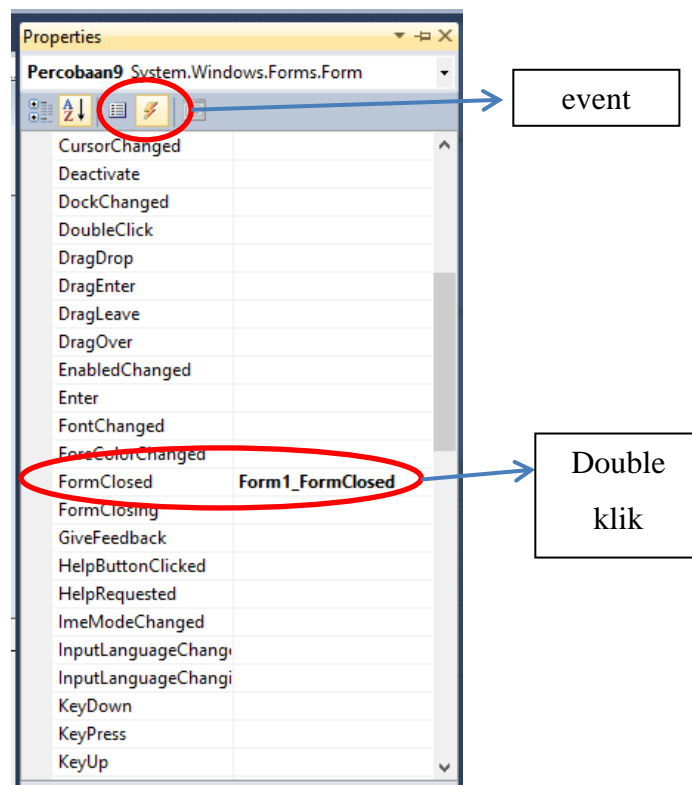
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
    //if the source image is not yet open..don't execute
    if (sourceImage == null) return;
    setImageChannel(5);
}
```

24. Double klik button Reset dan tambahkan :

```
private void button3_Click(object sender, EventArgs e)
{
    if (sourceImage == null) return;
    setImageChannel(0);
    radioButtonReset();
}

