KUIS 1

PENGOLAHAN CITRA

MK401

POLITEKNIK NEGERI Batam

Disusun oleh :

Ricky Silitonga (4211901034)

PROGRAM STUDI TEKNIK MEKATRONIKA

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI BATAM

2020

Buatlah suatu aplikasi pengolahan citra yang dapat melakukan operasi pengolahan citra sbb :

1. Konversi RGB ke Gray

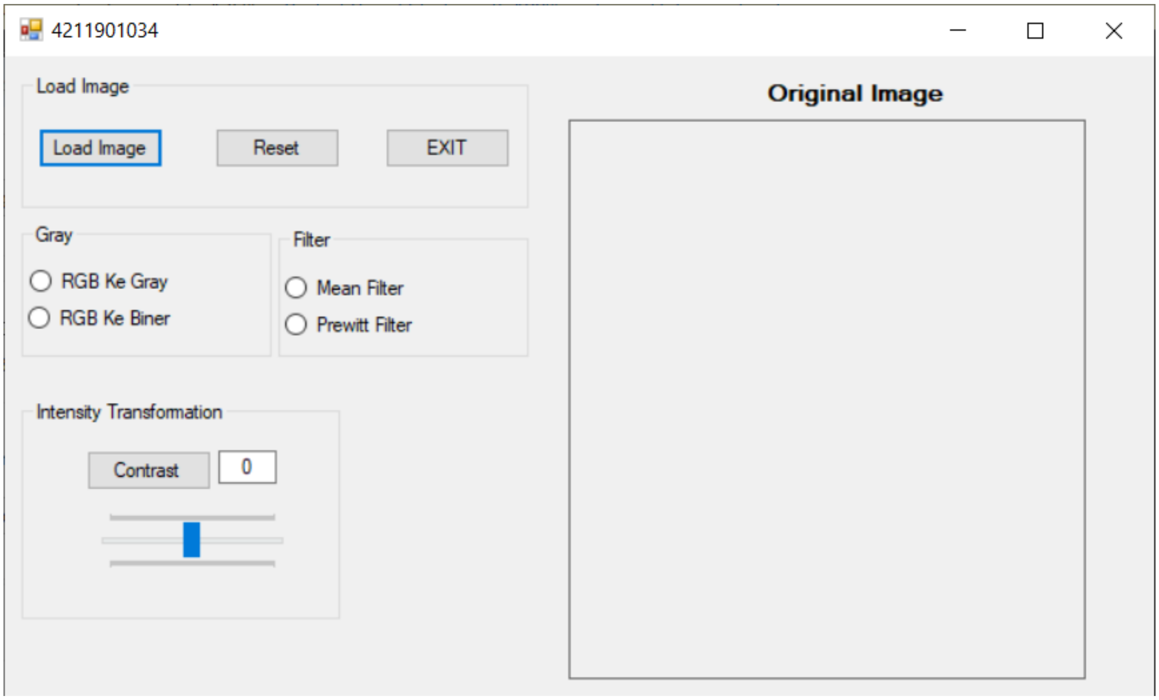
2. Konversi RGB ke Biner

3. RGB intensity transformation (Contrast)

4. Image Smoothing using Mean Filter

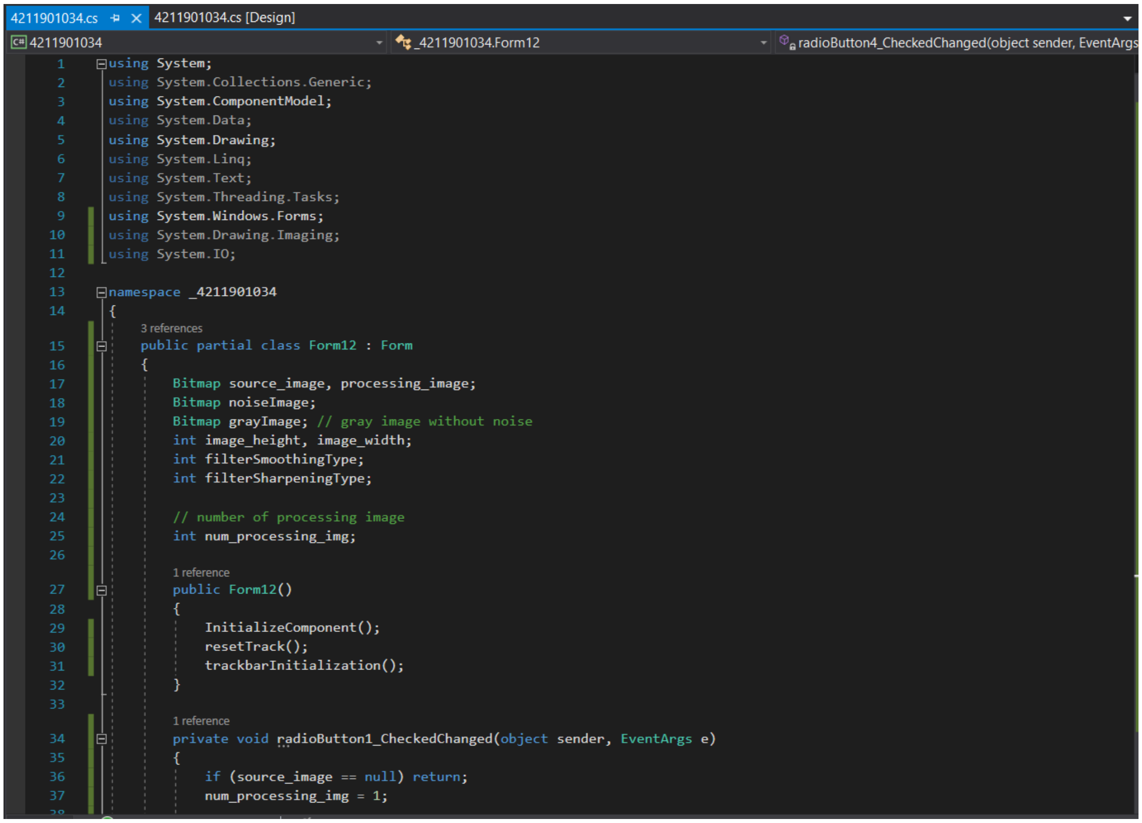
5. Image Sharpening (edge detection) using Prewitt

