TUGAS PERCOBAAN 5

PENGOLAHAN CITRA

MK401
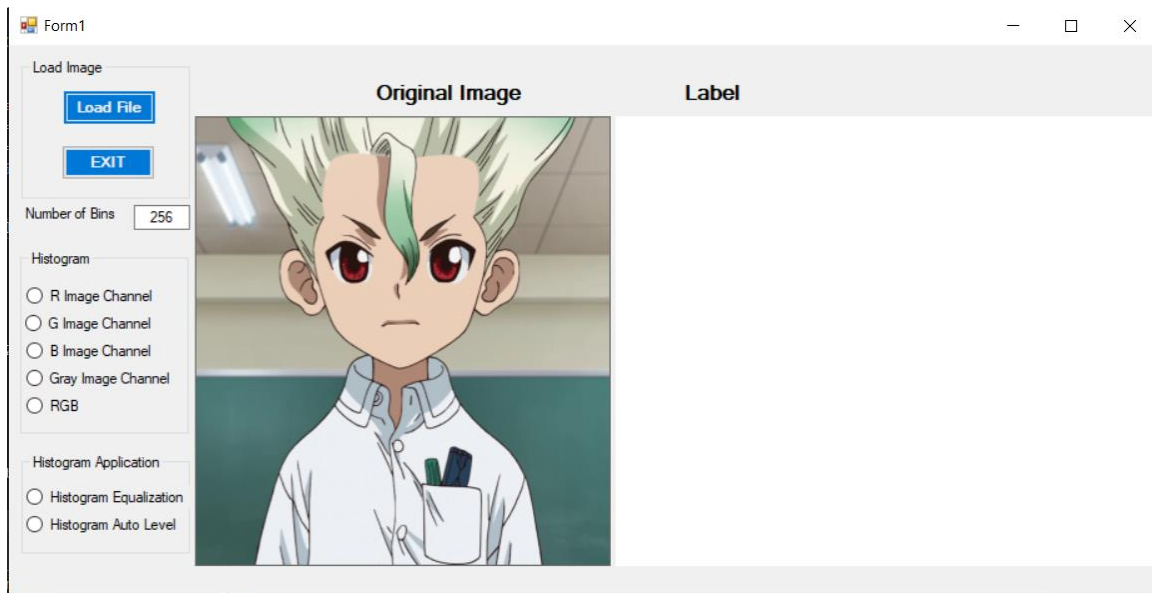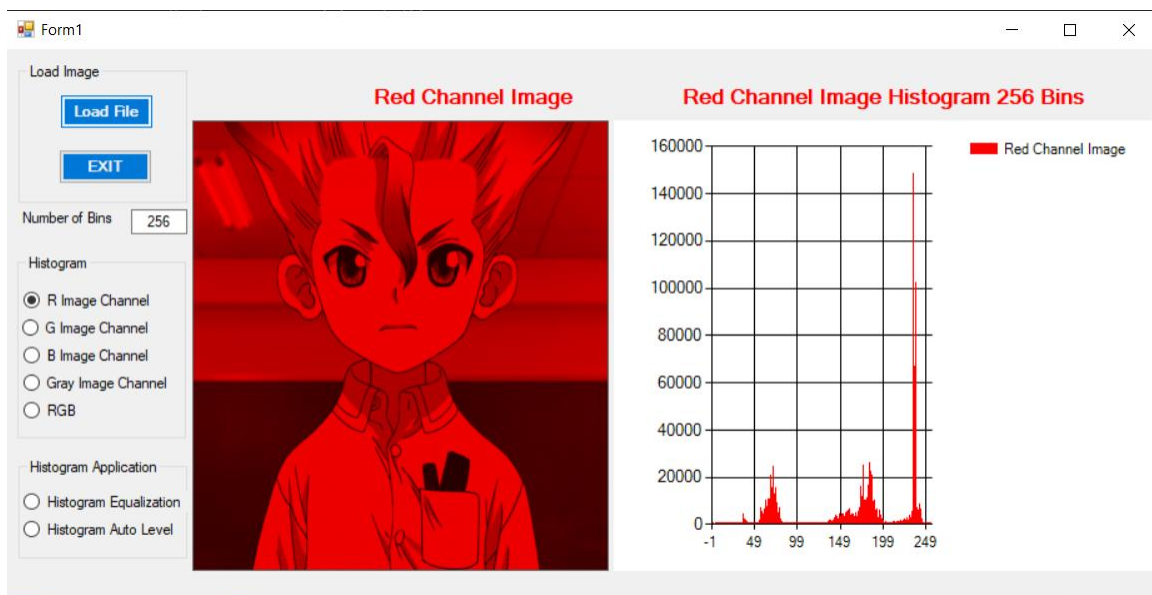
Disusun oleh :

Ricky Silitonga (4211901034)

