TUGAS PERCOBAAN 3

PENGOLAHAN CITRA

MK401

POLITEKNIK NEGERI Batam

Disusun oleh :

Ricky Silitonga (4211901034)

PROGRAM STUDI TEKNIK MEKATRONIKA

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI BATAM

2020

MANIPULASI CITRA RGB

Tugas dan Pertanyaan

1. Tambahkan kode program untuk radioButton G Image, B Image dan Invers Image.

```
// G button
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 2;

    // ganti text label 1
    label1.Text = "Green Image";

    setImageProcessing(num_processing_img);
}

// B button
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 3;

    // ganti text label 1
    label1.Text = "Blue Image";

    setImageProcessing(num_processing_img);
}
// inverse button
private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 4;

    // ubah label 1
    label1.Text = "Inverse Image";
    setImageProcessing(num_processing_img);
}
```

2. Tambahkan kode program pada fungsi setImageProcessing(int procNo) untuk menampilkan GImage, BImage dan Invers Image

```
// green image
else if(proc_number == 2)
{
    int g = w.G; // green value
    Color greenColor = Color.FromArgb(0, g, 0);
    processing_image.SetPixel(x, y, greenColor);
}
// blue image
else if(proc_number == 3)
{
    int b = w.B; // blue value
```

```
            Color blueColor = Color.FromArgb(0, 0, b);
            processing_image.SetPixel(x, y, blueColor);
        }
        // invers image
        else if(proc_number == 4)
        {
            int rInverse = 255 - w.R;
            int gInverse = 255 - w.G;
            int bInverse = 255 - w.B;

            Color inverse_color = Color.FromArgb(rInverse, gInverse, bInverse);
            processing_image.SetPixel(x, y, inverse_color);
        }
```

3. Tambahkan kode program untuk radioButton Image Resampling 4, 8, 16 dan 32

```
147     private void radioButton8_CheckedChanged(object sender, EventArgs e) // 4
148     {
149         if (radioButton7.Checked == false) return;
150         // change the label to resample image
151         label1.Text = "Resample image 8";
152         setResampleLevel(4);
153         imageResample();
154     }
        1 reference
155     private void radioButton9_CheckedChanged(object sender, EventArgs e) // 8
156     {
157         if (radioButton7.Checked == false) return;
158         // change the label to resample image
159         label1.Text = "Resample image 16";
160         setResampleLevel(8);
161         imageResample();
162     }
        1 reference
163     private void radioButton10_CheckedChanged(object sender, EventArgs e) // 16
164     {
165         if (radioButton7.Checked == false) return;
166         // change the label to resample image
167         label1.Text = "Resample image 16";
168         setResampleLevel(16);
169         // resample_level = 16;
170         imageResample();
171     }
        1 reference
172     private void radioButton11_CheckedChanged(object sender, EventArgs e) // 32
173     {
174         if (radioButton7.Checked == false) return;
175         // change the label to resample image
176         label1.Text = "Resample image 64";
177         setResampleLevel(32);
178         imageResample();
```

4. Tambahkan kode program untuk radioButton Image Quantization 4, 8, 16 dan 32

```
191     private void radioButton13_CheckedChanged(object sender, EventArgs e) // 4
192     {
193         if (radioButton12.Checked == false) return;
194
195         // change the label to resample image
196         label1.Text = "Quantization image 4";
197         setQuantizationLevel(4);
198         imageQuantization();
199     }
        1 reference
200     private void radioButton14_CheckedChanged(object sender, EventArgs e) //8
201     {
202         if (radioButton12.Checked == false) return;
203         // change the label to resample image
204         label1.Text = "Quantization image 8";
205         setQuantizationLevel(8);
206         imageQuantization();
207     }
        1 reference
208     private void radioButton15_CheckedChanged(object sender, EventArgs e) // 16
209     {
210         if (radioButton12.Checked == false) return;
211         // change the label to resample image
212         label1.Text = "Quantization image 16";
213         setQuantizationLevel(16);
214         imageQuantization();
215     }
        1 reference
216     private void radioButton16_CheckedChanged(object sender, EventArgs e) // 32
217     {
218         if (radioButton12.Checked == false) return;
219         // change the label to resample image
220         label1.Text = "Quantization image 32";
221         setQuantizationLevel(32);
222         imageQuantization();
```

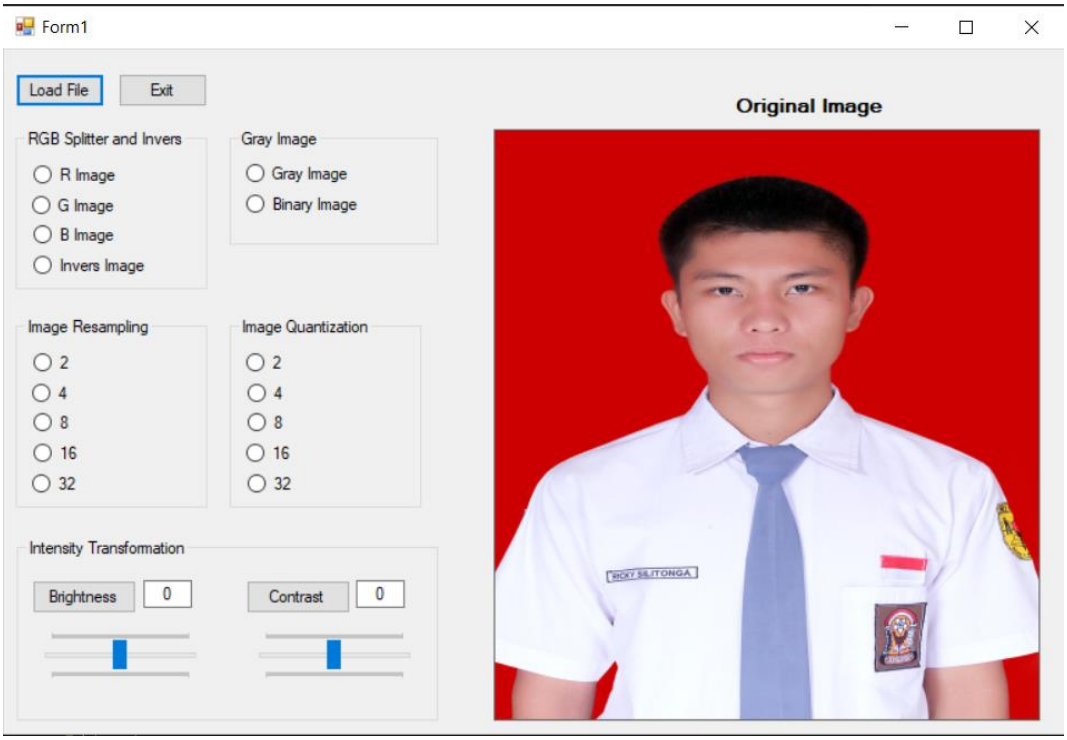
5. Tambahkan kode program untuk trackBar Brightness

```
if (processing_image == null) return;
int brightness = (int)trackBar1.Value;

// seting contrast
setBrightness(brightness);

// text box
    textBox1.Text = string.Format("{0}", trackBar1.Value);
```

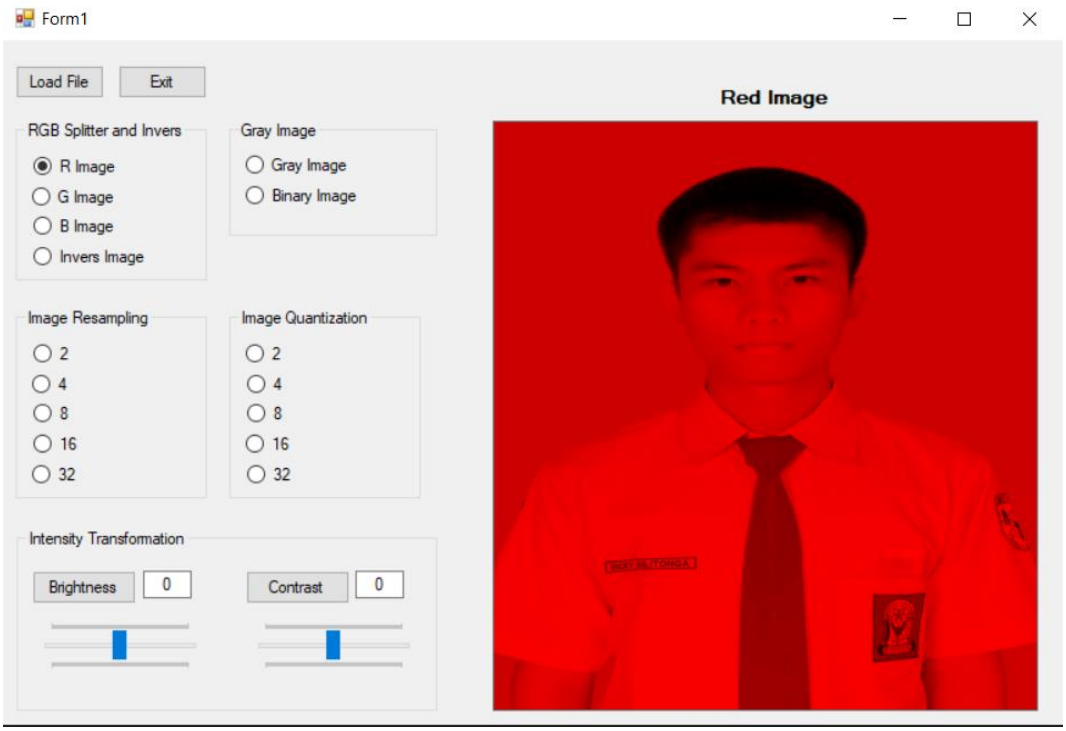
6. Tambahkan kode program untuk button Contrast

```
if (processing_image == null) return;
double contrast = double.Parse(textBox1.Text);
if (contrast < 0 || contrast > 255) return;
// setting brightness
setContrast(contrast);

// menampilkan nilai pada trackbar
    trackBar1.Value = int.Parse(textBox1.Text);
```