Screen shoot GUI

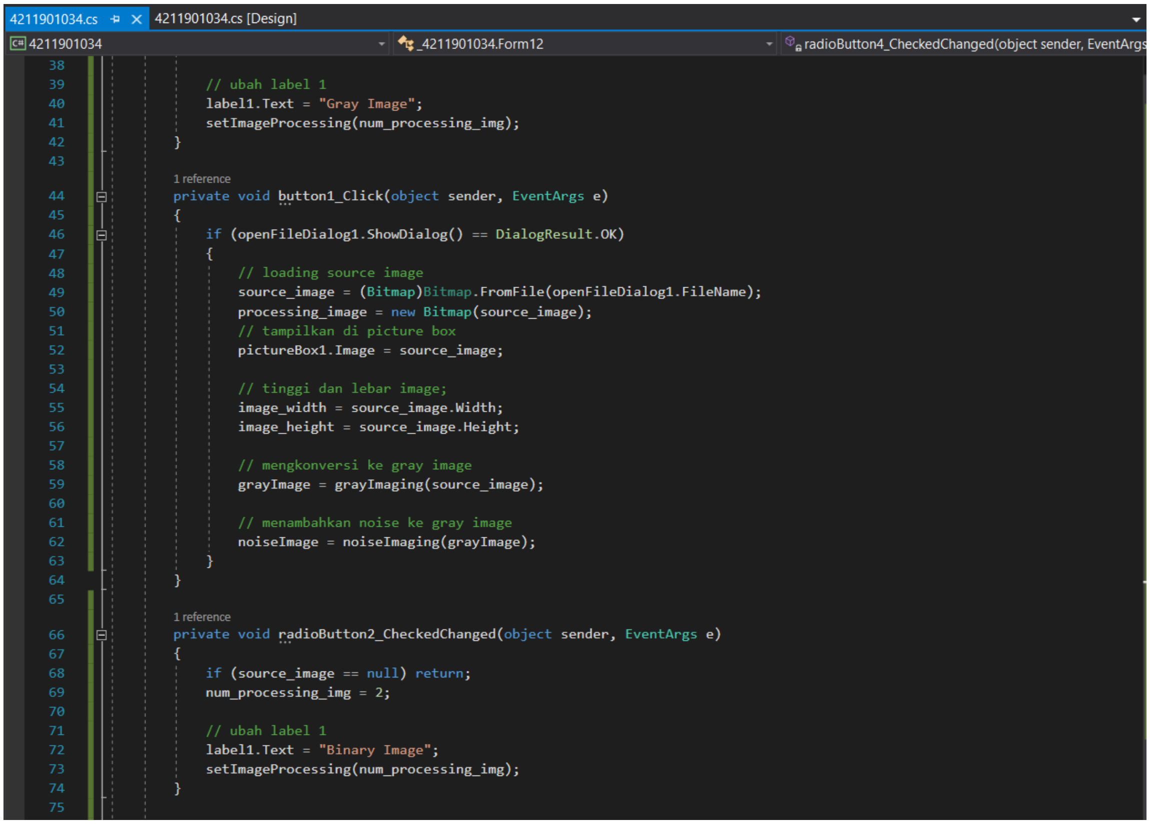


Tampilan awal program

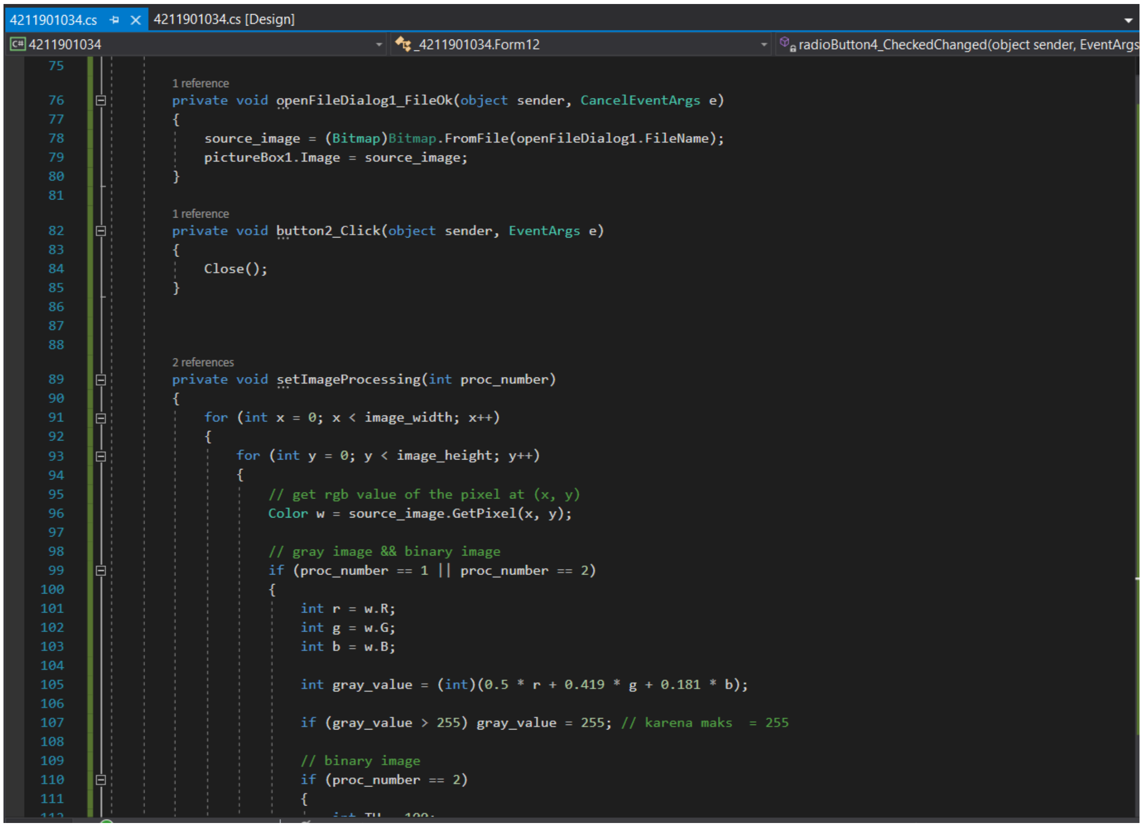
Screenshoot SourceCode



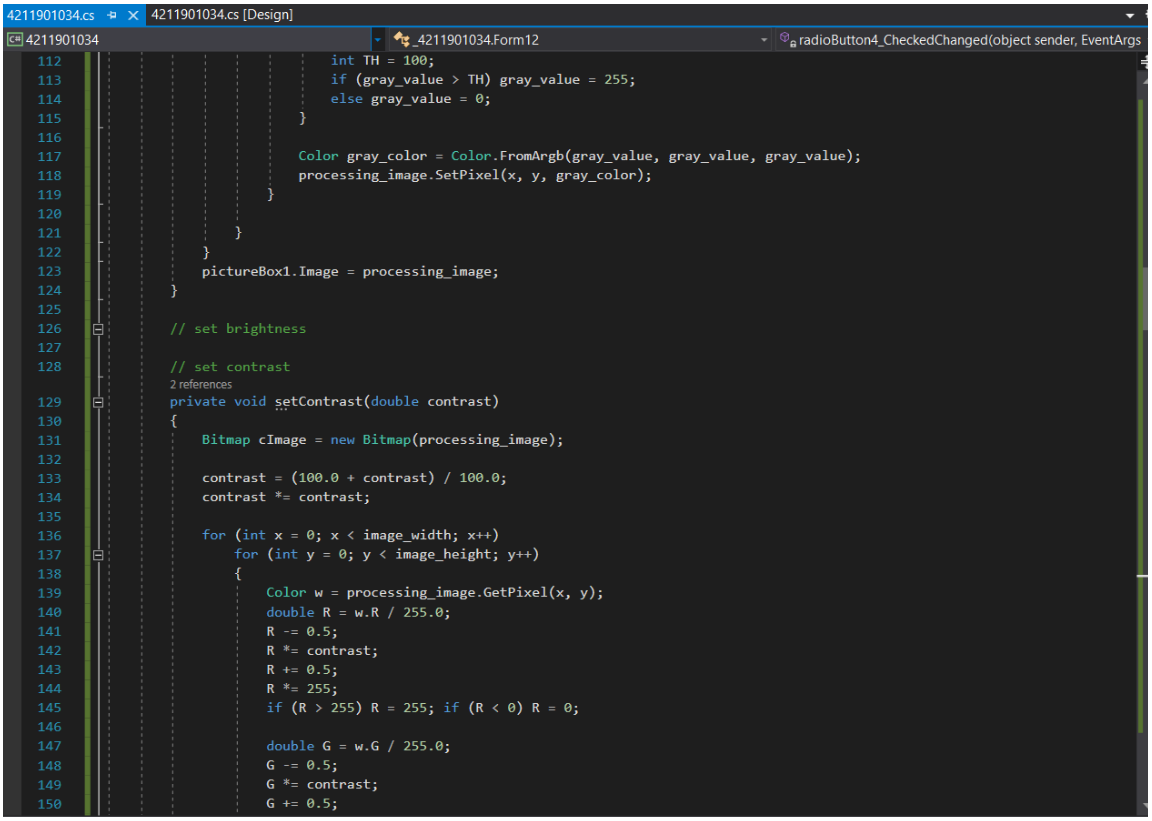
```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing.Imaging;
using System.IO;

namespace _4211901034
{
    public partial class Form12 : Form
    {
        Bitmap source_image, processing_image;
        Bitmap noiseImage;
        Bitmap grayImage; // gray image without noise
        int image_height, image_width;
        int filterSmoothingType;
        int filterSharpeningType;

        // number of processing image
        int num_processing_img;

        public Form12()
        {
            InitializeComponent();
            resetTrack();
            trackbarInitialization();
        }

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



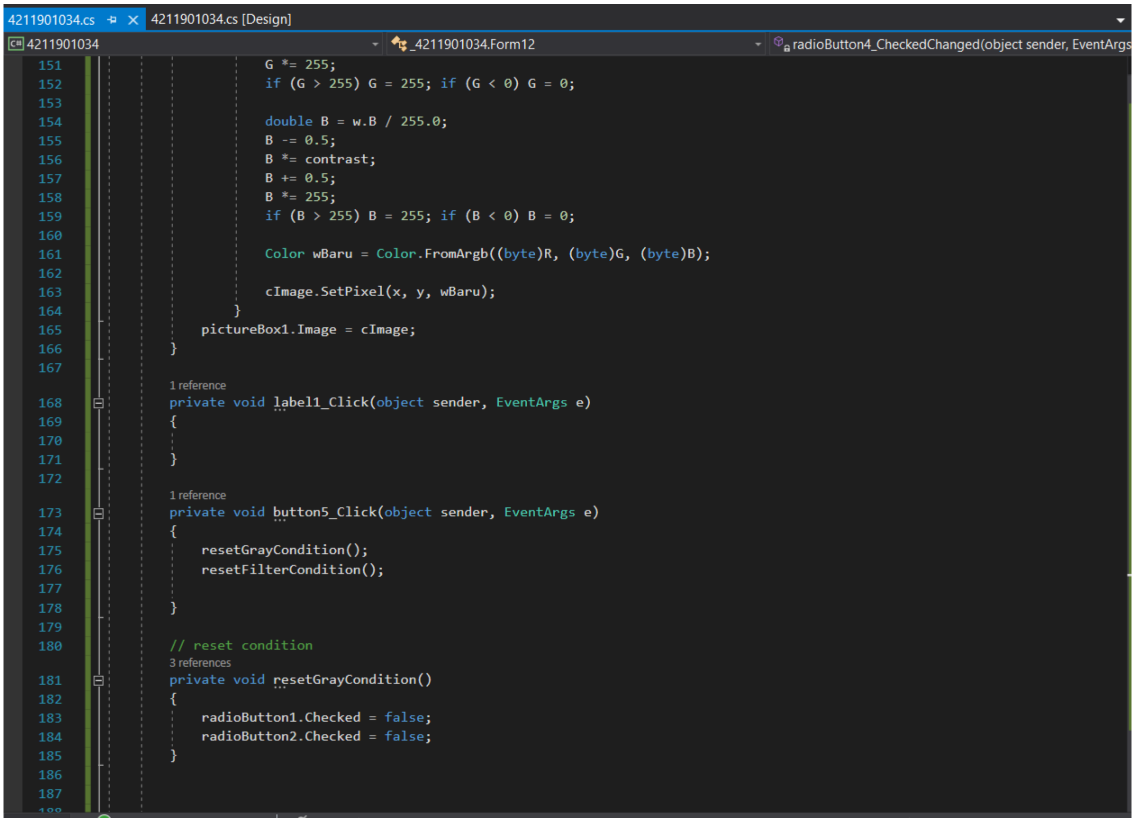
```

