

# 13

## MOTION TRACKING

Pada pokok bahasan ini, mahasiswa akan mempelajari beberapa algoritma untuk mendeteksi objek yang bergerak secara real time menggunakan kamera.

### **Pokok Bahasan :**

1. Algoritma pendeteksi gerak (motion detection algorithm)
2. Algoritma pemrosesan object bergerak (motion detection processing)

### **Latihan:**

1. Membuat aplikasi pendeteksian objek bergerak secara *real time*

### 11.1. Motion Tracking

Capaian pembelajaran: memahami dan mengaplikasikan motion tracking algoritma untuk mentracking objek bergerak yang tercapture camera.

Dalam motion tracking sedikitnya ada dua algoritma yang akan dipelajari : algoritma pendeteksi objek dan algoritma pemrosesan objek bergerak. Algoritma pendeteksi objek bertujuan untuk mendeteksi objek yang bergerak. Algoritma pemrosesan objek bertujuan untuk memproses objek bergerak yang terdeteksi seperti melakukan highlight, membuat region dll.

### 11.2. Algoritma pendeteksi objek

Dalam library AForge ada 3 algoritma dalam pendeteksi objek

#### 1. Two frames difference

Algoritma ini mendeteksi objek bergerak dengan mengurangi frame yang berurutan.

Semakin besar perbedaannya maka semakin terdeteksi adanya objek yang bergerak

#### 2. Simple background modeling motion detector

Algoritma ini mendeteksi objek bergerak dengan mengurangi suatu frame dengan background-nya. Background di update setelah sekian frame jika tidak ada pergerakan.

#### 3. Custom frame difference motion detector

Algoritma ini mendeteksi objek bergerak dengan mengurangi suatu frame dengan background-didefinisikan sebelumnya. Setelah didapatkan objeknya, maka kemudian objek bergerak didefinisikan dengan mengurangi antar frame

### 11.3. Algoritma pemrosesan objek

Dalam library AForge ada 3 algoritma dalam pemrosesan objek

#### 1. Motion area highlighting

Algoritma ini bertujuan untuk meng-*highlight* motion area yang ditemukan oleh algoritma deteksi objek.

#### 2. Motion border highlighting

Algoritma ini bertujuan untuk meng-*highlight borders* dari objek yang bergerak yang ditemukan oleh algoritma deteksi objek.

### 3. Grid motion area processing

Algoritma ini bertujuan untuk melakukan *grid processing* dari objek yang bergerak. Yang berarti seluruh objek bergerak dibagi dalam beberapa grid.

### 4. Blob counting objects processing

Algoritma ini melakukan pemisahan objek bergerak berdasarkan jumlah pixel yang bergerak dalam suatu objek. Jika melewati *threshold*nya maka dianggap objek bergerak.