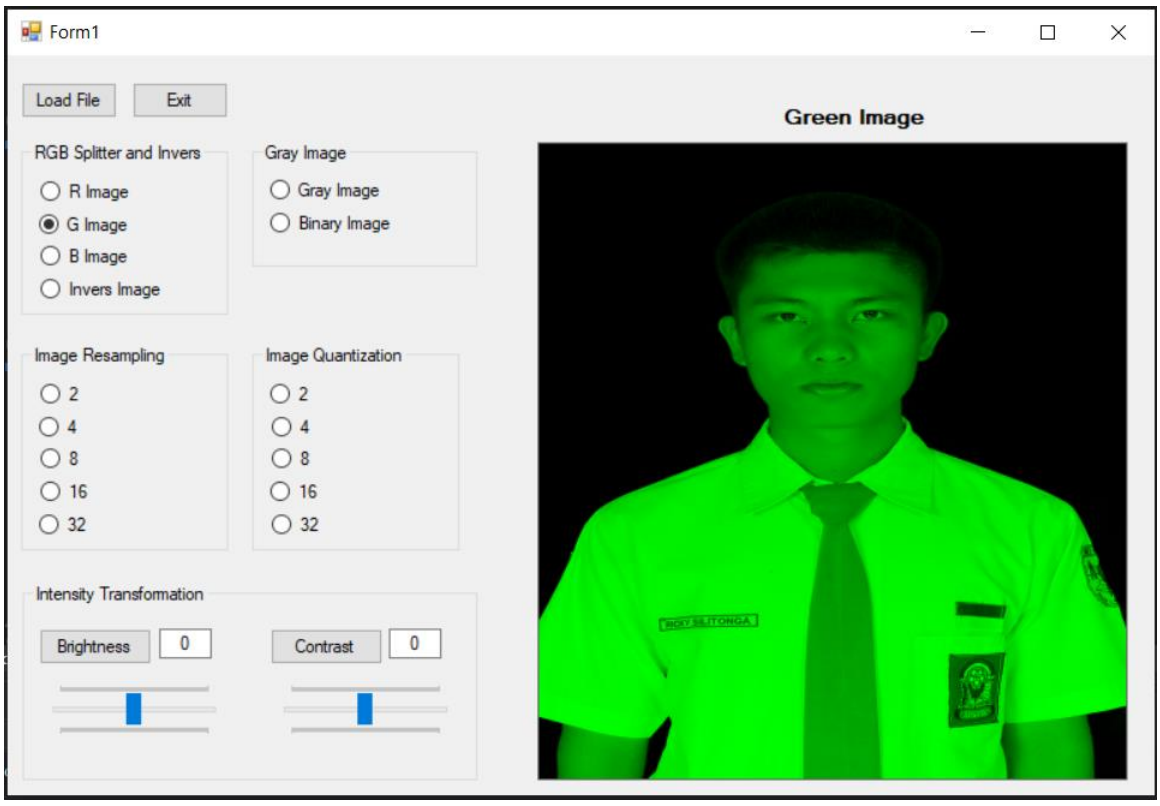
Hasil screen shoot program



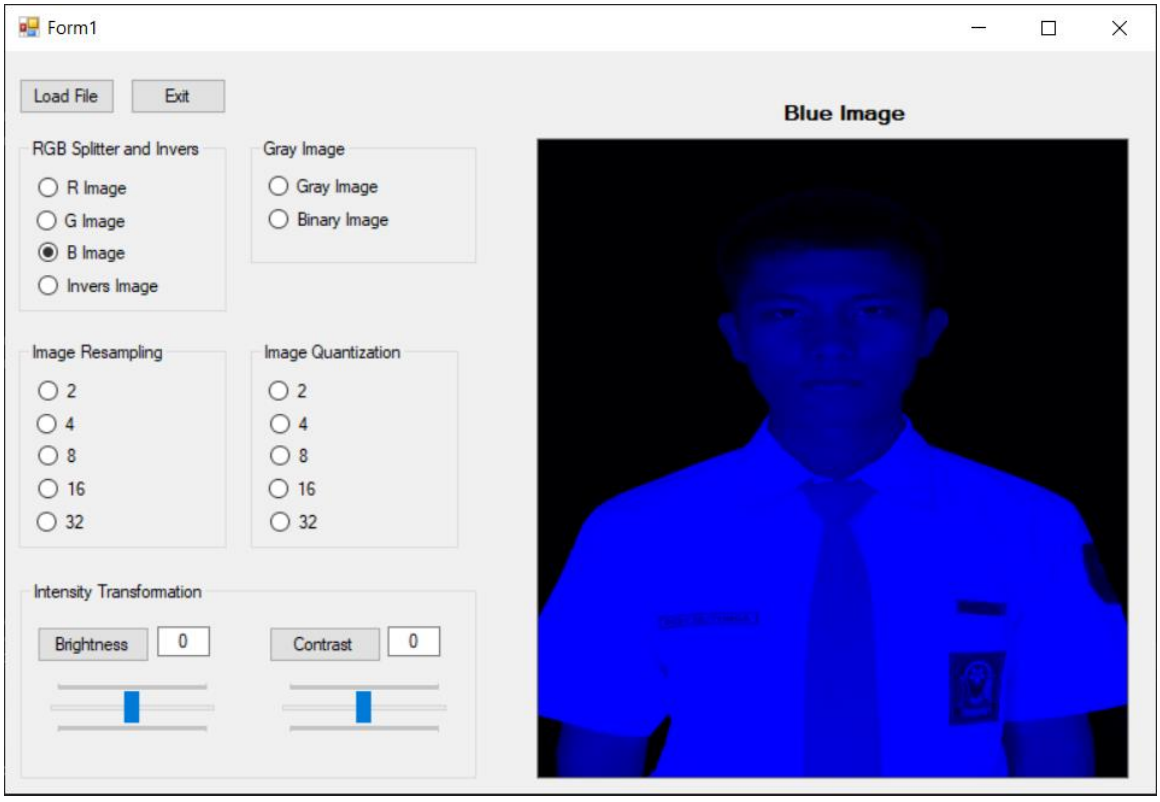
Gambar Asli



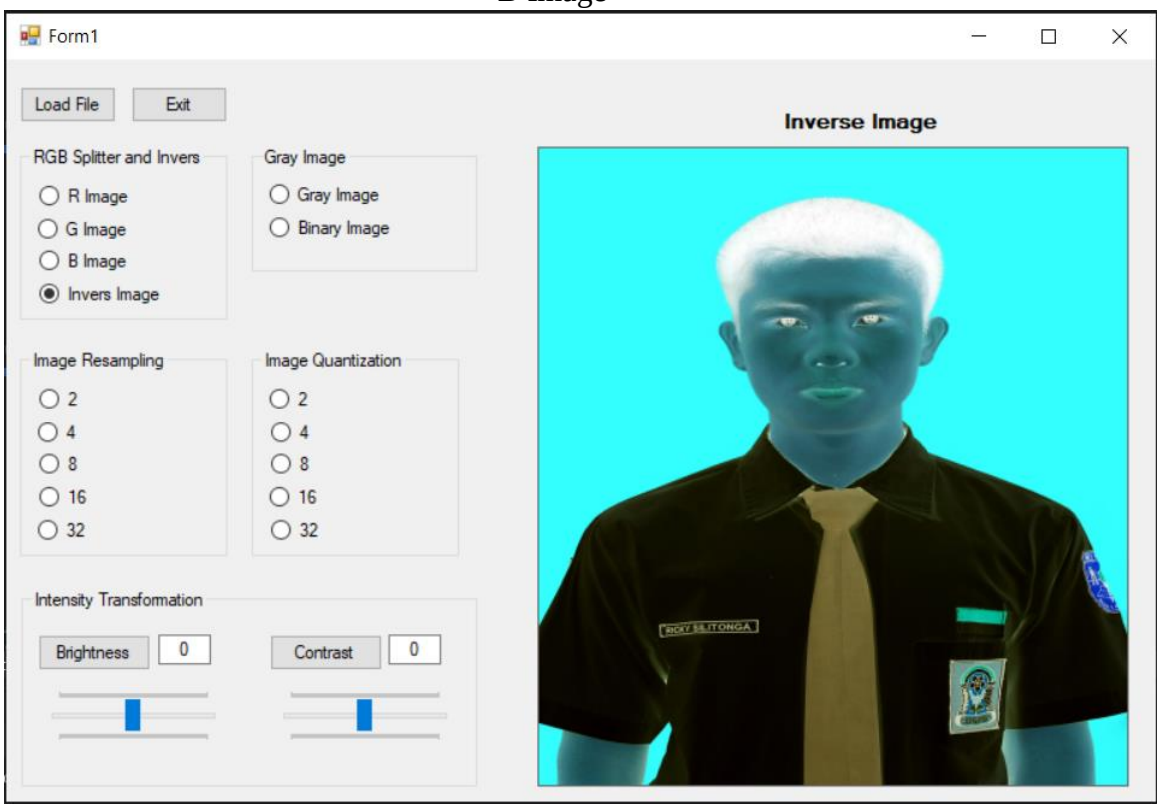
R Image


Form1
Load File
Exit
RGB Splitter and Invers
R Image
G Image
B Image
Invers Image
Gray Image
Gray Image
Binary Image
Image Resampling
2
4
8
16
32
Image Quantization
2
4
8
16
32
Intensity Transformation
Brightness
0
Contrast
0
Green Image