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

        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;

                // mengkonversi ke gray image
                grayImage = grayImaging(source_image);

                // menambahkan noise ke gray image
                noiseImage = noiseImaging(grayImage);
            }
        }

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

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

```



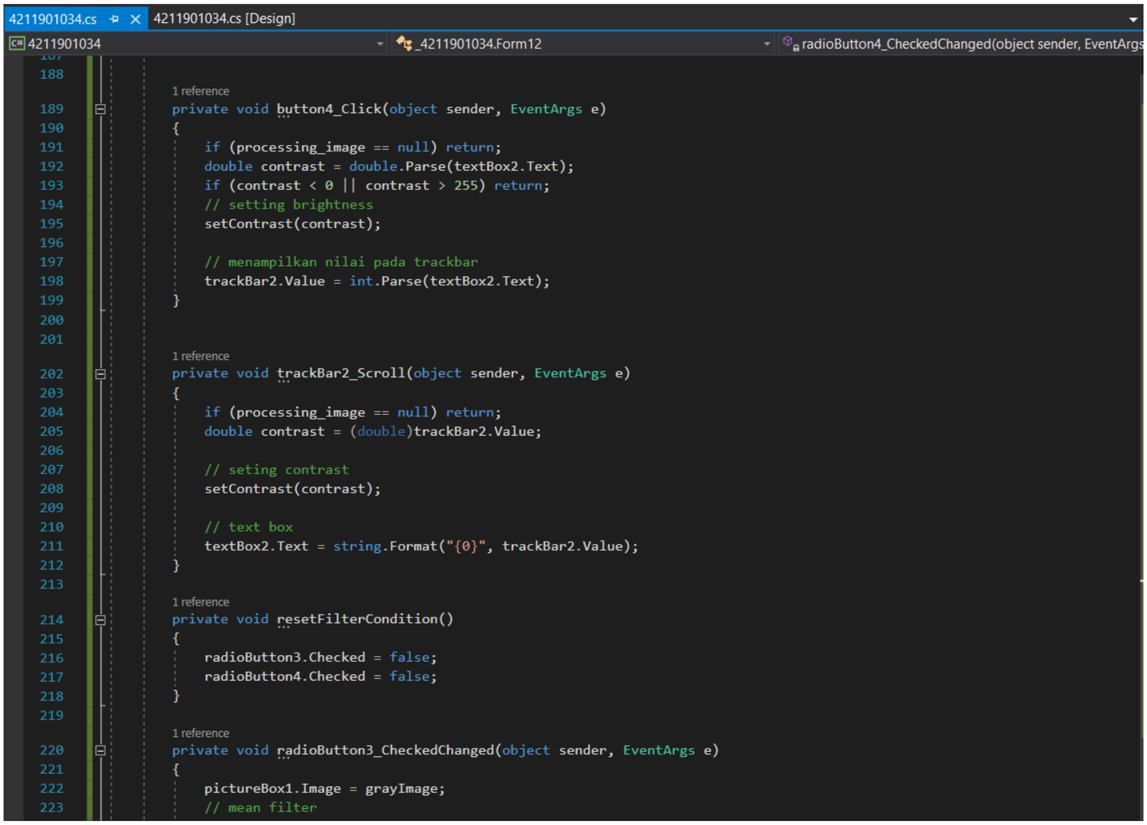
```
        private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
        {
            source_image = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);
            pictureBox1.Image = source_image;
        }

        private void button2_Click(object sender, EventArgs e)
        {
            Close();
        }



        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);

                    // gray image && binary image
                    if (proc_number == 1 || proc_number == 2)
                    {
                        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 == 2)
                        {
```



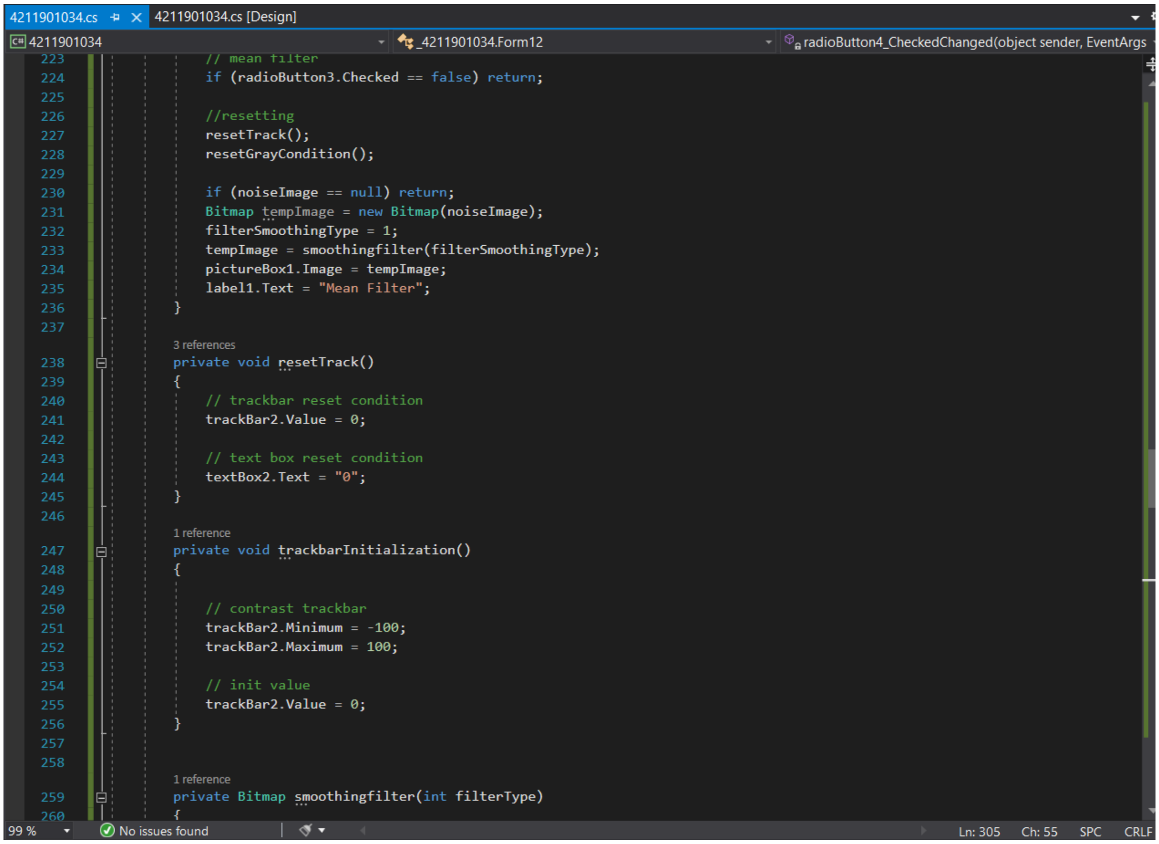
```
                            int TH = 100;
                            if (gray_value > TH) gray_value = 255;
                            else gray_value = 0;
                        }