Motion area highlighting



Motion border highlighting



Grid motion area processing



Blob counting objects processing

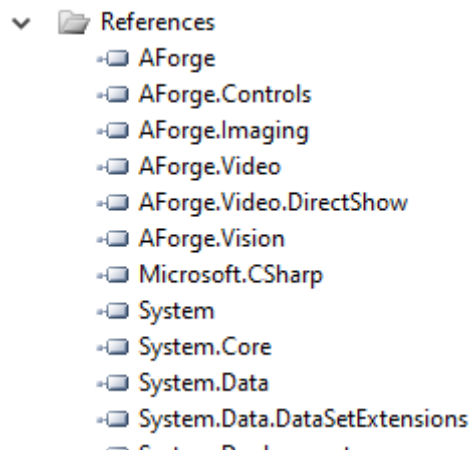
## 11.4. Latihan

### Tujuan

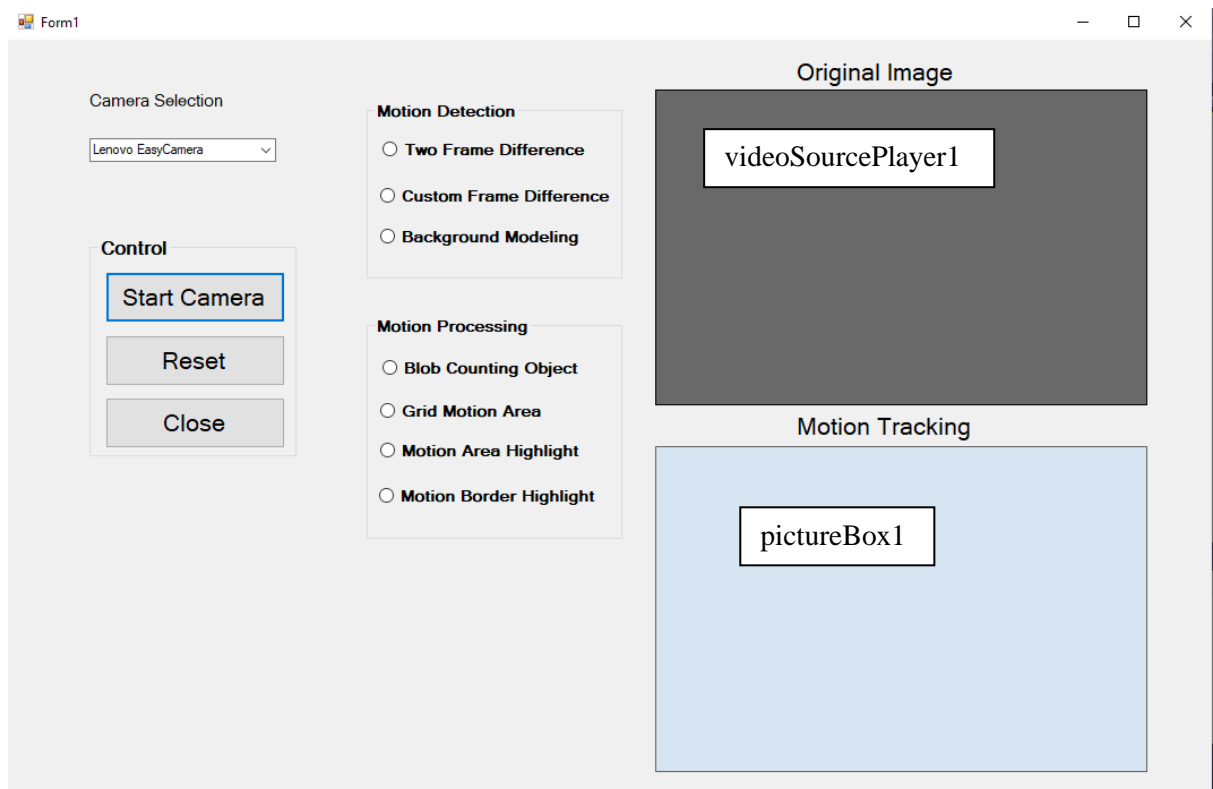
Memahami penggunaan AForge.NET dan mengaplikasikannya dalam mendeteksi objek bergerak.

### Prosedur

1. Silahkan tambahkan projek baru dan tambahkan library **AForge.dll**, **AForge.Control.dll**, **AForge.Imaging.dll**, **AForge.Math.dll**, **AForge.Video.dll**, **AForge.Video.DirectShow.dll** dan **AForge.Vision.dll** pada menu **References**



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



3. Tambahkan libray berikut :

```
using AForge;
using AForge.Imaging;
using AForge.Imaging.Filters;
using AForge.Video;
using AForge.Video.DirectShow;
using AForge.Vision.Motion;
using System.Collections;
```