G Image

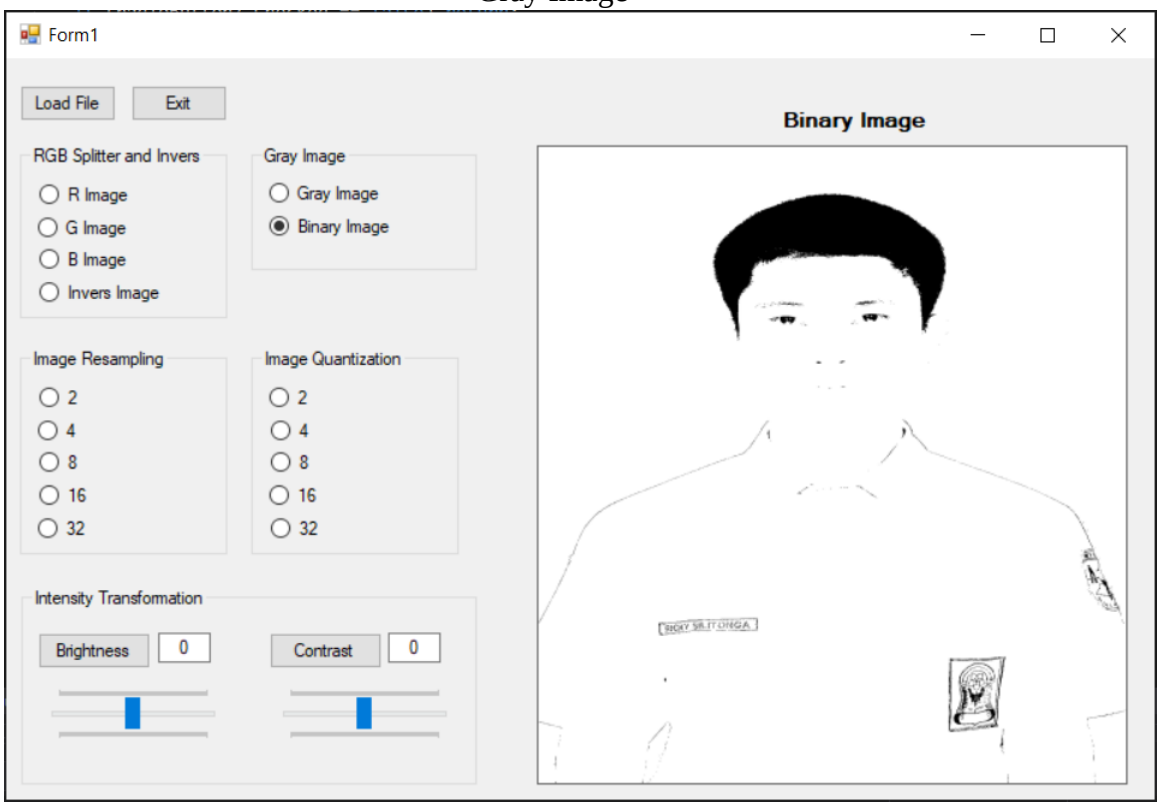

Form1
Load File
Exit
RGB Splitter and Invers
R Image
G Image
B Image
Invers Image
Gray Image
Gray Image
Binary Image
Image Resampling
2
4
8
16
32
Image Quantization
2
4
8
16
32
Intensity Transformation
Brightness
0
Contrast
0
Blue Image