PROGRAM STUDI TEKNIK MEKATRONIKA

JURUSAN TEKNIK ELEKTRO

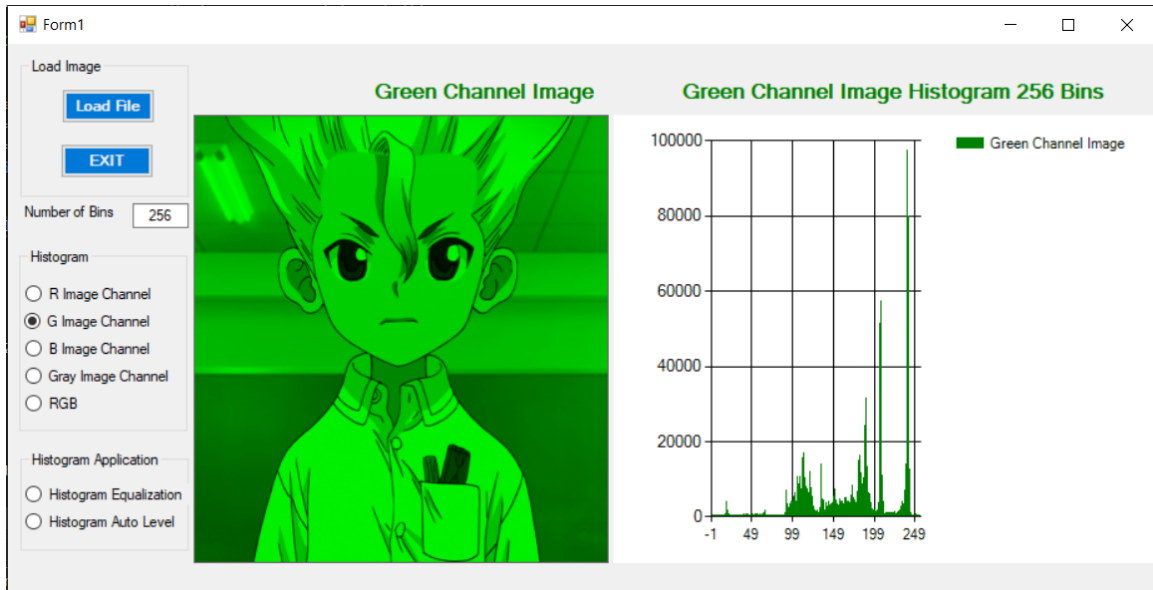**POLITEKNIK NEGERI BATAM**

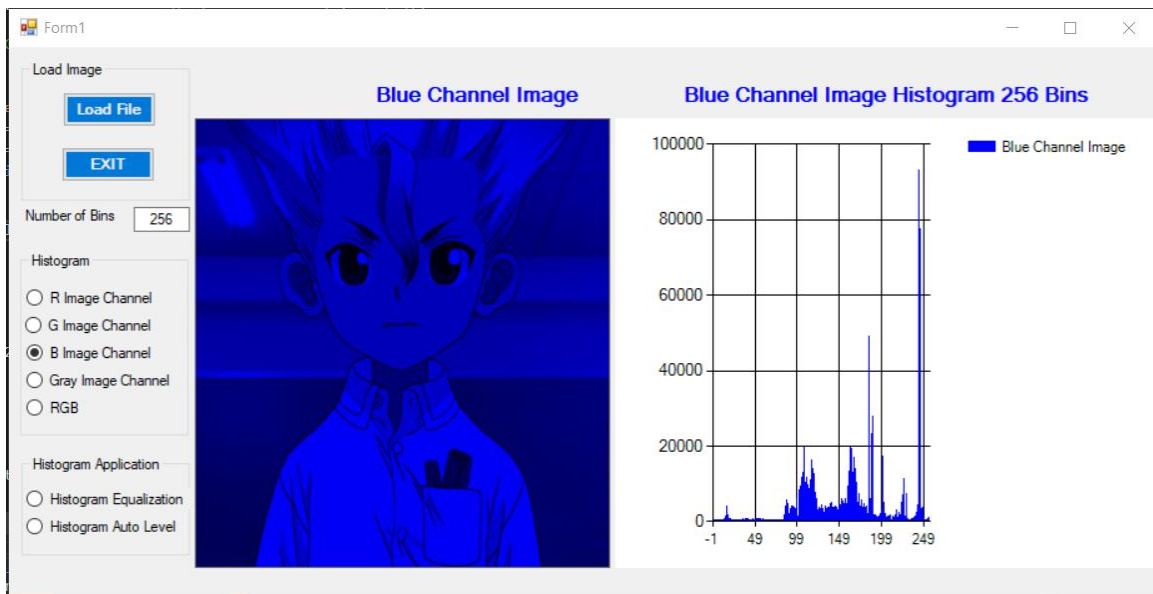**2020**