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

                }
            }
            pictureBox1.Image = processing_image;
        }

        // set brightness

        // set 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;
```



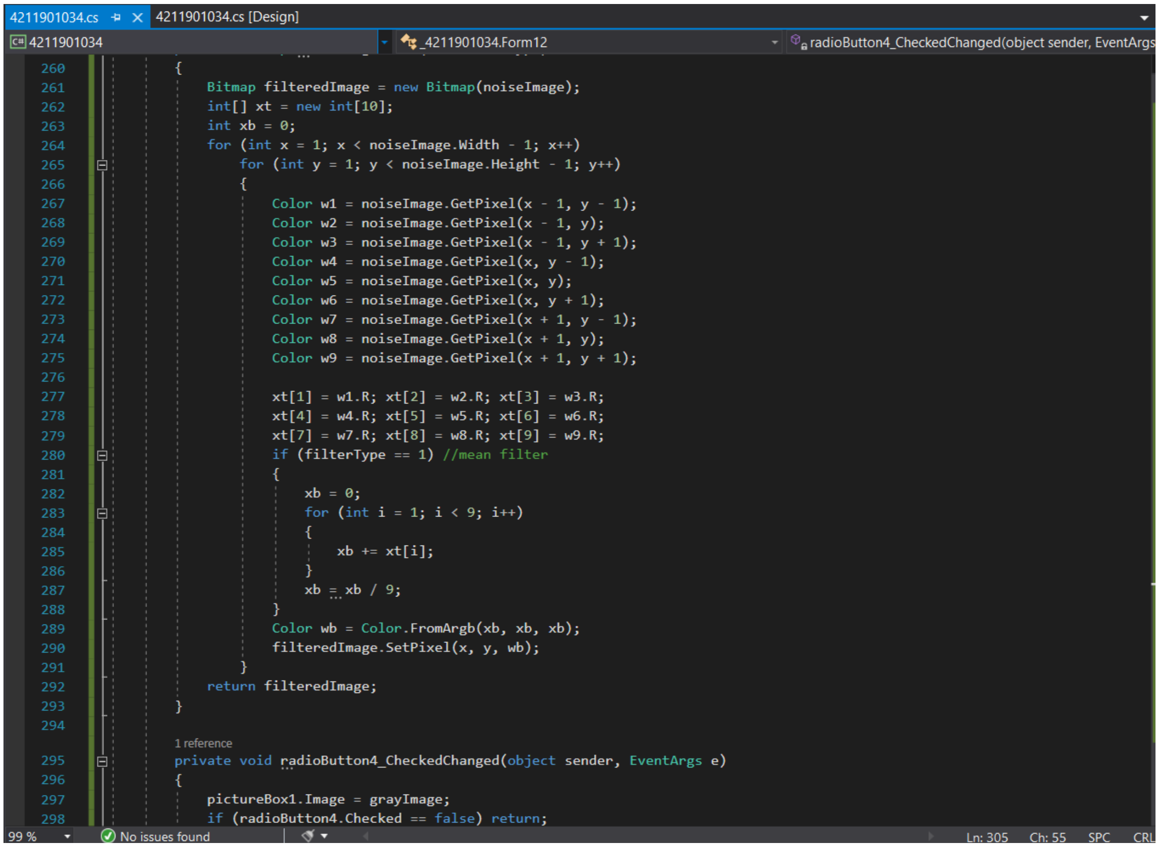
```
                    G *= 255;
                    if (G > 255) G = 255; if (G < 0) G = 0;

                    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;
        }

        1 reference
        private void label1_Click(object sender, EventArgs e)
        {

        }

        1 reference
        private void button5_Click(object sender, EventArgs e)
        {
            resetGrayCondition();
            resetFilterCondition();

        }

        // reset condition
        3 references
        private void resetGrayCondition()
        {
            radioButton1.Checked = false;
            radioButton2.Checked = false;
        }
```



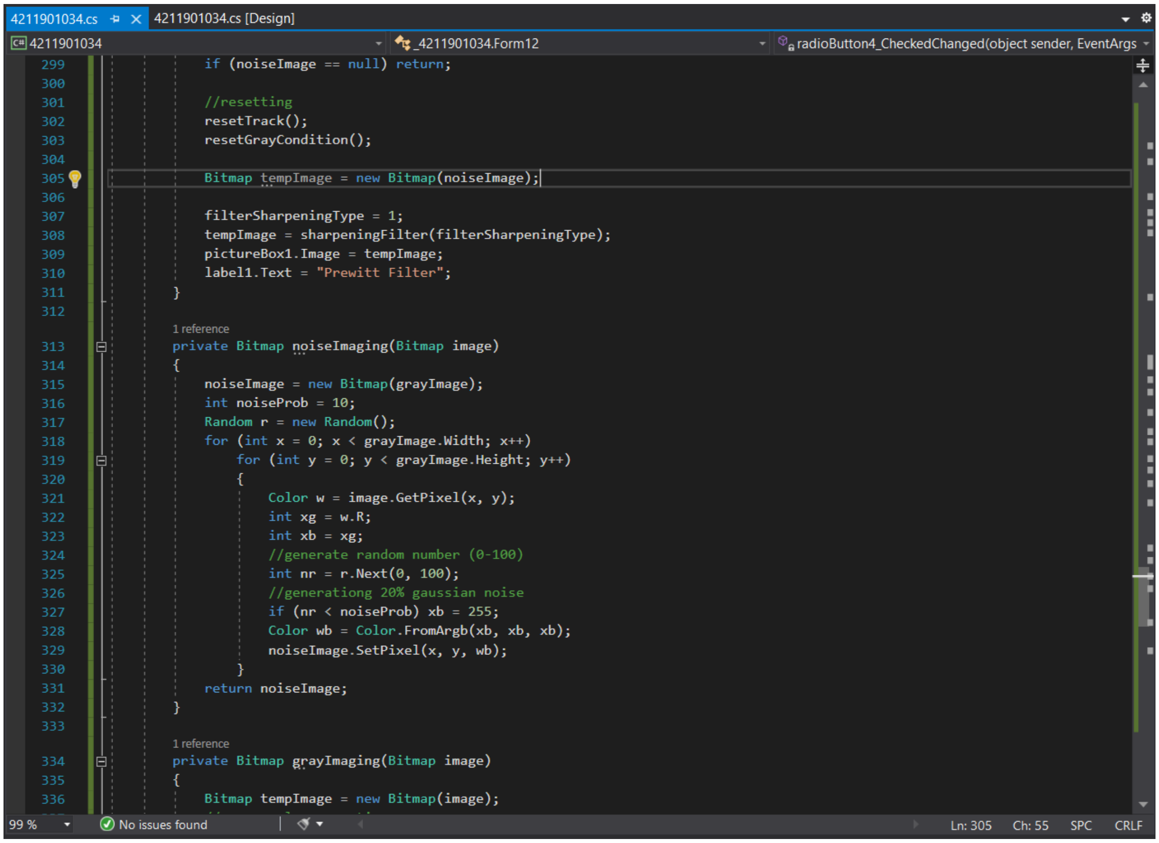
```

        1 reference
        private void button4_Click(object sender, EventArgs e)
        {
            if (processing_image == null) return;
            double contrast = double.Parse(textBox2.Text);
            if (contrast < 0 || contrast > 255) return;
            // setting brightness
            setContrast(contrast);