B Image

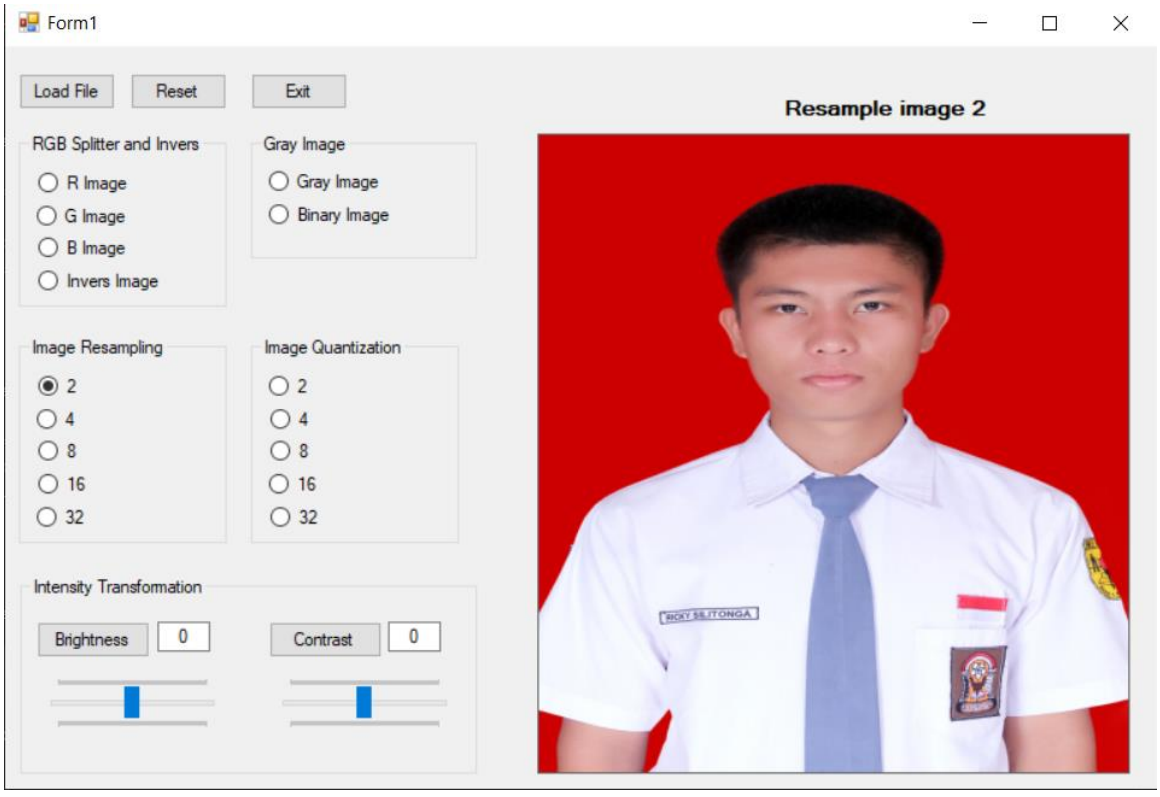


Inverse Image

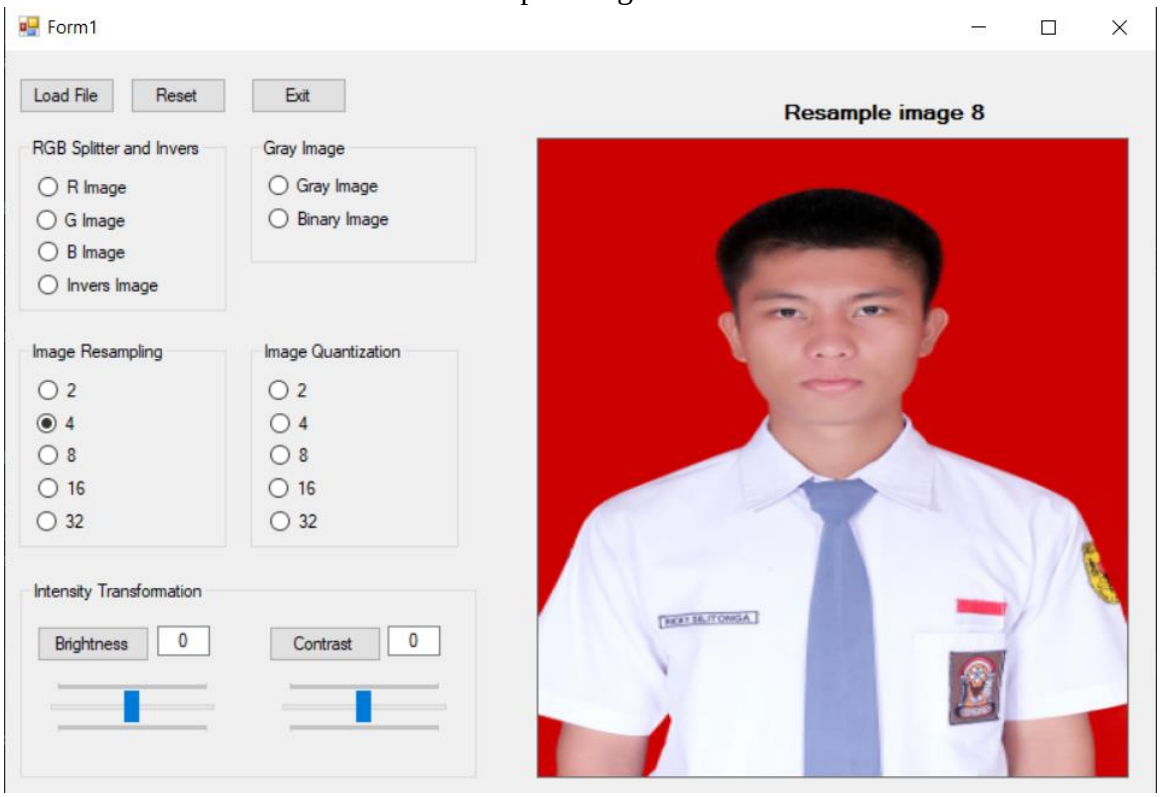
Gray Image



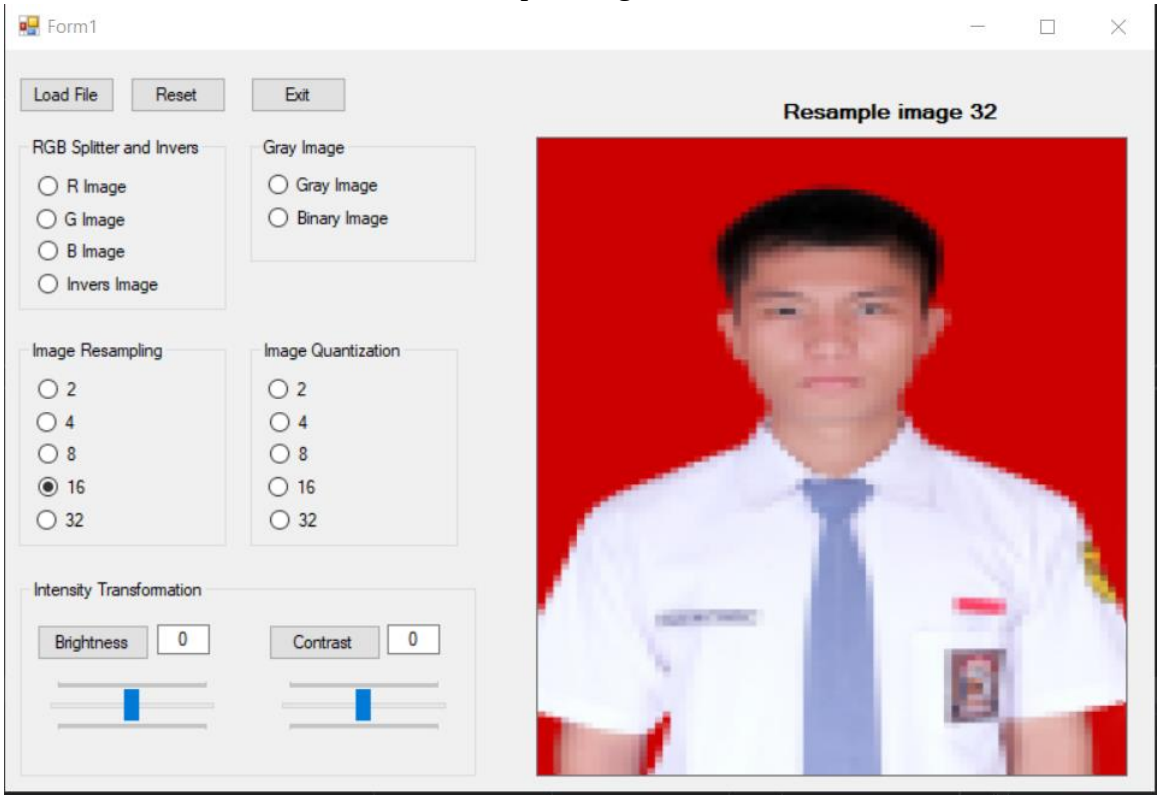
Binary Image



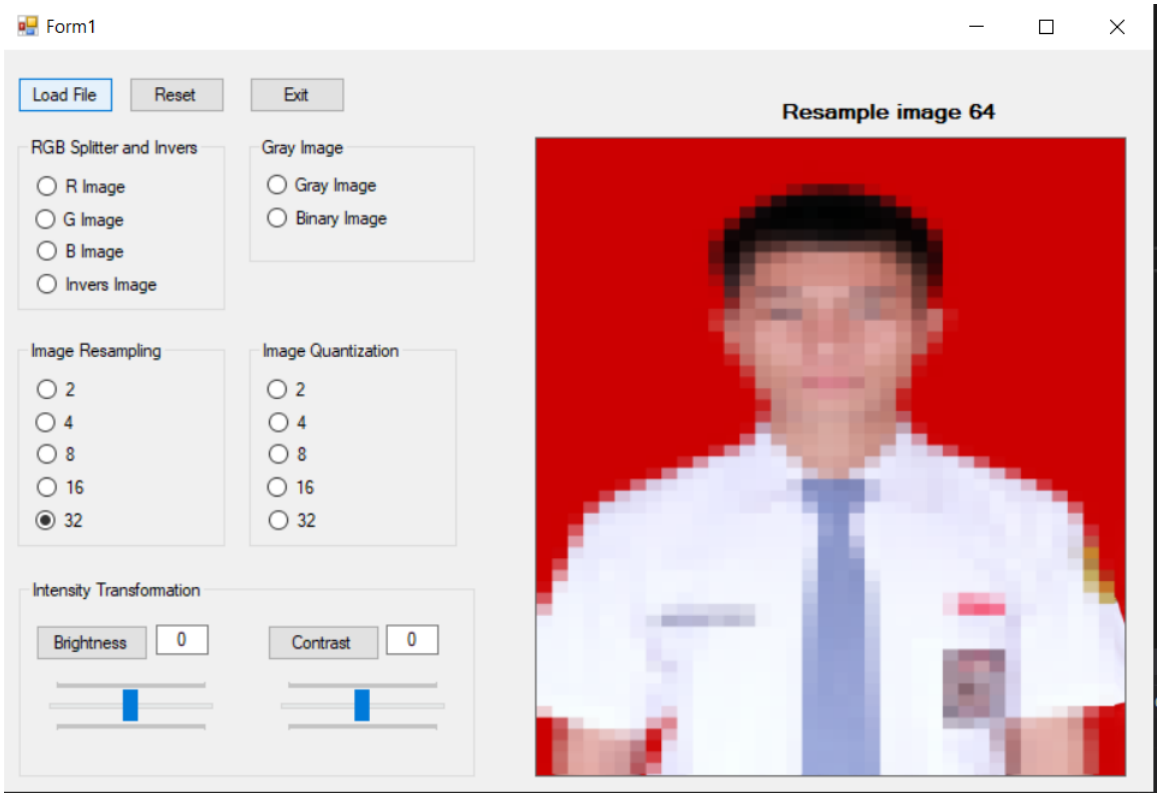
Resample Image 2



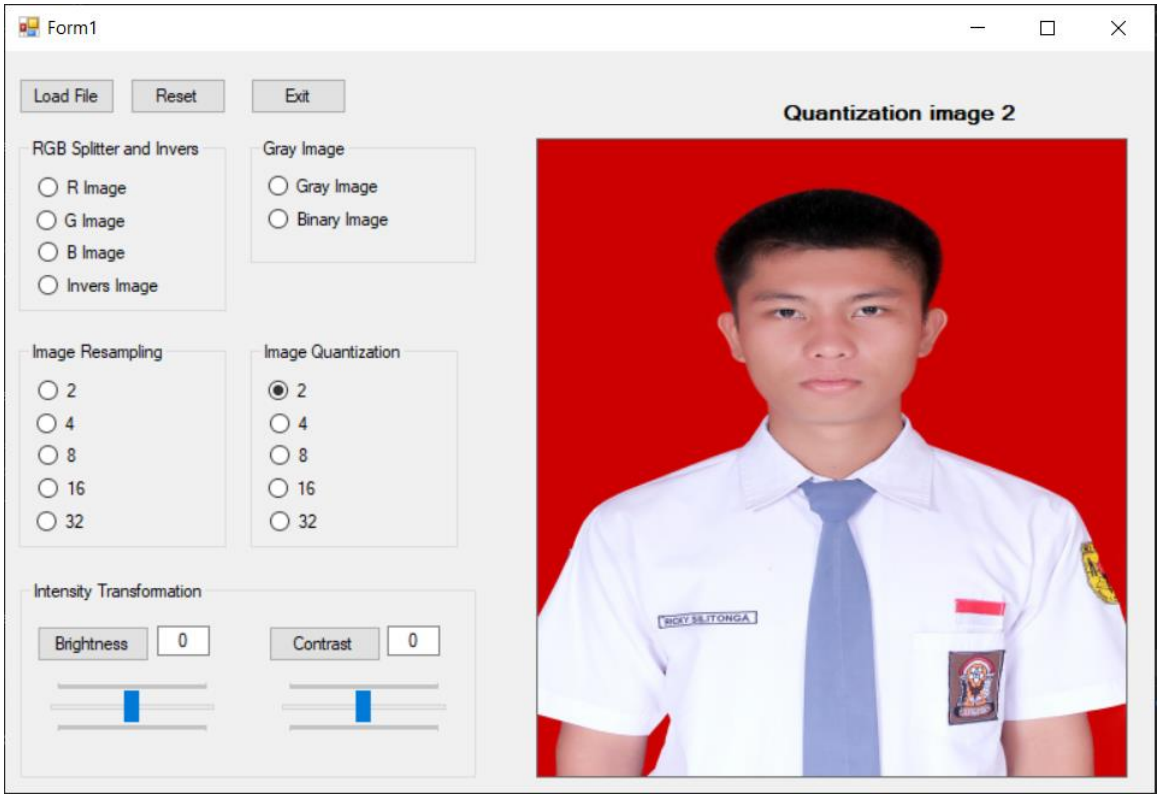
Resample Image 4



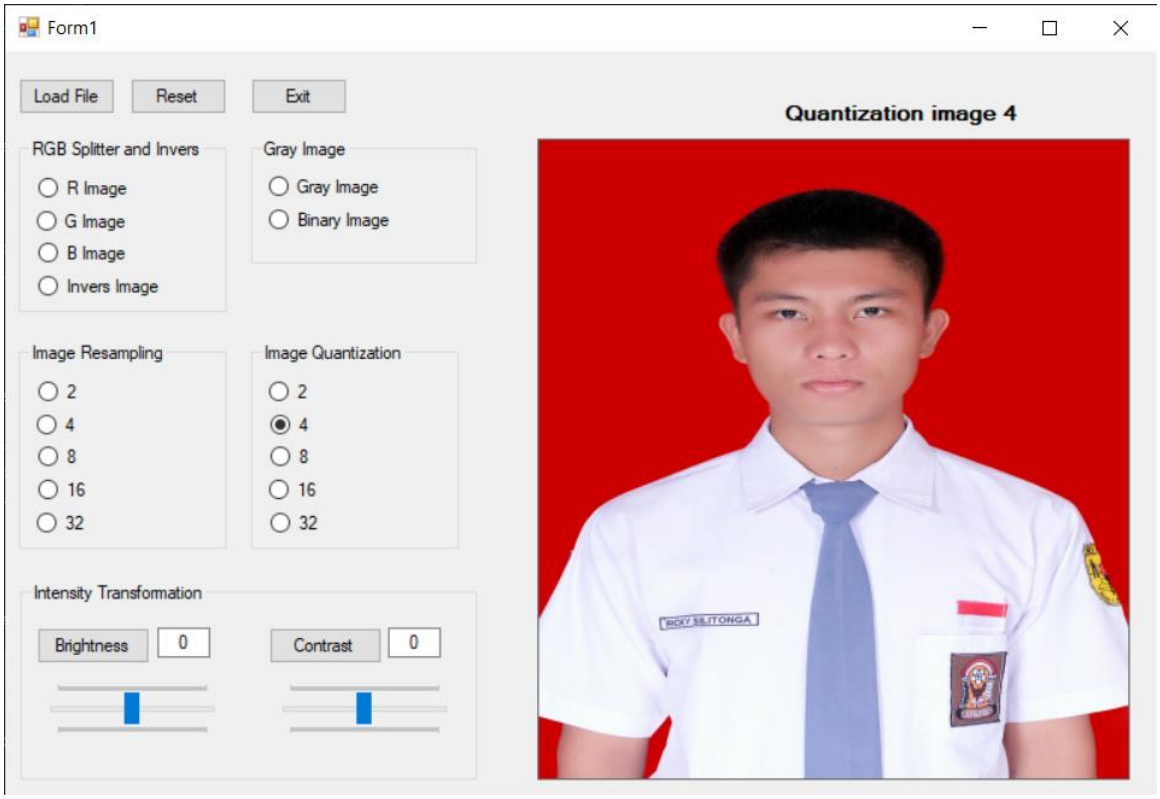
Resample Image 16



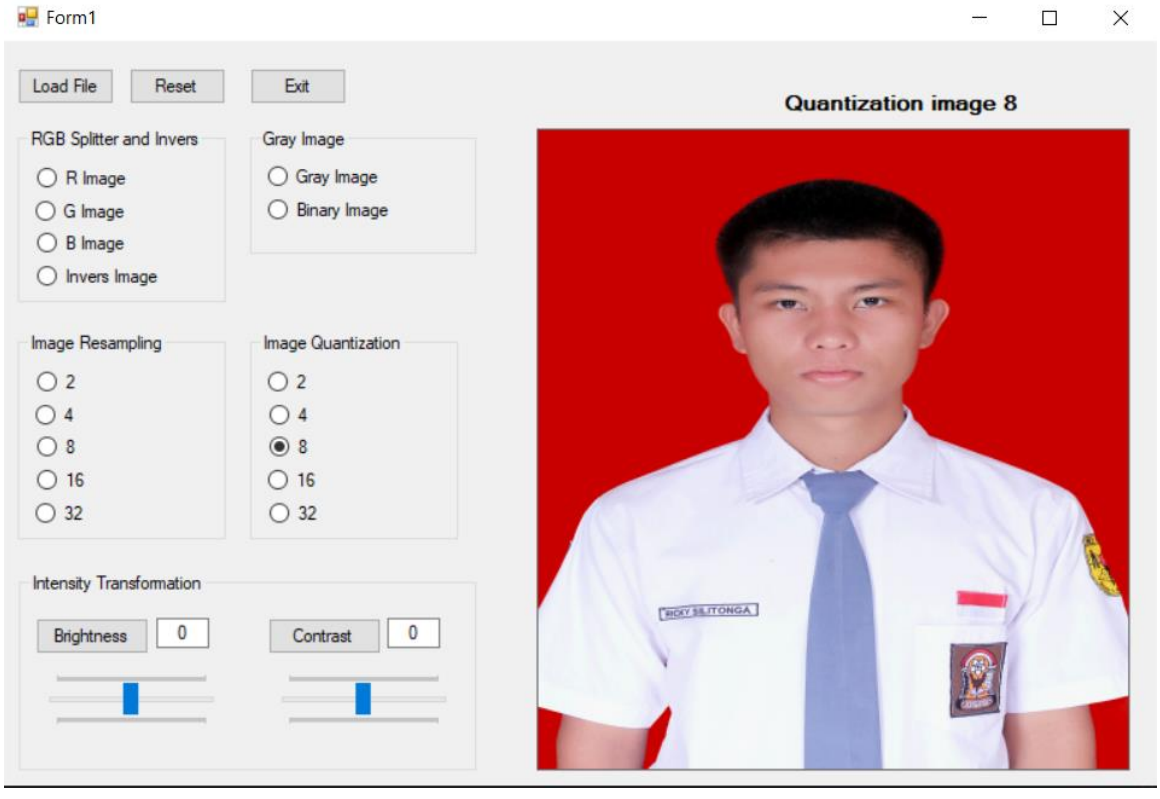
Resample Image 32



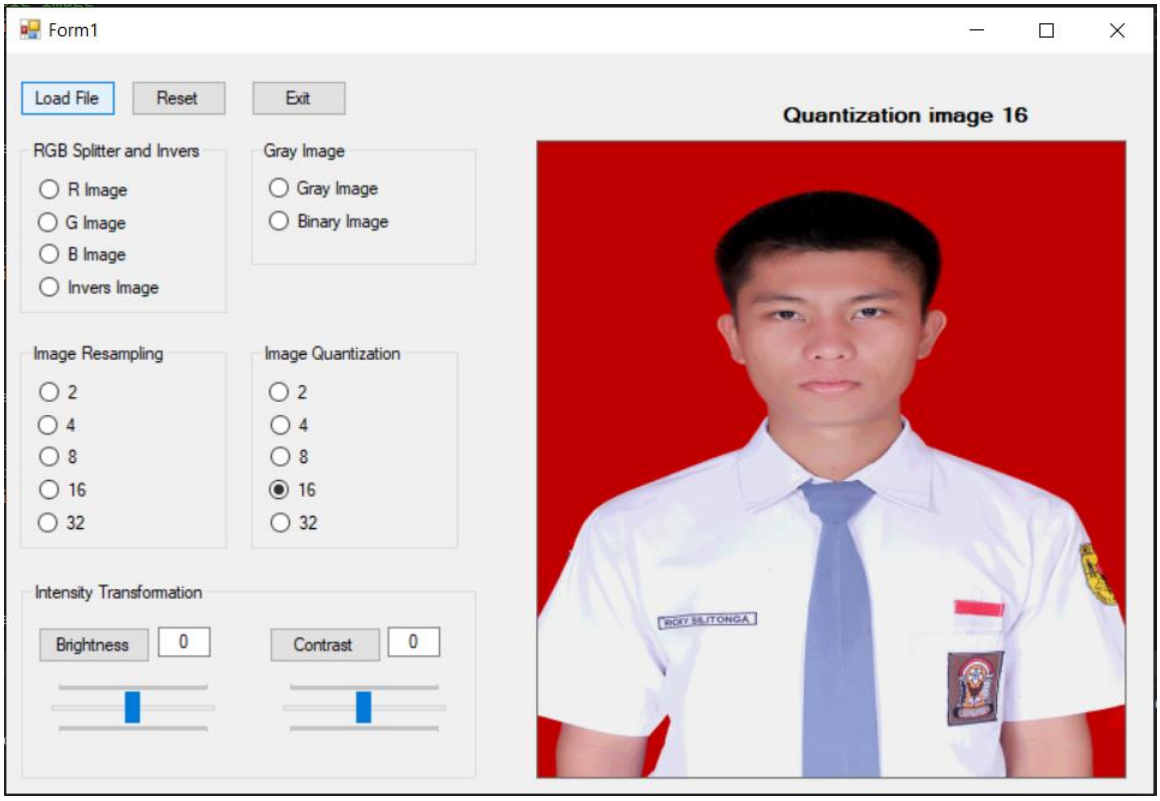
Quantization Image 2



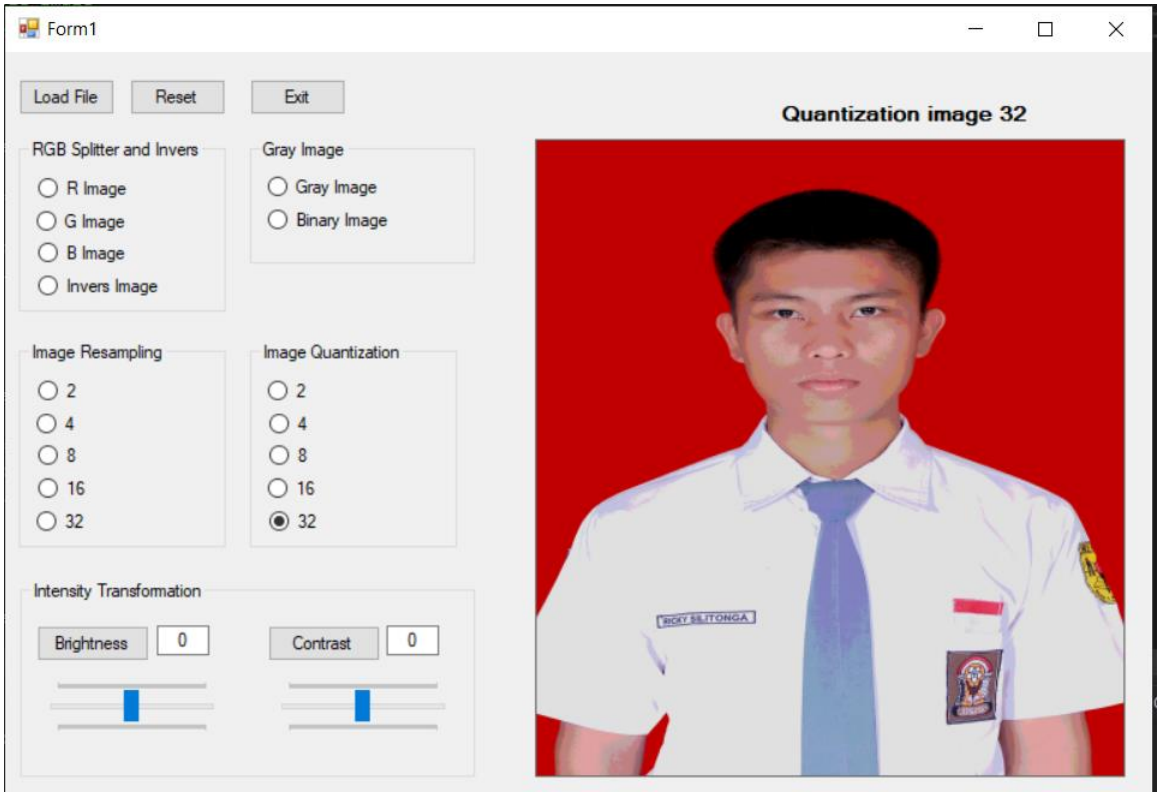
Quantization Image 4



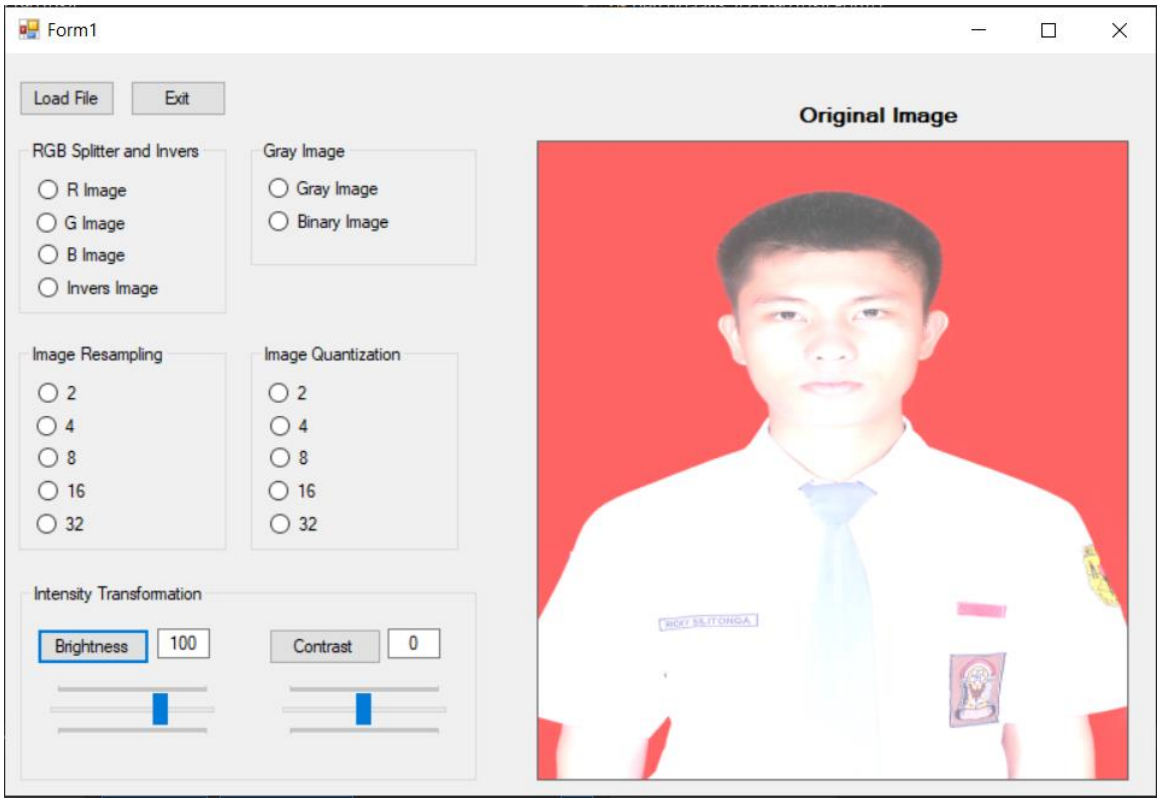
Quantization Image 8



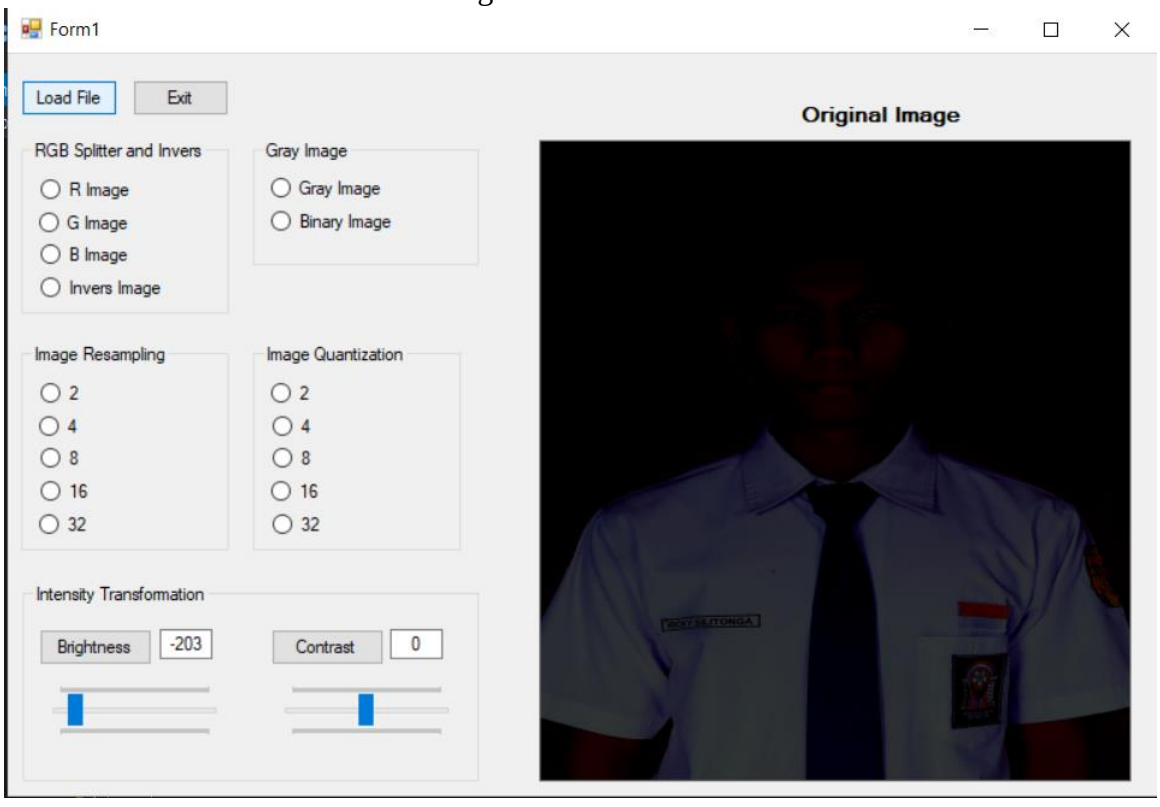
Quantization Image 16



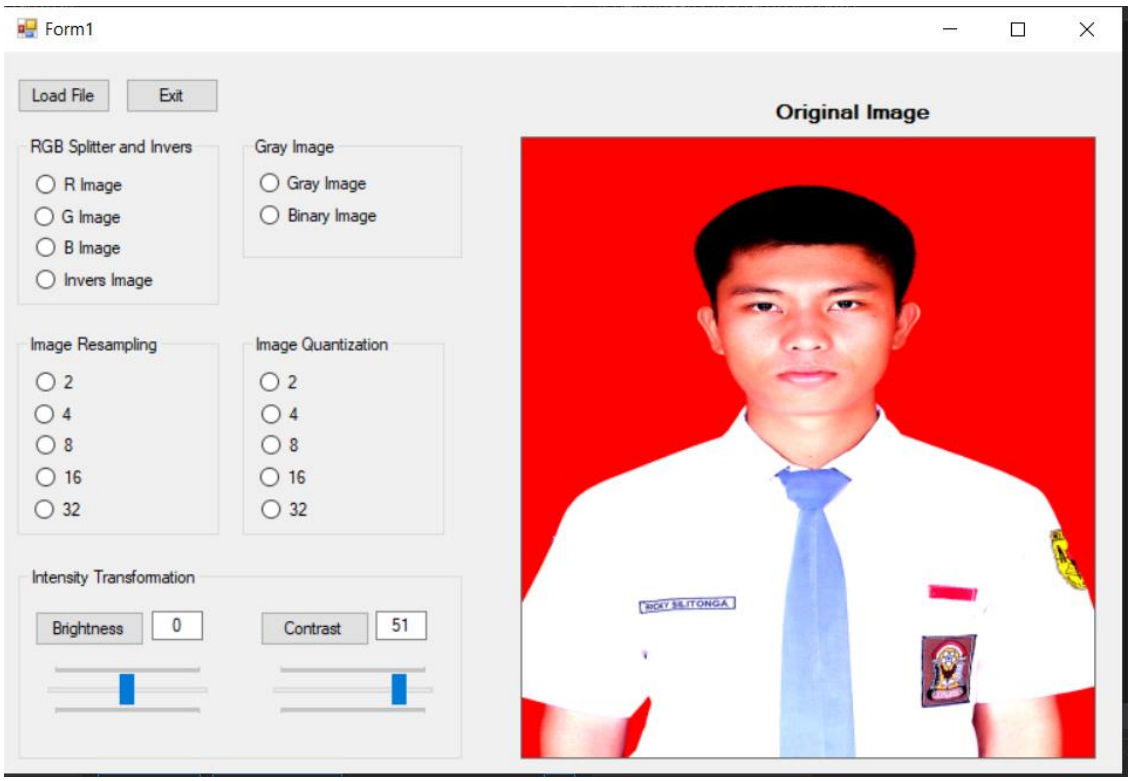
Quantization Image 32



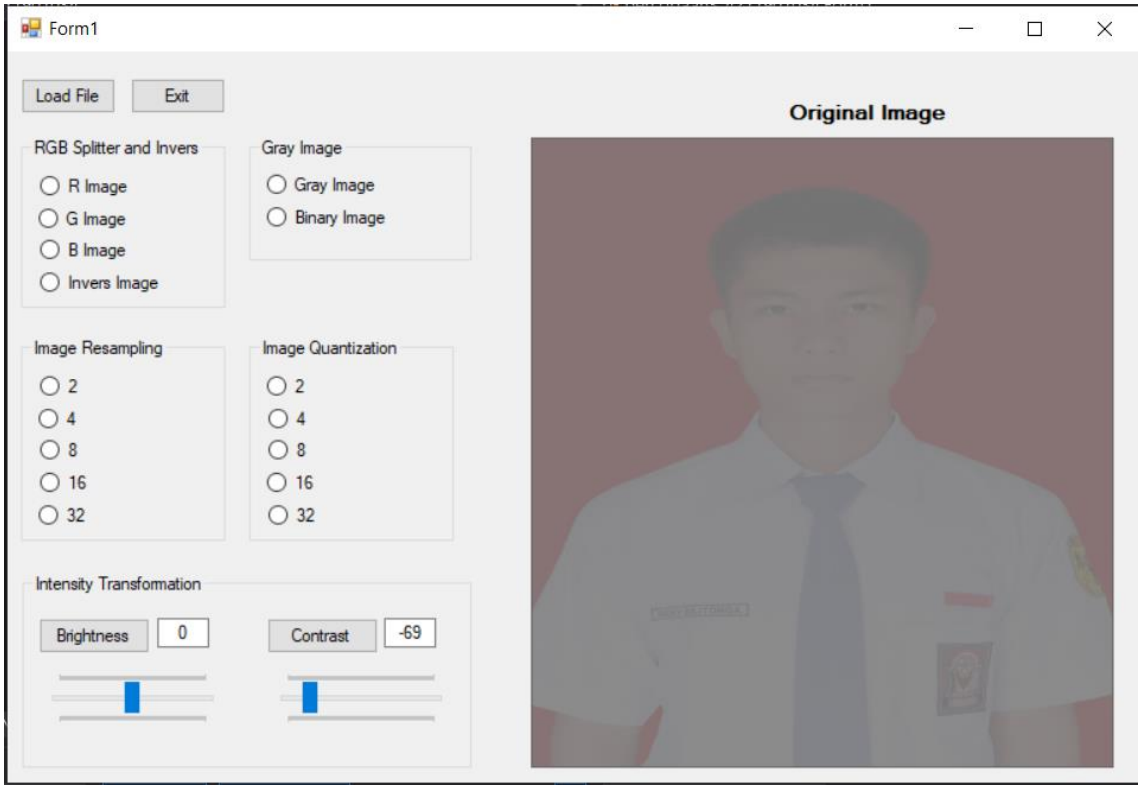
Brightness Button



Brightness Trackbar



Contrast button



Contrast Trackbar

Sourcecode

```
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.Drawing.Imaging;
using System.IO;

namespace percobaan3_4211901034
{
   public partial class Form1 : Form
   {
      // global variable
      Bitmap source_image, processing_image;
      int image_height, image_width;

      // number of processing image
      int num_processing_img;

      // resample level
      int resample_level;

      // level of intensity quantization
      int quantization_level;
      public Form1()
      {
         InitializeComponent();
         trackbarInitialization();
         textboxInitialization();
      }

      // load file button
      private void button1_Click(object sender, EventArgs e)
      {
         if(openFileDialog1.ShowDialog() == DialogResult.OK)
         {
            // loading source image
```

```
        source_image = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);
        processing_image = new Bitmap(source_image);
        // tampilkan di picture box
        pictureBox1.Image = source_image;

        // tinggi dan lebar image;
        image_width = source_image.Width;
        image_height = source_image.Height;
    }
}
// exit button
private void button2_Click(object sender, EventArgs e)
{
    Close();
}

// openfiledialog1
private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{
    source_image = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);
    pictureBox1.Image = source_image;
}

// radio button RGB Splitter dan invers

// R Button
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 1;

    // ganti text label 1
    label1.Text = "Red Image";

    setImageProcessing(num_processing_img);
}

// G button
private void radioButton2_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 2;
```

```
    // ganti text label 1
    label1.Text = "Green Image";

    setImageProcessing(num_processing_img);
}

// B button
private void radioButton3_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 3;

    // ganti text label 1
    label1.Text = "Blue Image";

    setImageProcessing(num_processing_img);
}
// inverse button
private void radioButton4_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 4;

    // ubah label 1
    label1.Text = "Inverse Image";
    setImageProcessing(num_processing_img);
}

// gray image groupbox
// gray image button