Histogram dan Aplikasinya

Load Image dengan gambar original

Red Image Channel 256 BIN

Green Image Channel 256 BIN

Blue Image Channel 256 BIN
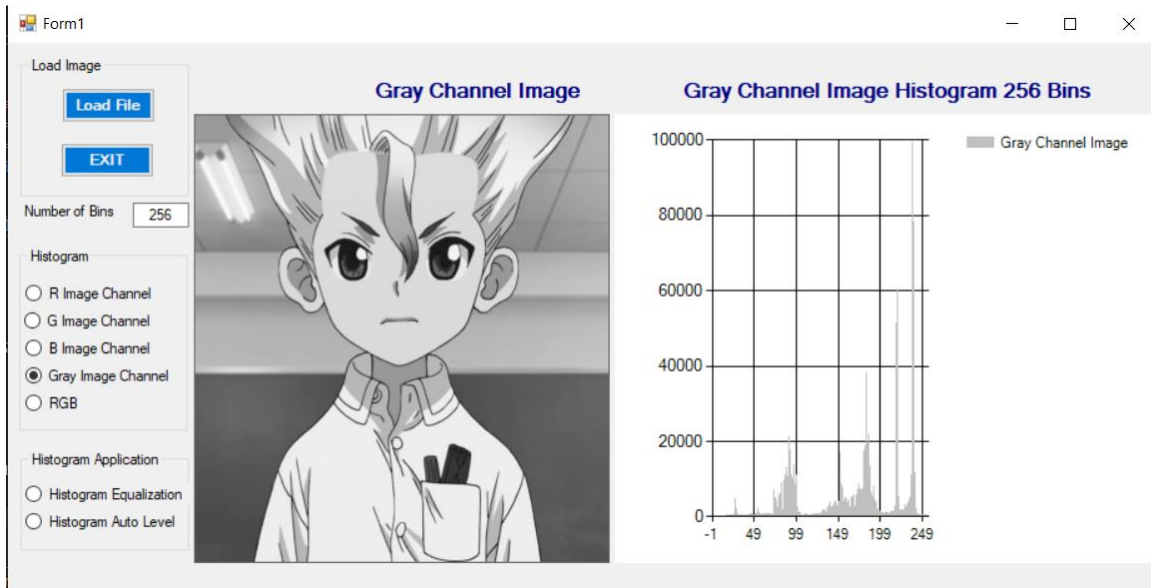
Gray Channel Image 256 BIN
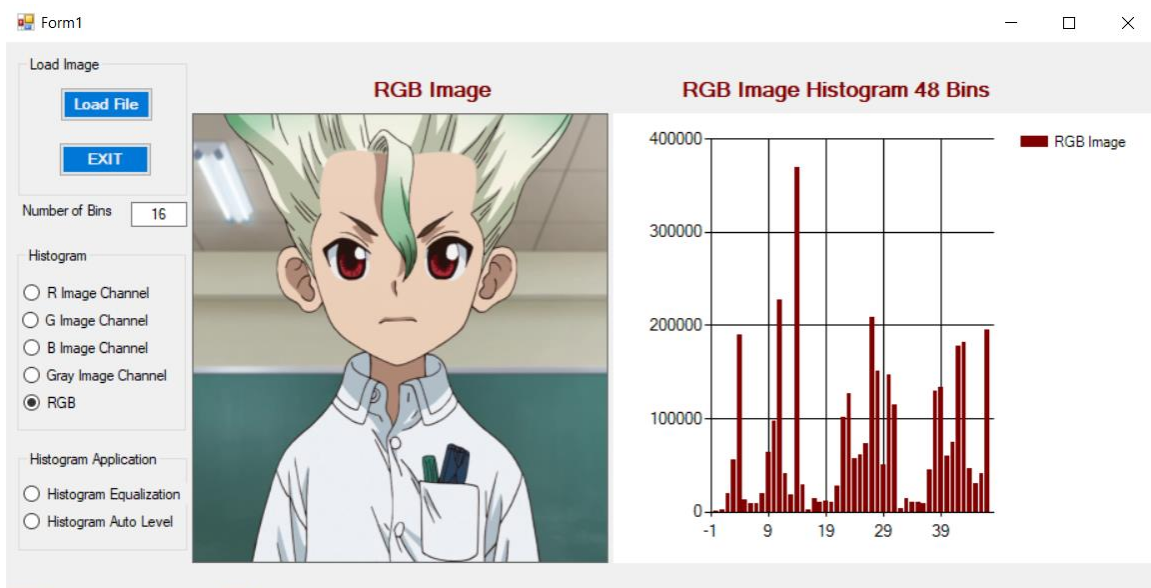
RGB Image 16 * 3 = 48 BIN