            // menampilkan nilai pada trackbar
            trackBar2.Value = int.Parse(textBox2.Text);
        }


        1 reference
        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);
        }

        1 reference
        private void resetFilterCondition()
        {
            radioButton3.Checked = false;
            radioButton4.Checked = false;
        }

        1 reference
        private void radioButton3_CheckedChanged(object sender, EventArgs e)
        {
            pictureBox1.Image = grayImage;
            // mean filter
```



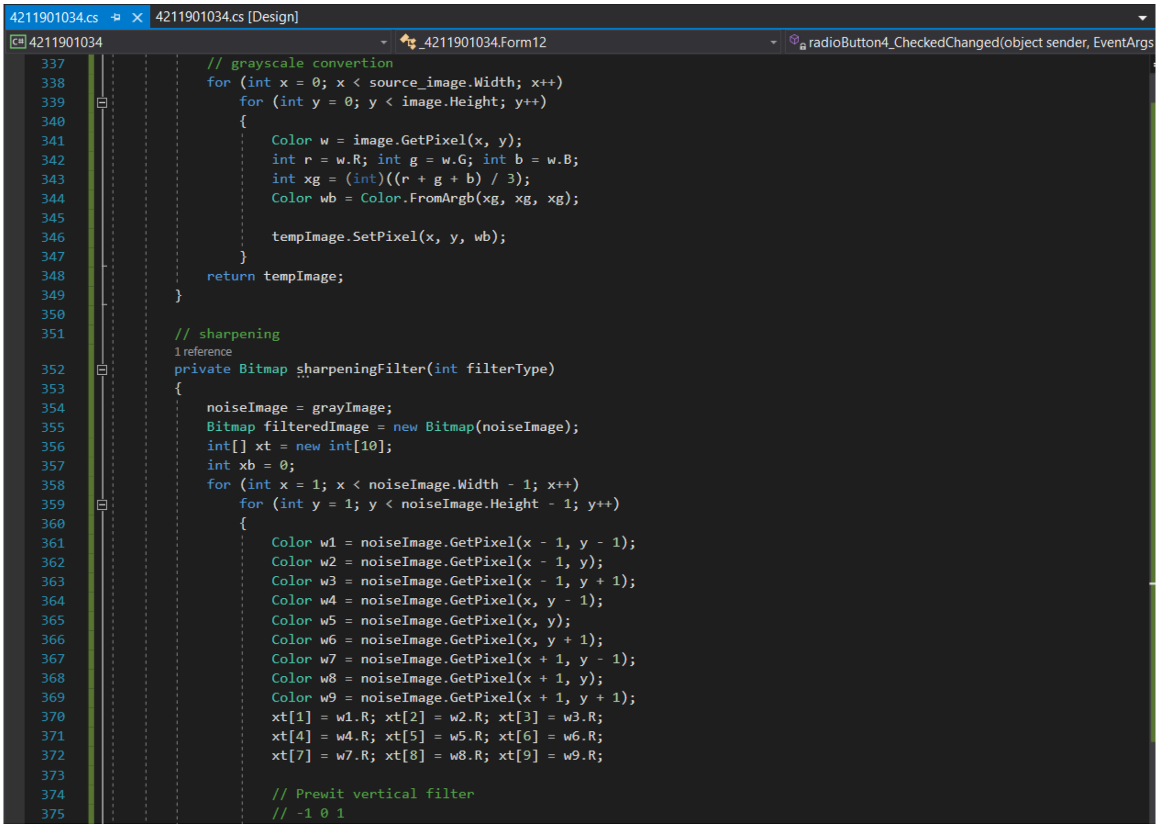
```
            // mean filter
            if (radioButton3.Checked == false) return;

            //resetting
            resetTrack();
            resetGrayCondition();

            if (noiseImage == null) return;
            Bitmap tempImage = new Bitmap(noiseImage);
            filterSmoothingType = 1;
            tempImage = smoothingfilter(filterSmoothingType);
            pictureBox1.Image = tempImage;
            label1.Text = "Mean Filter";
        }

        3 references
        private void resetTrack()
        {
            // trackbar reset condition
            trackBar2.Value = 0;

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

        1 reference
        private void trackbarInitialization()
        {

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

            // init value
            trackBar2.Value = 0;
        }


        1 reference
        private Bitmap smoothingfilter(int filterType)
        {
```



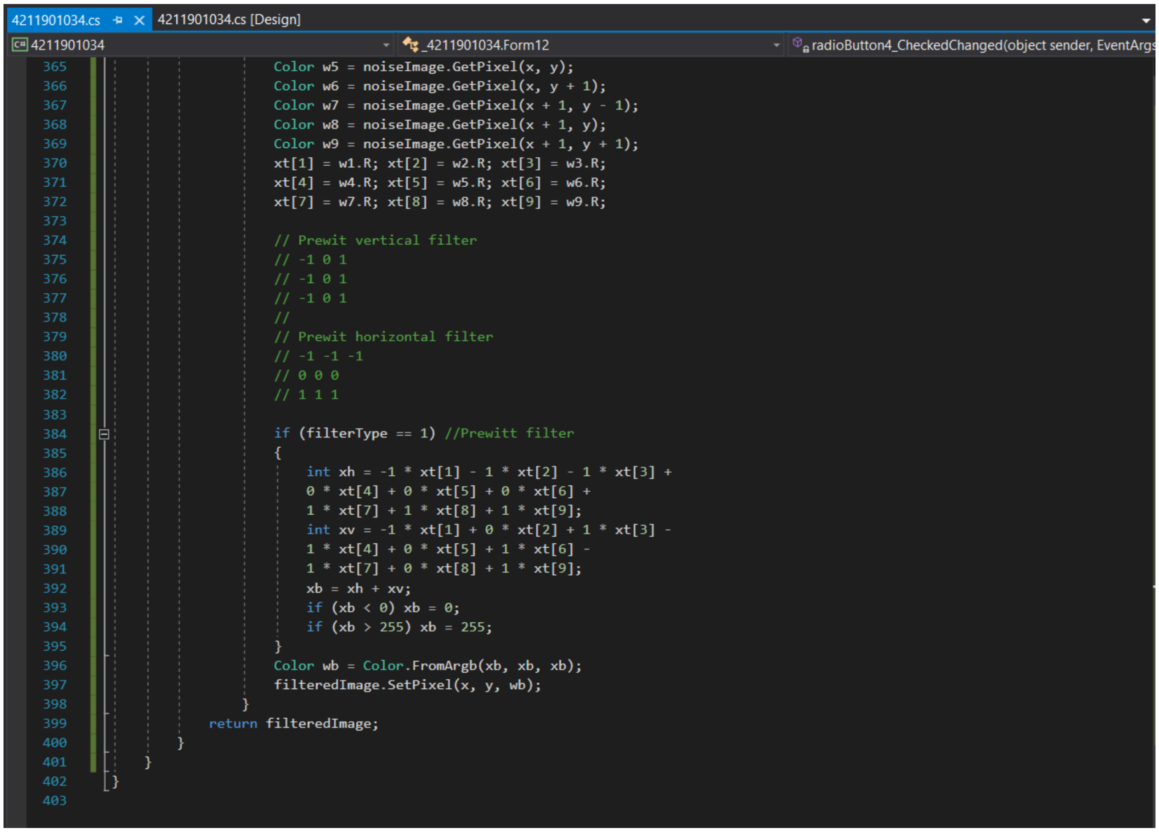
```
        {
            Bitmap filteredImage = new Bitmap(noiseImage);
            int[] xt = new int[10];
            int xb = 0;
            for (int x = 1; x < noiseImage.Width - 1; x++)
                for (int y = 1; y < noiseImage.Height - 1; y++)
                {
                    Color w1 = noiseImage.GetPixel(x - 1, y - 1);
                    Color w2 = noiseImage.GetPixel(x - 1, y);
                    Color w3 = noiseImage.GetPixel(x - 1, y + 1);
                    Color w4 = noiseImage.GetPixel(x, y - 1);
                    Color w5 = noiseImage.GetPixel(x, y);
                    Color w6 = noiseImage.GetPixel(x, y + 1);
                    Color w7 = noiseImage.GetPixel(x + 1, y - 1);
                    Color w8 = noiseImage.GetPixel(x + 1, y);
                    Color w9 = noiseImage.GetPixel(x + 1, y + 1);

                    xt[1] = w1.R; xt[2] = w2.R; xt[3] = w3.R;
                    xt[4] = w4.R; xt[5] = w5.R; xt[6] = w6.R;
                    xt[7] = w7.R; xt[8] = w8.R; xt[9] = w9.R;
                    if (filterType == 1) //mean filter
                    {
                        xb = 0;
                        for (int i = 1; i < 9; i++)
                        {
                            xb += xt[i];
                        }
                        xb = xb / 9;
                    }
                    Color wb = Color.FromArgb(xb, xb, xb);
                    filteredImage.SetPixel(x, y, wb);
                }
            return filteredImage;
        }