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
    if (source_image == null) return;
    num_processing_img = 5;

    // ubah label 1
    label1.Text = "Gray Image";
    setImageProcessing(num_processing_img);
}

// binary image button
private void radioButton6_CheckedChanged(object sender, EventArgs e)
{
```

```
    if (source_image == null) return;
    num_processing_img = 6;

    // ubah label 1
    label1.Text = "Binary Image";
    setImageProcessing(num_processing_img);
}


// image resampling radio button
private void radioButton7_CheckedChanged(object sender, EventArgs e) // 2
{
    if (radioButton7.Checked == false) return;

    // change the label to resample image
    label1.Text = "Resample image 2";
    setResampleLevel(2);
    imageResample();
}
private void radioButton8_CheckedChanged(object sender, EventArgs e) // 4
{
    if (radioButton7.Checked == false) return;
    // change the label to resample image
    label1.Text = "Resample image 8";
    setResampleLevel(4);
    imageResample();
}
private void radioButton9_CheckedChanged(object sender, EventArgs e) // 8
{
    if (radioButton7.Checked == false) return;
    // change the label to resample image
    label1.Text = "Resample image 16";
    setResampleLevel(8);
    imageResample();
}
private void radioButton10_CheckedChanged(object sender, EventArgs e) // 16
{
    if (radioButton7.Checked == false) return;
    // change the label to resample image
    label1.Text = "Resample image 16";
    setResampleLevel(16);
    // resample_level = 16;
    imageResample();
```

```
}
private void radioButton11_CheckedChanged(object sender, EventArgs e) // 32
{
   if (radioButton7.Checked == false) return;
   // change the label to resample image
   label1.Text = "Resample image 64";
   setResampleLevel(32);
   imageResample();
}


// image quantization radio button
private void radioButton12_CheckedChanged(object sender, EventArgs e) // 2
{
   if (radioButton12.Checked == false) return;
   // change the label to resample image
   label1.Text = "Quantization image 2";
   setQuantizationLevel(2);
   imageQuantization();
}
private void radioButton13_CheckedChanged(object sender, EventArgs e) // 4
{
   if (radioButton12.Checked == false) return;

   // change the label to resample image
   label1.Text = "Quantization image 4";
   setQuantizationLevel(4);
   imageQuantization();
}
private void radioButton14_CheckedChanged(object sender, EventArgs e) //8
{
   if (radioButton12.Checked == false) return;
   // change the label to resample image
   label1.Text = "Quantization image 8";
   setQuantizationLevel(8);
   imageQuantization();
}
private void radioButton15_CheckedChanged(object sender, EventArgs e) // 16
{
   if (radioButton12.Checked == false) return;
   // change the label to resample image
   label1.Text = "Quantization image 16";
   setQuantizationLevel(16);
```

```
    imageQuantization();
}
private void radioButton16_CheckedChanged(object sender, EventArgs e) // 32
{
    if (radioButton12.Checked == false) return;
    // change the label to resample image
    label1.Text = "Quantization image 32";
    setQuantizationLevel(32);
    imageQuantization();
}


// brightness button