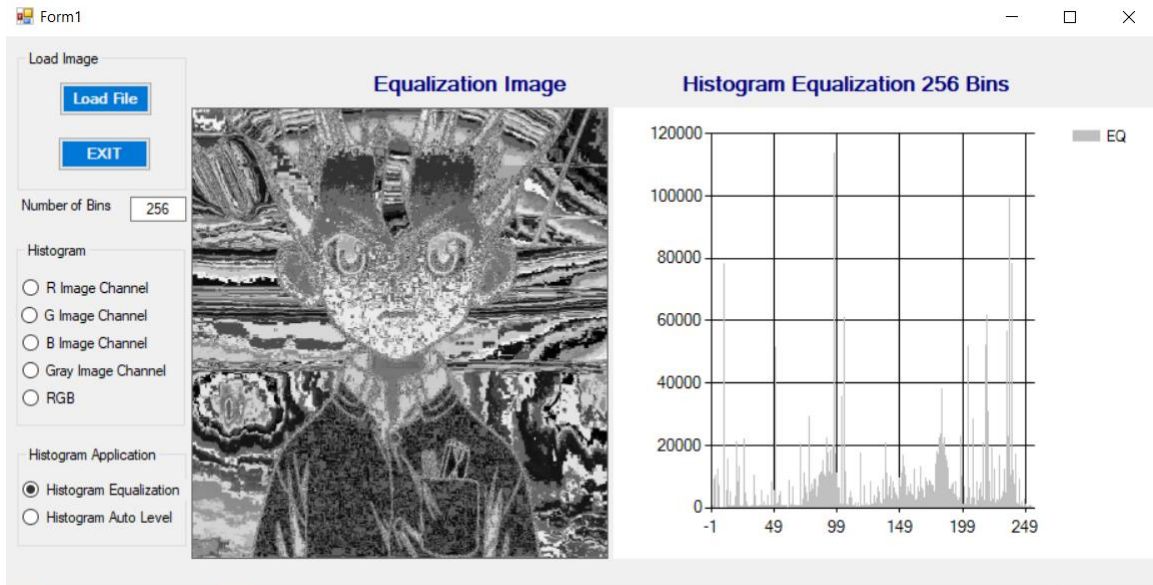
Gambar dan Histogram setelah equalization
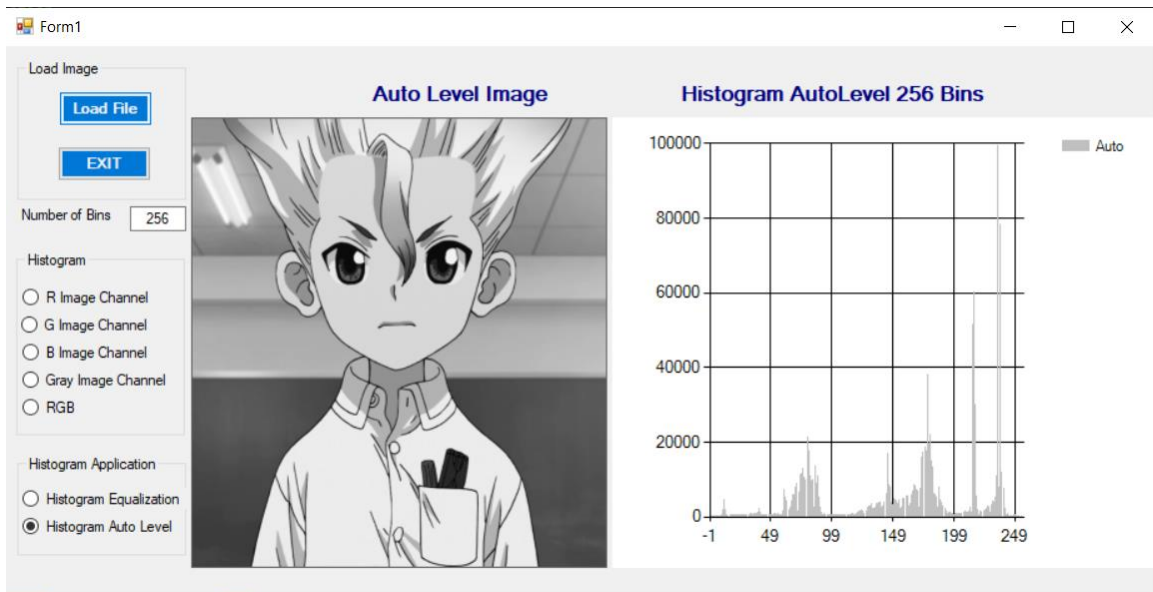
Gambar dan Histogram setelah autolevel