        1 reference
        private void radioButton4_CheckedChanged(object sender, EventArgs e)
        {
            pictureBox1.Image = grayImage;
            if (radioButton4.Checked == false) return;
```



```
            if (noiseImage == null) return;

            //resetting
            resetTrack();
            resetGrayCondition();

            Bitmap tempImage = new Bitmap(noiseImage);

            filterSharpeningType = 1;
            tempImage = sharpeningFilter(filterSharpeningType);
            pictureBox1.Image = tempImage;
            label1.Text = "Prewitt Filter";
        }

        1 reference
        private Bitmap noiseImaging(Bitmap image)
        {
            noiseImage = new Bitmap(grayImage);
            int noiseProb = 10;
            Random r = new Random();
            for (int x = 0; x < grayImage.Width; x++)
                for (int y = 0; y < grayImage.Height; y++)
                {
                    Color w = image.GetPixel(x, y);
                    int xg = w.R;
                    int xb = xg;
                    //generate random number (0-100)
                    int nr = r.Next(0, 100);
                    //generationg 20% gaussian noise
                    if (nr < noiseProb) xb = 255;
                    Color wb = Color.FromArgb(xb, xb, xb);
                    noiseImage.SetPixel(x, y, wb);
                }
            return noiseImage;
        }

        1 reference
        private Bitmap grayImaging(Bitmap image)
        {
            Bitmap tempImage = new Bitmap(image);
```

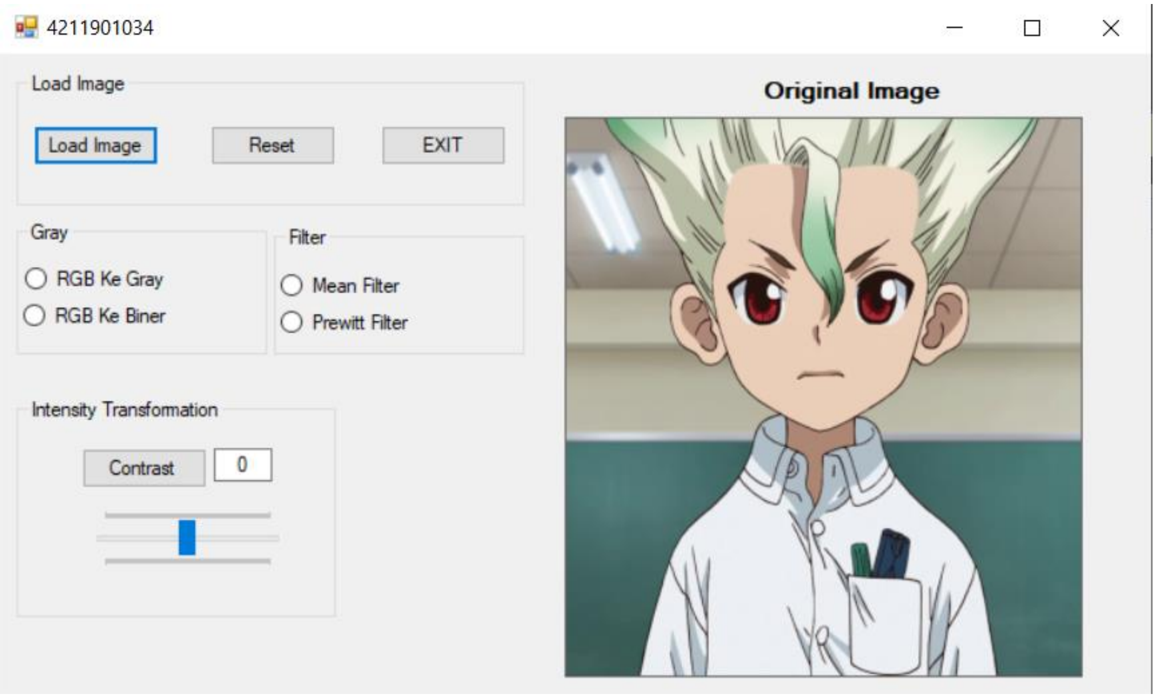


```
            // grayscale convertion
            for (int x = 0; x < source_image.Width; x++)
                for (int y = 0; y < image.Height; y++)
                {
                    Color w = image.GetPixel(x, y);
                    int r = w.R; int g = w.G; int b = w.B;
                    int xg = (int)((r + g + b) / 3);
                    Color wb = Color.FromArgb(xg, xg, xg);

                    tempImage.SetPixel(x, y, wb);
                }
            return tempImage;
        }

        // sharpening
        1 reference
        private Bitmap sharpeningFilter(int filterType)
        {
            noiseImage = grayImage;
            Bitmap filteredImage = new Bitmap(noiseImage);
            int[] xt = new int[10];
            int xb = 0;
            for (int x = 1; x < noiseImage.Width - 1; x++)
                for (int y = 1; y < noiseImage.Height - 1; y++)
                {
                    Color w1 = noiseImage.GetPixel(x - 1, y - 1);
                    Color w2 = noiseImage.GetPixel(x - 1, y);
                    Color w3 = noiseImage.GetPixel(x - 1, y + 1);
                    Color w4 = noiseImage.GetPixel(x, y - 1);
                    Color w5 = noiseImage.GetPixel(x, y);
                    Color w6 = noiseImage.GetPixel(x, y + 1);
                    Color w7 = noiseImage.GetPixel(x + 1, y - 1);
                    Color w8 = noiseImage.GetPixel(x + 1, y);
                    Color w9 = noiseImage.GetPixel(x + 1, y + 1);
                    xt[1] = w1.R; xt[2] = w2.R; xt[3] = w3.R;
                    xt[4] = w4.R; xt[5] = w5.R; xt[6] = w6.R;
                    xt[7] = w7.R; xt[8] = w8.R; xt[9] = w9.R;

                    // Prewit vertical filter
                    // -1 0 1
```

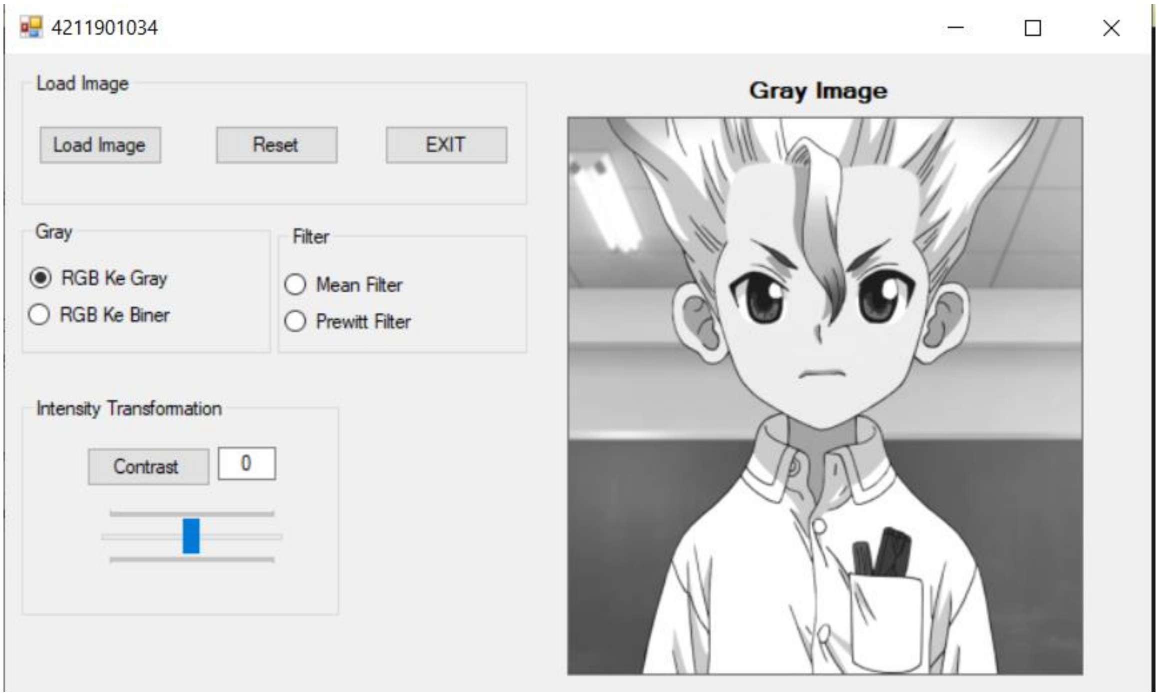