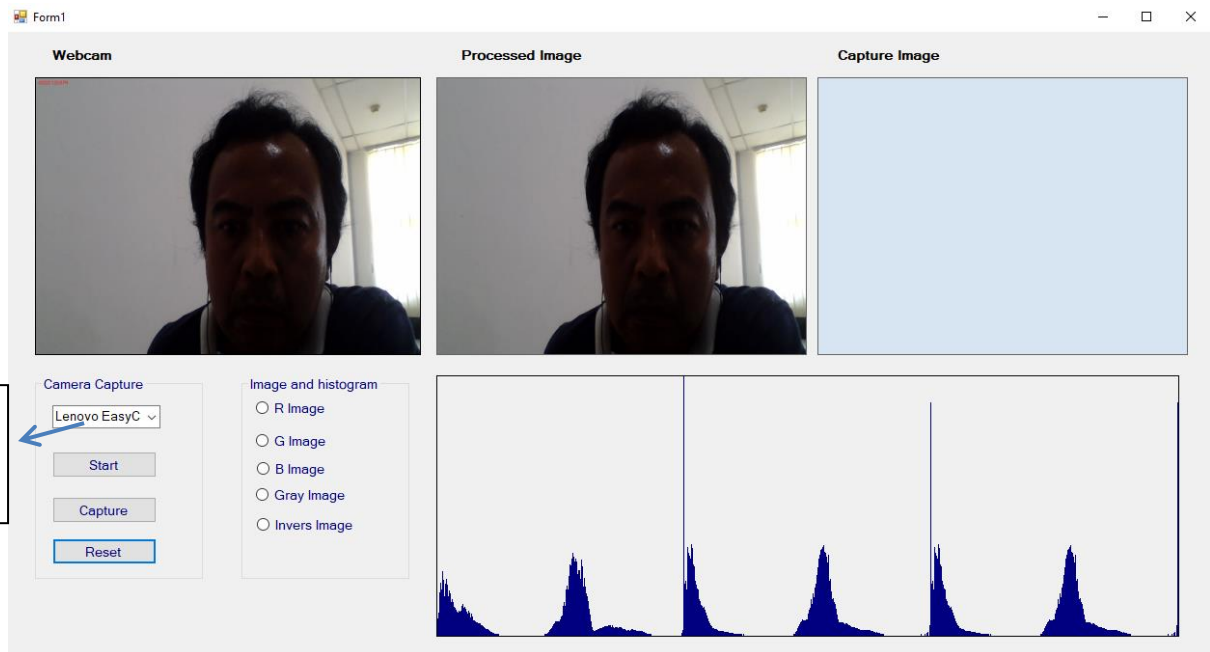
private void radioButtonReset()
{
    radioButton1.Checked = false;
    radioButton2.Checked = false;
    radioButton3.Checked = false;
    radioButton4.Checked = false;
}
```

25. Untuk menutup jendela form dan memberhentikan proses pembacaan video, tambahkan program berikut dengan **meng-klik kanan** form → properties → event. Double klik **FormClosed**. Tambahkan program berikut :