4. Tambahkan global variable berikut :

```
private FilterInfoCollection videoDevices;
private VideoCaptureDevice videoDevice;
private VideoCapabilities[] snapshotCapabilities;
private ArrayList listCamera = new ArrayList();

//image variabel
Bitmap sourceImage = null;
Bitmap captureImage = null;

//initialization for motion detection
TwoFramesDifferenceDetector TFDdetector = new TwoFramesDifferenceDetector();
SimpleBackgroundModelingDetector SBMdetector = new SimpleBackgroundModelingDetector();
CustomFrameDifferenceDetector CFDdetector = new CustomFrameDifferenceDetector();

//initialization for motion processing
BlobCountingObjectsProcessing BCProcessing = new BlobCountingObjectsProcessing();
GridMotionAreaProcessing GMAprocessing = new GridMotionAreaProcessing();
MotionAreaHighlighting MAHprocessing = new MotionAreaHighlighting();
MotionBorderHighlighting MBHprocessing = new MotionBorderHighlighting();

//initialization for motion detector
MotionDetector motionDetector = null;

/* type of motion detection
   Two frame difference = 0
   simple background modelling = 1;
   Custom frame difference = 2
*/
int motionDetectionType = 5; //init of motion detection

/*type of motion processing
   blob counting object = 0
   grid motion area = 1
   motion area highlight =2
   motion border highlight = 3;
*/
int motionProcessingType = 5; //init of motion processing
```

5. Tambahkan fungsi radioButtonInit(); di bawah InitializeComponent();:

```
public Form1()
{
    InitializeComponent();

    //tambahkan ini
    radioButtonInit();
}
```

6. Tambahkan fungsi berikut untuk mendeteksi objek dan memproses hasil deteksi objek:

```
private void motionTracking(Bitmap srcImage)
{
    if (srcImage == null) return ;

    captureImage = (Bitmap)srcImage.Clone();

    //Threshold untuk blob counting
    //dimana hanya objek dengan ukuran minimal 300x300 yang akan diproses
    BCOProcessing.MinObjectsHeight = 300;
    BCOProcessing.MinObjectsWidth = 300;

    GMAprocessing.GridHeight = 50;
    GMAprocessing.GridWidth = 50;

    // initial untk highlight warna
    //blob counting
    BCOProcessing.HighlightColor = Color.Red;
    // grid motion area
    GMAprocessing.HighlightColor = Color.Green;
    // motion area
    MAHprocessing.HighlightColor = Color.Blue;
    //motion border
    MBHprocessing.HighlightColor = Color.LightYellow;

    //two frame difference
    if (motionDetectionType == 0)
    {
        //Blob counting
        if (motionProcessingType == 0)
        {
            motionDetector = new MotionDetector(TFDdetector, BCOProcessing);
            motionDetector.ProcessFrame(captureImage);
        }
        //Grid motion area
        else if (motionProcessingType == 1)
        {
            motionDetector = new MotionDetector(TFDdetector, GMAprocessing);
            motionDetector.ProcessFrame(captureImage);
        }
        //motion area highlight
        else if (motionProcessingType == 2)
        {
            motionDetector = new MotionDetector(TFDdetector, MAHprocessing);
            motionDetector.ProcessFrame(captureImage);
        }
        //motion border highlight
        else if (motionProcessingType == 3)
        {
            motionDetector = new MotionDetector(TFDdetector, MBHprocessing);
            motionDetector.ProcessFrame(captureImage);
        }
        //none
        else
        {
            //          return;
        }
    }
}
```

```

//simple backgroud modeling
else if (motionDetectionType == 1)
{
    //Blob counting
    if (motionProcessingType == 0)
    {
        motionDetector = new MotionDetector(SBMdetector, BCOprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
    //Grid motion area
    else if (motionProcessingType == 1)
    {
        motionDetector = new MotionDetector(SBMdetector, GMAprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
    //motion area highlight
    else if (motionProcessingType == 2)
    {
        motionDetector = new MotionDetector(SBMdetector, MAHprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
    //motion border highlight
    else if (motionProcessingType == 3)
    {
        motionDetector = new MotionDetector(SBMdetector, MBHprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
    //none
    else
    {
        //    return;
    }
}
//custom frame diference
else if (motionDetectionType == 2)
{
    //Blob counting
    if (motionProcessingType == 0)
    {
        motionDetector = new MotionDetector(CFDdetector, BCOprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
    //Grid motion area
    else if (motionProcessingType == 1)
    {
        motionDetector = new MotionDetector(CFDdetector, GMAprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
    //motion area highlight
    else if (motionProcessingType == 2)
    {
        motionDetector = new MotionDetector(CFDdetector, MAHprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
    //motion border highlight
    else if (motionProcessingType == 3)
    {
        motionDetector = new MotionDetector(CFDdetector, MBHprocessing);
        motionDetector.ProcessFrame(captureImage);
    }
}