```
                    Color w5 = noiseImage.GetPixel(x, y);
                    Color w6 = noiseImage.GetPixel(x, y + 1);
                    Color w7 = noiseImage.GetPixel(x + 1, y - 1);
                    Color w8 = noiseImage.GetPixel(x + 1, y);
                    Color w9 = noiseImage.GetPixel(x + 1, y + 1);
                    xt[1] = w1.R; xt[2] = w2.R; xt[3] = w3.R;
                    xt[4] = w4.R; xt[5] = w5.R; xt[6] = w6.R;
                    xt[7] = w7.R; xt[8] = w8.R; xt[9] = w9.R;

                    // Prewit vertical filter
                    // -1 0 1
                    // -1 0 1
                    // -1 0 1
                    //
                    // Prewit horizontal filter
                    // -1 -1 -1
                    // 0 0 0
                    // 1 1 1

                    if (filterType == 1) //Prewitt filter
                    {
                        int xh = -1 * xt[1] - 1 * xt[2] - 1 * xt[3] +
                        0 * xt[4] + 0 * xt[5] + 0 * xt[6] +
                        1 * xt[7] + 1 * xt[8] + 1 * xt[9];
                        int xv = -1 * xt[1] + 0 * xt[2] + 1 * xt[3] -
                        1 * xt[4] + 0 * xt[5] + 1 * xt[6] -