private void button4_Click(object sender, EventArgs e)
{
    if (processing_image == null) return;
    int brightness = int.Parse(textBox1.Text);
    if (brightness < 0 || brightness > 255) return;
    // setting brightness
    setBrightness(brightness);

    // menampilkan nilai pada trackbar
    trackBar1.Value = int.Parse(textBox1.Text);
}

// contrast button
private void button3_Click(object sender, EventArgs e)
{
    if (processing_image == null) return;
    double contrast = double.Parse(textBox1.Text);
    if (contrast < 0 || contrast > 255) return;
    // setting brightness
    setContrast(contrast);

    // menampilkan nilai pada trackbar
    trackBar1.Value = int.Parse(textBox1.Text);
}

// brightness trackbar
private void trackBar1_Scroll(object sender, EventArgs e)
{
    if (processing_image == null) return;
    int brightness = (int)trackBar1.Value;
```

```
    // seting contrast
    setBrightness(brightness);

    // text box
    textBox1.Text = string.Format("{0}", trackBar1.Value);
}

// contrast trackbar
private void trackBar2_Scroll(object sender, EventArgs e)
{
    if (processing_image == null) return;
    double contrast = (double)trackBar2.Value;

    // seting contrast
    setContrast(contrast);

    // text box
    textBox2.Text = string.Format("{0}", trackBar2.Value);
}



// my function

// initialization
// textbox init
private void textboxInitialization()
{
    textBox1.Text = "0";
    textBox2.Text = "0";
}

// trackbar init
private void trackbarInitialization()
{
    // brightness trackbar
    trackBar1.Minimum = -255;
    trackBar1.Maximum = 255;

    // contrast trackbar
    trackBar2.Minimum = -100;
    trackBar2.Maximum = 100;
```

```
    // init value
    trackBar1.Value = 0;
    trackBar2.Value = 0;
}

// reset condition
private void resetCondition()
{
    // radiobutton reset condition
    radioButton1.Checked = false;
    radioButton2.Checked = false;
    radioButton3.Checked = false;
    radioButton4.Checked = false;
    radioButton5.Checked = false;
    radioButton6.Checked = false;
    radioButton7.Checked = false;
    radioButton8.Checked = false;
    radioButton9.Checked = false;
    radioButton10.Checked = false;
    radioButton11.Checked = false;
    radioButton12.Checked = false;
    radioButton13.Checked = false;
    radioButton14.Checked = false;
    radioButton15.Checked = false;
    radioButton16.Checked = false;

    // trackbar reset condition
    trackBar1.Value = 0;
    trackBar2.Value = 0;

    // text box reset condition
    textBox1.Text = "0";
    textBox2.Text = "0";
}

// processing image funtion
private void setImageProcessing(int proc_number)
{
    for(int x=0; x<image_width; x++)
    {
        for(int y=0; y<image_height; y++)
        {
```

```
// get rgb value of the pixel at (x, y)
Color w = source_image.GetPixel(x, y);

// r image
if(proc_number == 1)
{
   int r = w.R; // red value
   Color redColor = Color.FromArgb(r, 0, 0);

   processing_image.SetPixel(x, y, redColor);
}
// green image
else if(proc_number == 2)
{
   int g = w.G; // green value
   Color greenColor = Color.FromArgb(0, g, 0);
   processing_image.SetPixel(x, y, greenColor);
}
// blue image