```

```

        //none
        else
        {
            //    return;
        }
    }
    pictureBox1.Image = captureImage;
}

```

7. Tambahkan beberapa fungsi berikut :

```

private void radioButtonInit()
{
    radioButtonTwoFrame.Checked = false;
    radioButtonCustomFrame.Checked = false;
    radioButtonBackSub.Checked = false;
    radioButtonBlob.Checked = false;
    radioButtonGrid.Checked = false;
    radioButtonHighlight.Checked = false;
    radioButtonBorder.Checked = false;
}

private void setMotionDetectionType(int iType)
{
    motionDetectionType = iType;
}

private void setMotionProcessingType(int iType)
{
    motionProcessingType = iType;
}

```

8. Tambahkan fungsi berikut untuk mengakses kamera

```

private static string _usbcamera;
public string usbcamera
{
    get { return _usbcamera; }
    set { _usbcamera = value; }
}

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 { }
}

```



```

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 { }
}

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());
    }
}

```

9. Double klik form dan tambahkan program berikut :

```
private void Form1_Load(object sender, EventArgs e)
{
    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;
}
```

10. Tambahkan event untuk **Form1\_FormClosed** sbg berikut :

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

11. Tambahkan event untuk **videoSourcePlayer1\_NewFrame** sbg berikut :

```
private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap image)
{
    try
    {
        sourceImage = image.Clone() as Bitmap;
        //tracking image
        motionTracking(sourceImage);
    }
    catch
    {
    }
}
```

12. Double klik button **start camera** dan tambahkan program berikut:

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

13. Double klik button **Reset** dan tambahkan program berikut:

```
private void buttonReset_Click(object sender, EventArgs e)
{
    radioButtonInit();
    setMotionDetectionType(5);
    setMotionProcessingType(5);
}
```

14. Double klik button **Close** dan tambahkan program berikut:

```
private void buttonClose_Click(object sender, EventArgs e)
{
    Close();
}
```