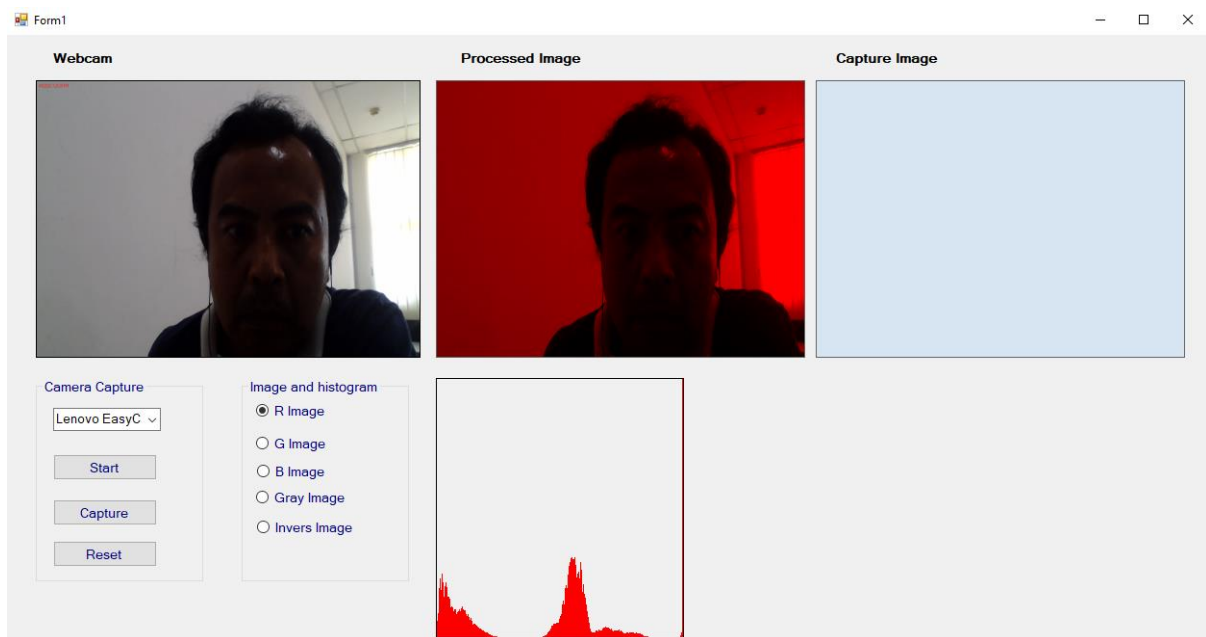


```
private void Form1_FormClosed(object sender, FormClosedEventArgs e)
{
    if (videoDevice != null && videoDevice.IsRunning)
        videoDevice.Stop();
}
```

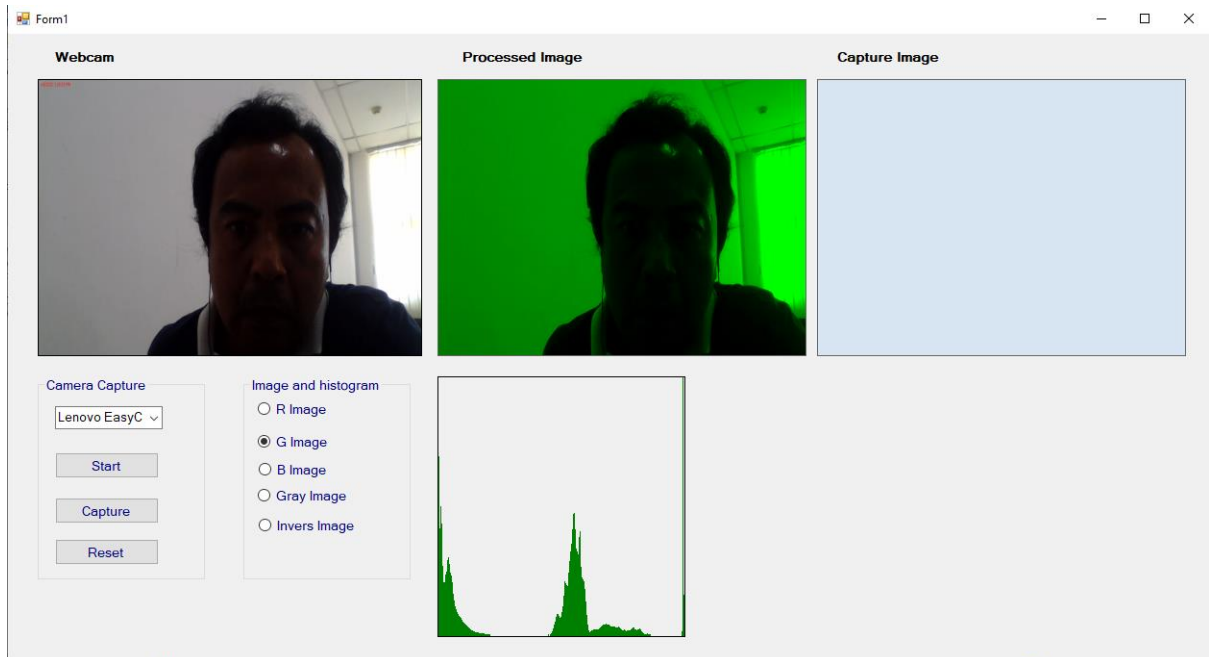
26. Jalankan program dan klik tombol **Start** : (histogram RGB mirip dikarenakan gambar kurang pencahayaan)



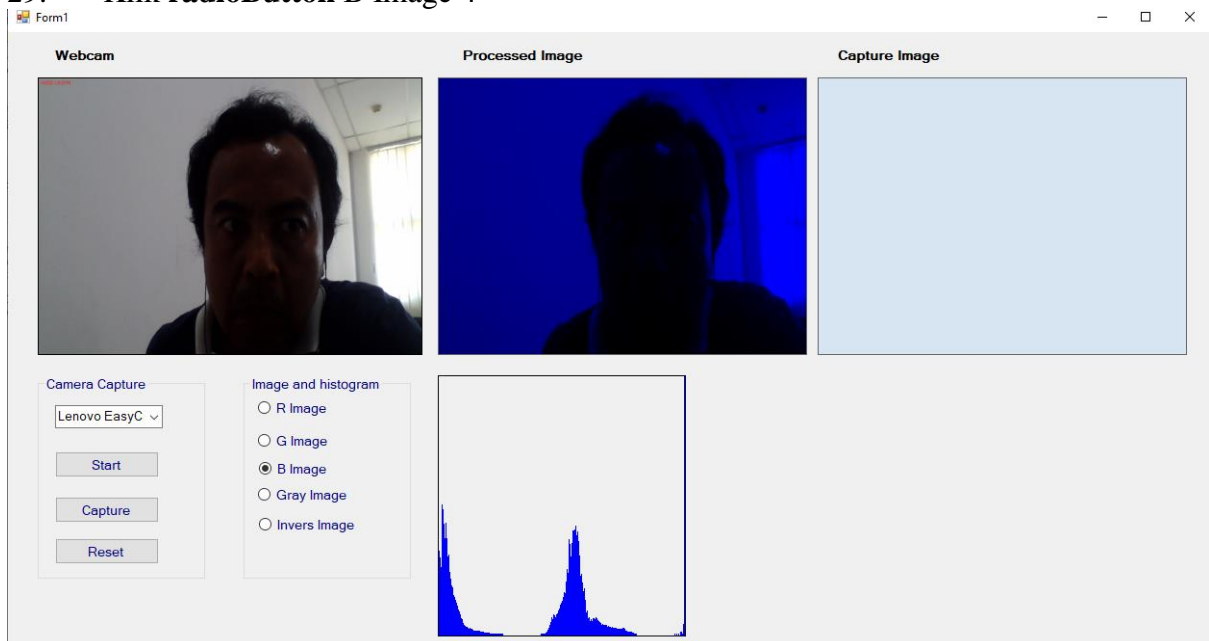