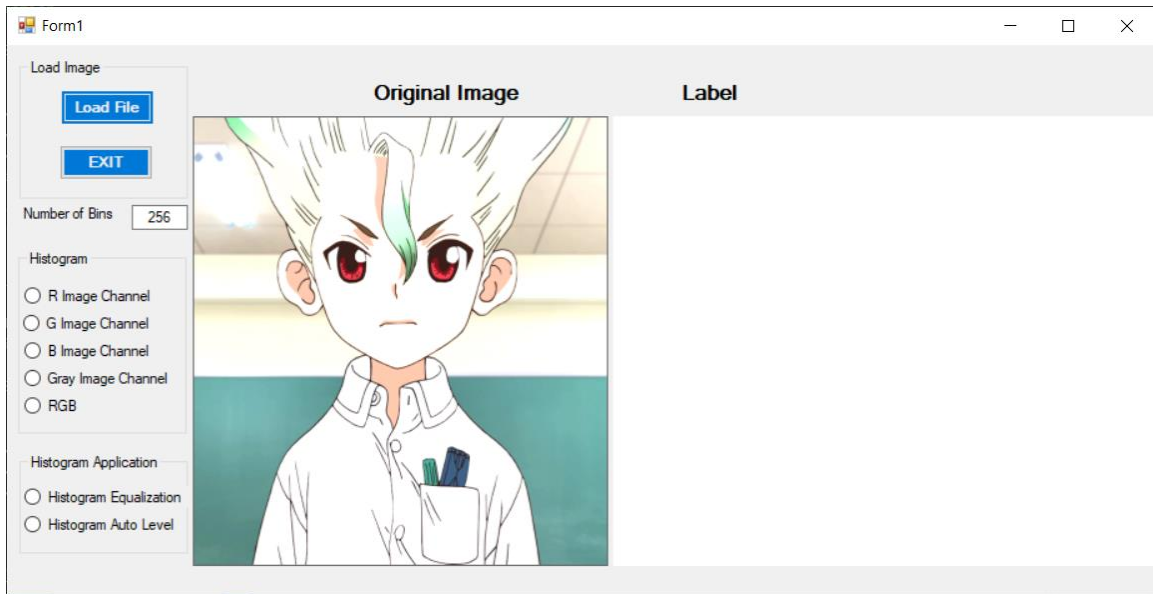
Load high brightness image
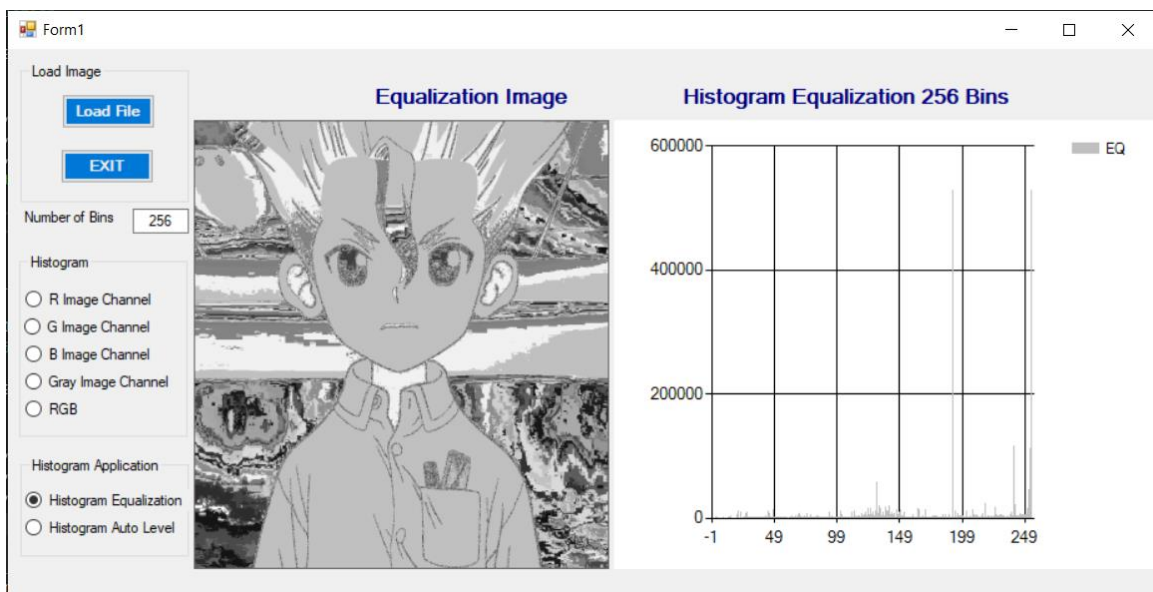
Gambar dan Histogram image high brightness setelah equalization