else if(proc_number == 3)
{
   int b = w.B; // blue value
   Color blueColor = Color.FromArgb(0, 0, b);
   processing_image.SetPixel(x, y, blueColor);
}
// invers image
else if(proc_number == 4)
{
   int rInverse = 255 - w.R;
   int gInverse = 255 - w.G;
   int bInverse = 255 - w.B;

   Color inverse_color = Color.FromArgb(rInverse, gInverse, bInverse);
   processing_image.SetPixel(x, y, inverse_color);
}
// gray image && binary image
else if(proc_number == 5 || proc_number == 6)
{
   int r = w.R;
   int g = w.G;
   int b = w.B;

   int gray_value = (int)(0.5 * r + 0.419 * g + 0.181 * b);
```

```
                if (gray_value > 255) gray_value = 255; // karena maks  = 255

                // binary image
                if(proc_number == 6)
                {
                   int TH = 100;
                   if (gray_value > TH) gray_value = 255;
                   else gray_value = 0;
                }

                Color gray_color = Color.FromArgb(gray_value, gray_value,
gray_value);
                processing_image.SetPixel(x, y, gray_color);
             }

          }
       }
       pictureBox1.Image = processing_image;
    }

    // seting resample image
    private void setResampleLevel(int iLevel)
    {
       resample_level = iLevel;
    }

    // set Quantization Level
    private void setQuantizationLevel(int iLevel)
    {
       quantization_level = iLevel;
    }

    // image resample
    private void imageResample()
    {
       if (source_image == null) return;

       //resampling to new Width and new Height
       int ht = (int)(image_height / resample_level);
       int wd = (int)(image_width / resample_level);
       int i, j, k, l, new_valueR, new_valueG, new_valueB;
       for (i = 0; i < ht; i++)///
```

```
    {
        for (j = 0; j < wd; j++)///
        {
            new_valueR = 0; new_valueG = 0; new_valueB = 0;
            for (k = 0; k < resample_level; k++)
            {
                for (l = 0; l < resample_level; l++)
                {
                    Color w = source_image.GetPixel(j * resample_level + l, i *
                    resample_level + k);
                    int r = w.R; //red value
                    int g = w.G; //green value
                    int b = w.B; //blue value
                    new_valueR = new_valueR + r;
                    new_valueG = new_valueG + g;
                    new_valueB = new_valueB + b;
                }
            }
            new_valueR = (int)(new_valueR / (resample_level * resample_level));
            new_valueG = (int)(new_valueG / (resample_level * resample_level));
            new_valueB = (int)(new_valueB / (resample_level * resample_level));
            if (new_valueR > 255) new_valueR = 255;
            if (new_valueG > 255) new_valueG = 255;
            if (new_valueB > 255) new_valueB = 255;
            Color colorRed = Color.FromArgb(new_valueR, new_valueG,
            new_valueB);
            for (k = 0; k < resample_level; k++)
            {
                for (l = 0; l < resample_level; l++)
                {
                    processing_image.SetPixel(j * resample_level + l, i *
                    resample_level + k, colorRed);
                }