                        1 * xt[7] + 0 * xt[8] + 1 * xt[9];
                        xb = xh + xv;
                        if (xb < 0) xb = 0;
                        if (xb > 255) xb = 255;
                    }
                    Color wb = Color.FromArgb(xb, xb, xb);
                    filteredImage.SetPixel(x, y, wb);
                }
            return filteredImage;
        }
    }
}
```

Screenshoot of Image Processing of Each Operation



Load Original Image

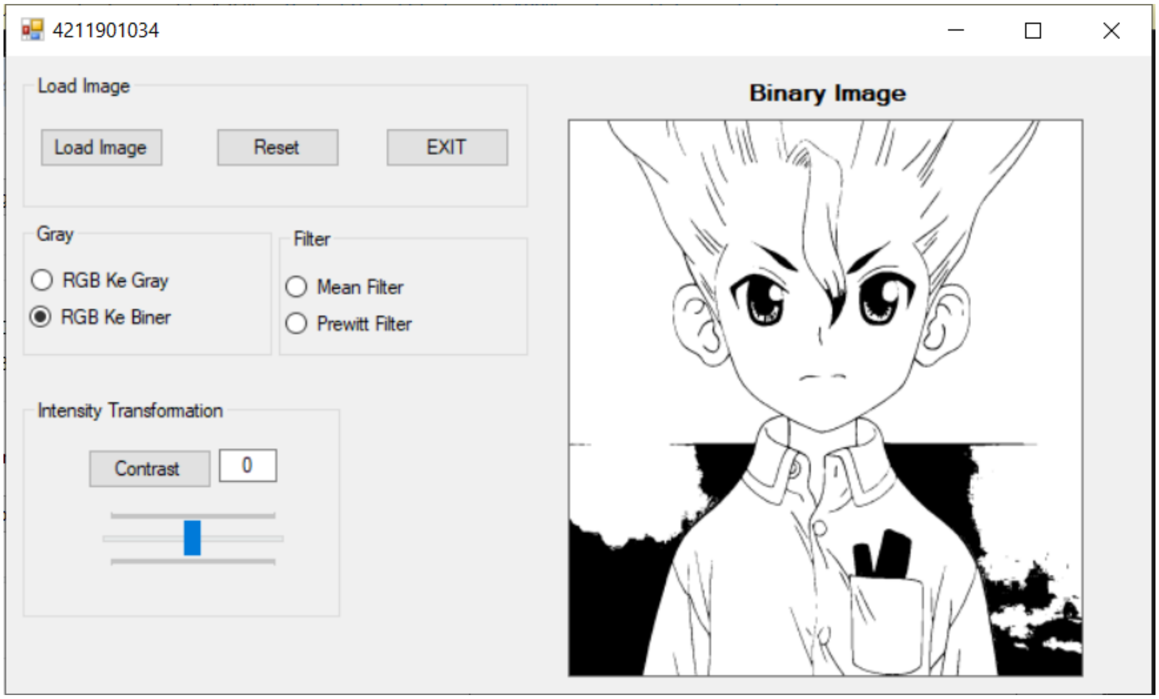


● RGB Ke Gray
○ RGB Ke Biner

○ Mean Filter
○ Prewitt Filter

Contrast: 0

Konversi RGB ke Gray

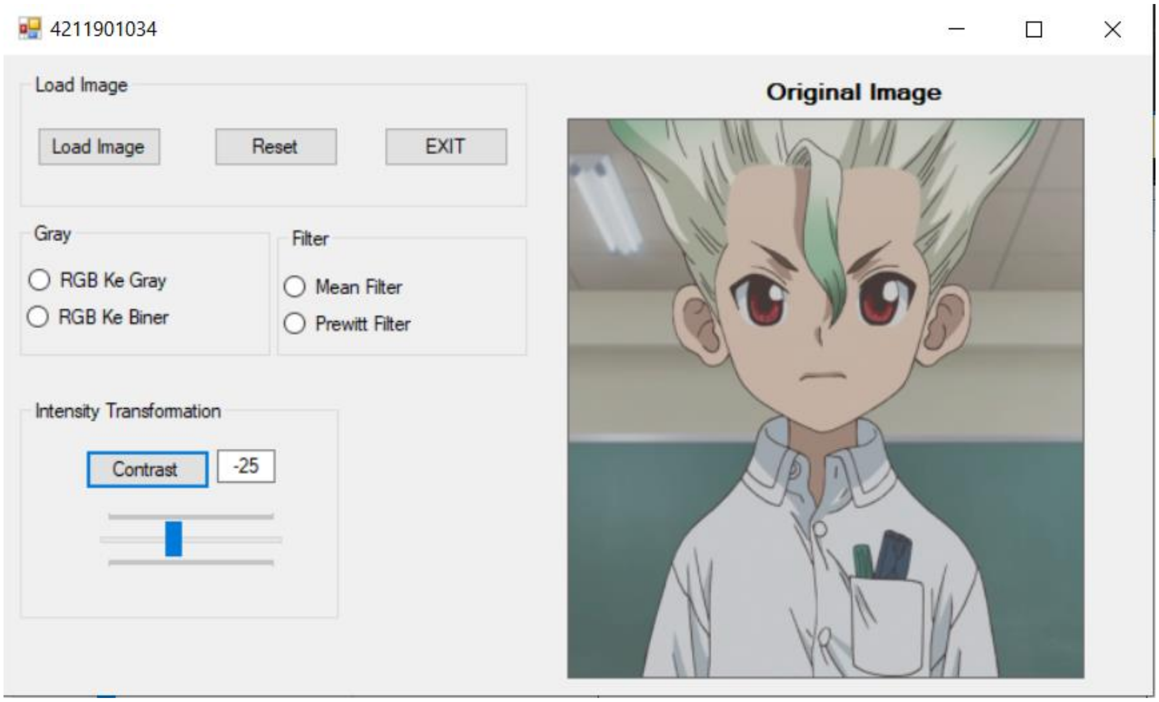


○ RGB Ke Gray
● RGB Ke Biner

○ Mean Filter
○ Prewitt Filter

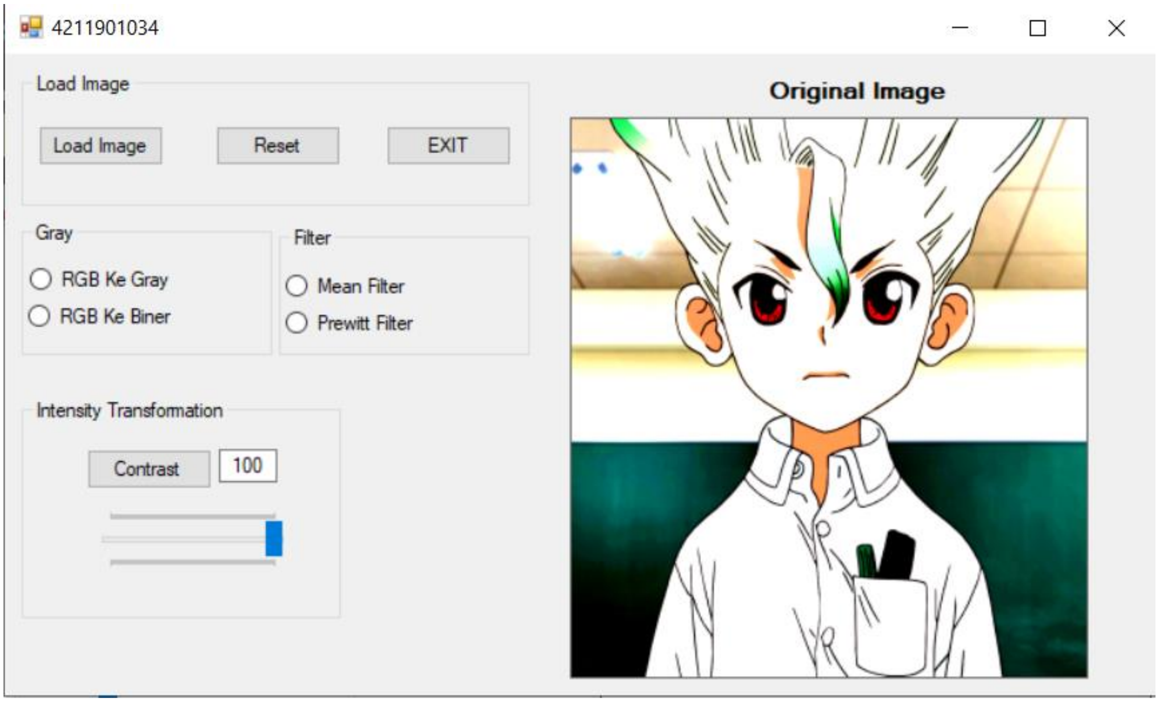
Contrast: 0

Konversi RGB ke Biner



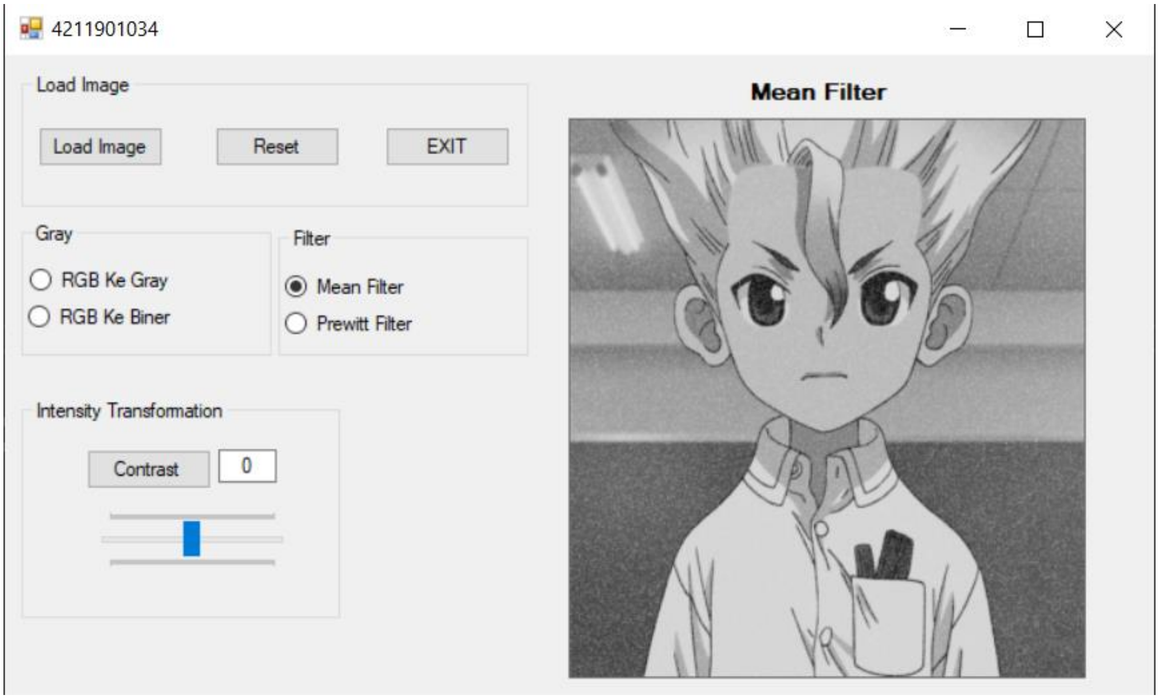
Contrast -25

Setting Contrast dengan button



Contrast 100

Setting Contrast max value dengan trackbar



Gray

○ RGB Ke Gray

○ RGB Ke Biner

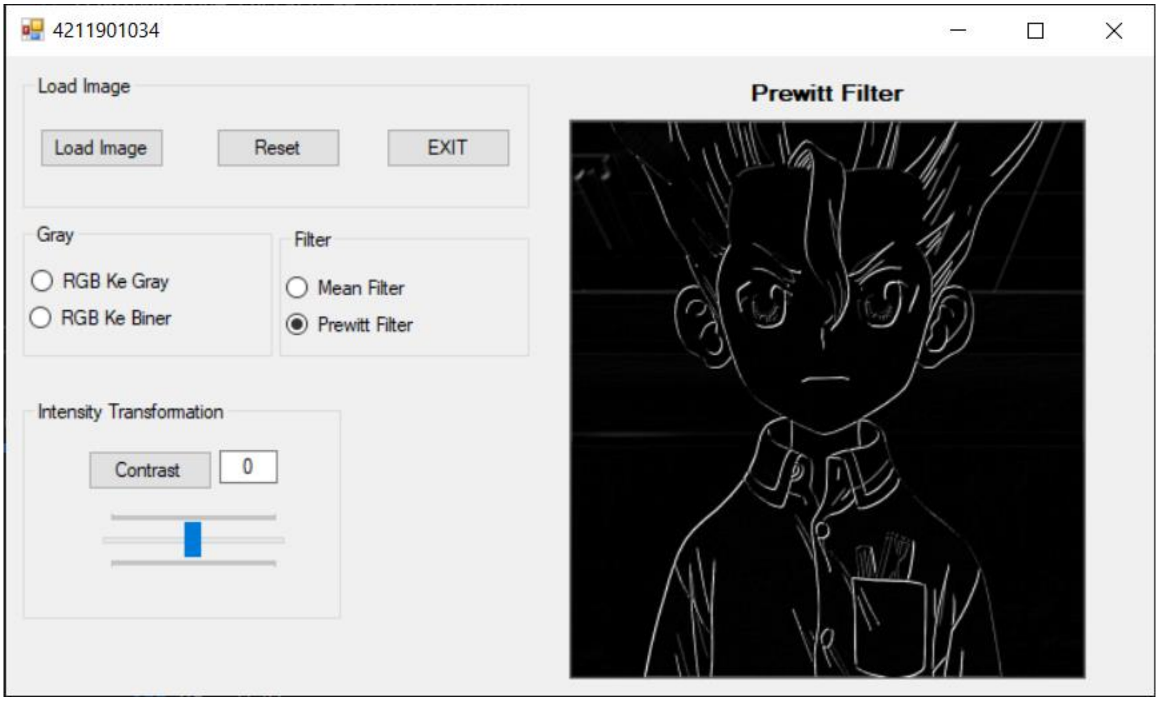
Filter

● Mean Filter

○ Prewitt Filter

Contrast: 0

Mean Filter



Gray

○ RGB Ke Gray

○ RGB Ke Biner

Filter

○ Mean Filter

● Prewitt Filter

Contrast: 0

Prewitt Filter