15. Double klik semua **radioButton** dan tambahkan program berikut:

```
private void radioButtonTwoFrame_CheckedChanged(object sender, EventArgs e)
{
    setMotionDetectionType(0);
}

private void radioButtonCustomFrame_CheckedChanged(object sender, EventArgs e)
{
    setMotionDetectionType(1);
}

private void radioButtonBackSub_CheckedChanged(object sender, EventArgs e)
{
    setMotionDetectionType(2);
}

private void radioButtonBlob_CheckedChanged(object sender, EventArgs e)
{
    setMotionProcessingType(0);
}

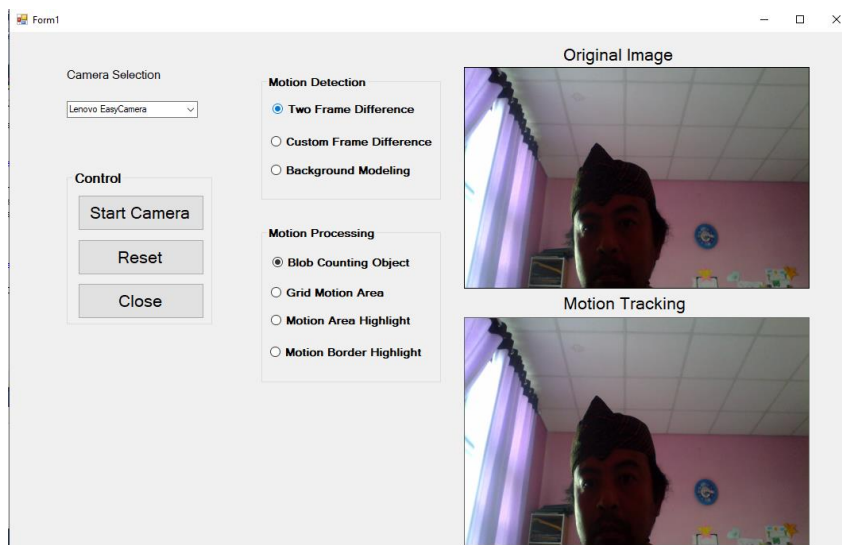
private void radioButtonGrid_CheckedChanged(object sender, EventArgs e)
{
    setMotionProcessingType(1);
}

private void radioButtonHighlight_CheckedChanged(object sender, EventArgs e)
{
    setMotionProcessingType(2);
}

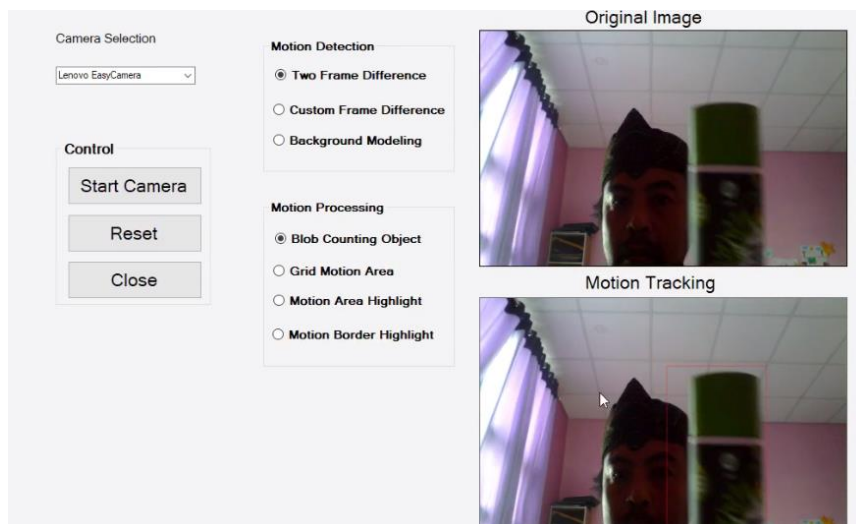
private void radioButtonBorder_CheckedChanged(object sender, EventArgs e)
{
    setMotionProcessingType(3);
}
```

16. Running program yang telah anda buat

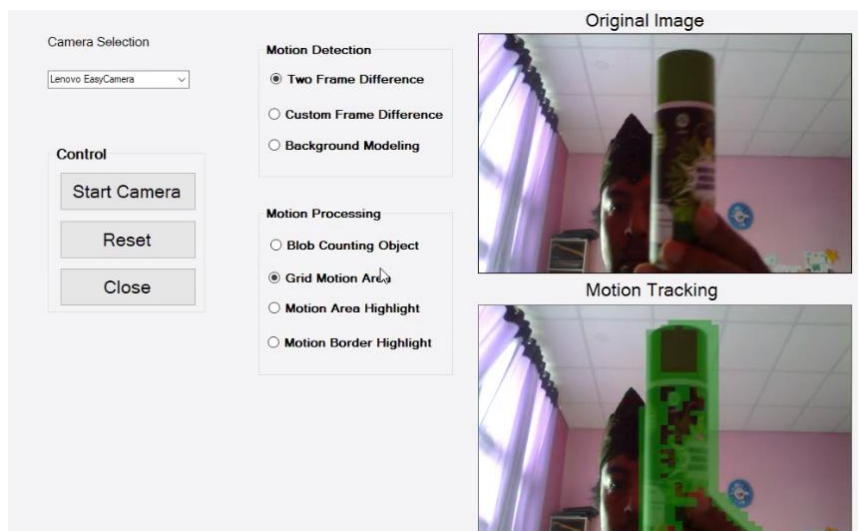
17. Berikut contoh beberapa hasil running program