            }
        }
    }
    pictureBox1.Image = processing_image;
}

// image quantization function
private void imageQuantization()
{
    if (source_image == null) return;
```

```
    for(int x=0; x<image_width; x++)
    {
        for(int y=0; y<image_height; y++)
        {
            Color w = source_image.GetPixel(x, y);
            int r = w.R;
            int g = w.G;
            int b = w.B;

            int rk = quantization_level * (int)(r / quantization_level);
            int gk = quantization_level * (int)(g / quantization_level);
            int bk = quantization_level * (int)(b / quantization_level);

            Color wBaru = Color.FromArgb(rk, gk, bk);
            processing_image.SetPixel(x, y, wBaru);
        }
    }
    pictureBox1.Image = processing_image;
}

// image brightness
private void setBrightness(int brightness)
{
    // inisialisasi bright image
    Bitmap bImage = new Bitmap(processing_image);

    for(int x=0; x<image_width; x++)
    {
        for(int y=0; y<image_height; y++)
        {
            Color w = processing_image.GetPixel(x, y);

            int R = (int)(brightness + w.R);
            if (R > 255) R = 255; if (R < 0) R = 0;
            int G = (int)(brightness + w.G);
            if (G > 255) G = 255; if (G < 0) G = 0;
            int B = (int)(brightness + w.B);
            if (B > 255) B = 255; if (B < 0) B = 0;

            // seting warna baru
            Color wBaru = Color.FromArgb(R, G, B);

            bImage.SetPixel(x, y, wBaru);
```

```
        }
    }
    pictureBox1.Image = bImage;
}

// contrast
private void setContrast(double contrast)
{
    Bitmap cImage = new Bitmap(processing_image);

    contrast = (100.0 + contrast) / 100.0;
    contrast *= contrast;

    for(int x=0; x<image_width; x++)
        for(int y=0; y<image_height; y++)
        {
            Color w = processing_image.GetPixel(x, y);
            double R = w.R / 255.0;
            R -= 0.5;
            R *= contrast;
            R += 0.5;
            R *= 255;
            if (R > 255) R = 255; if (R < 0) R = 0;

            double G = w.G / 255.0;
            G -= 0.5;
            G*= contrast;
            G += 0.5;
            G *= 255;
            if (G > 255) G = 255; if (G < 0) G = 0;

            double B = w.B / 255.0;
            B -= 0.5;
            B *= contrast;
            B += 0.5;
            B *= 255;
            if (B > 255) B = 255; if (B < 0) B = 0;

            Color wBaru = Color.FromArgb((byte)R, (byte)G, (byte)B);

            cImage.SetPixel(x, y, wBaru);
        }
    pictureBox1.Image = cImage;
```

```
        }

        // not used
        private void Form1_Load(object sender, EventArgs e)
        {

        }
        private void groupBox1_Enter(object sender, EventArgs e)
        {

        }

        private void pictureBox1_Click(object sender, EventArgs e)
        {

        }

        private void button5_Click(object sender, EventArgs e)
        {
           resetCondition();
        }

        private void textBox2_TextChanged(object sender, EventArgs e)
        {

        }
        private void label1_Click(object sender, EventArgs e)
        {

        }
    }
}
```