27. Klik **radioButton R Image** :



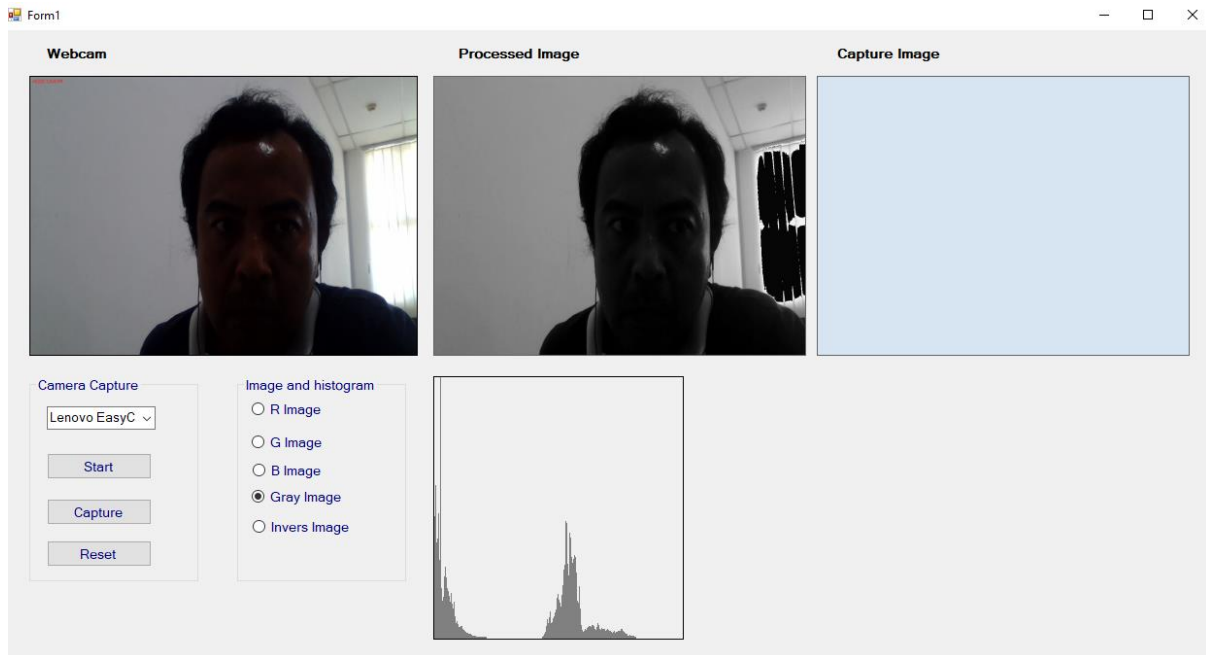
28. Klik **radioButton** G Image :



29. Klik **radioButton** B Image :

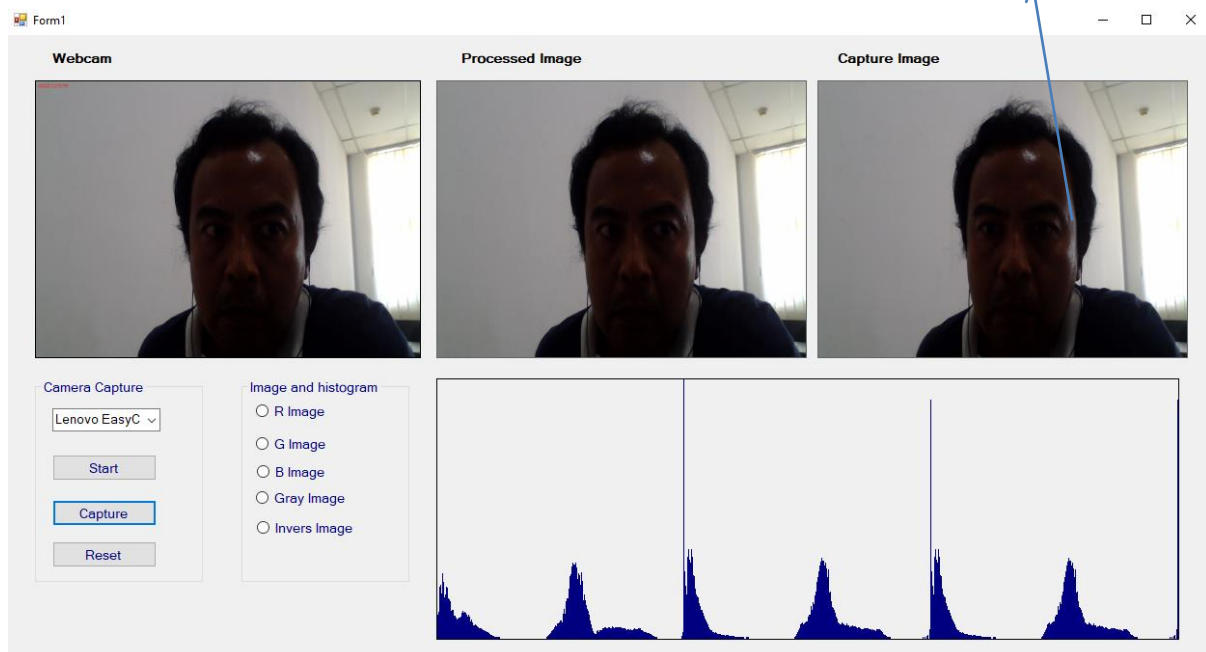


30. Klik **radioButton** Gray Image :



31. Klik **Button** capture :

Hasil Capture disimpan di : \\... \Percobaan
9_2020\bin\Debug\ImageCapture



9.3 Tugas dan Pertanyaan

1. Lengkapi koding untuk beberapa fungsi yang belum lengkap.