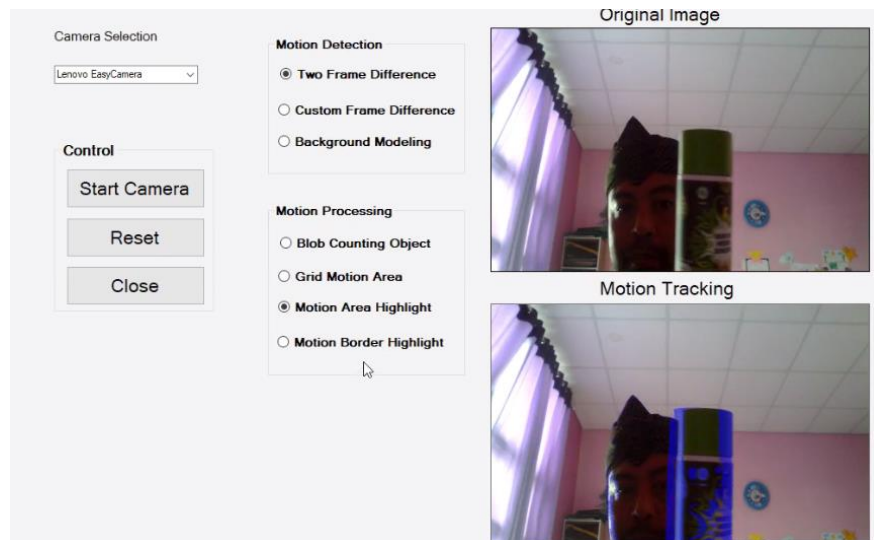
1. Kondisi awal sebelum ada objek yang bergerak



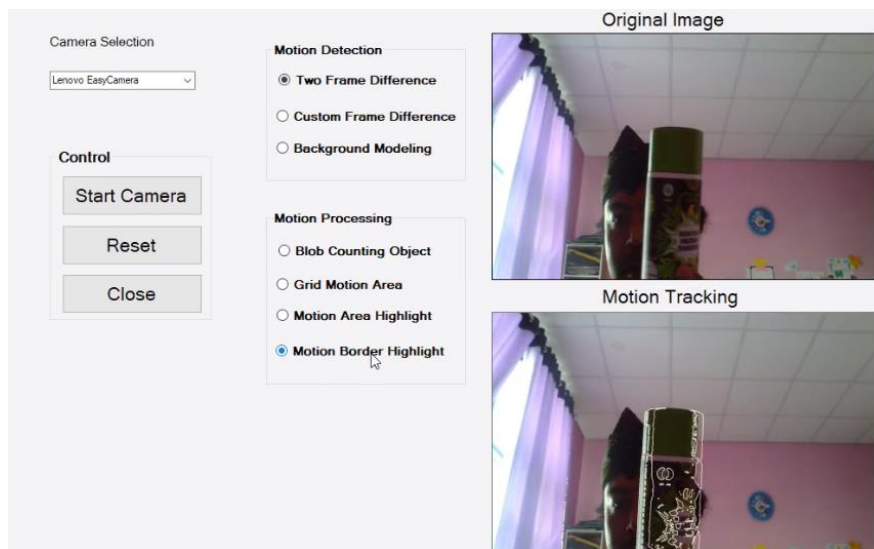
2. Deteksi objek dengan Two Frame difference dan blob counting



3. Deteksi objek dengan Two Frame difference dan Grid Motion Area



4. Deteksi objek dengan Two Frame difference dan Motion Area Highlight



5. Deteksi objek dengan Two Frame difference dan Motion Border Highlight

18. Silahkan dicoba dengan jenis algoritma yang lain