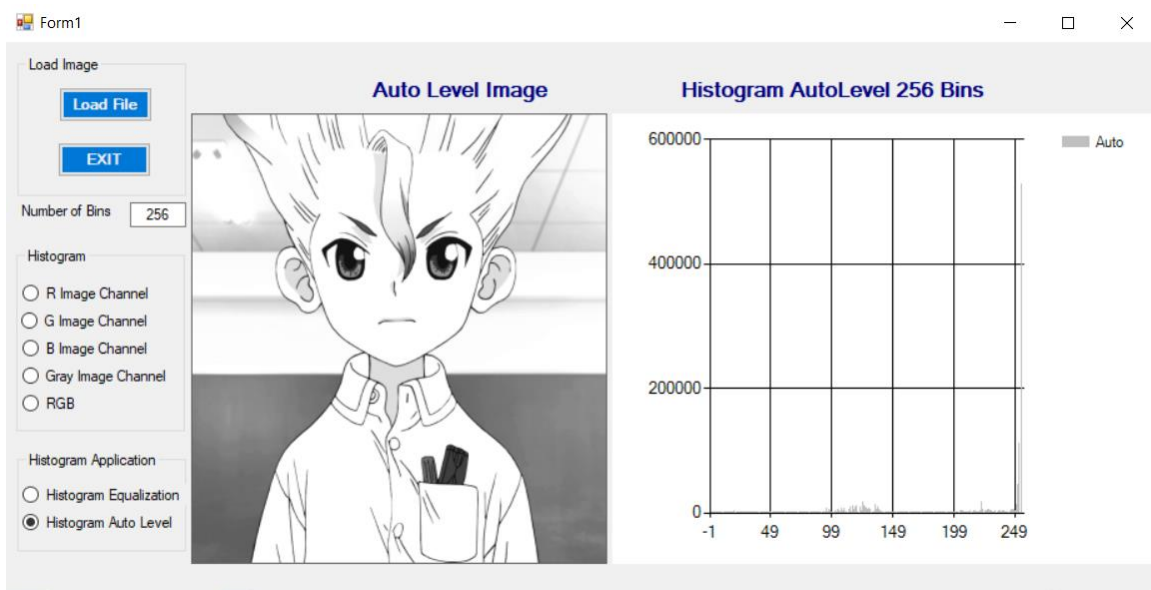
Gambar dan Histogram image high brightness setelah auto level
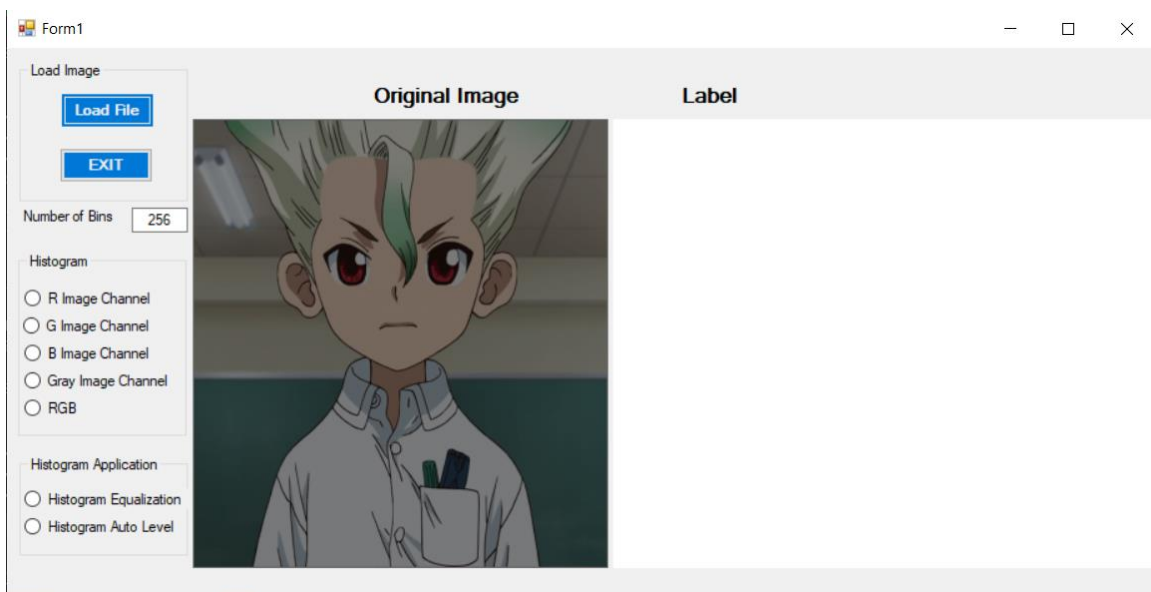
Load low brightness image
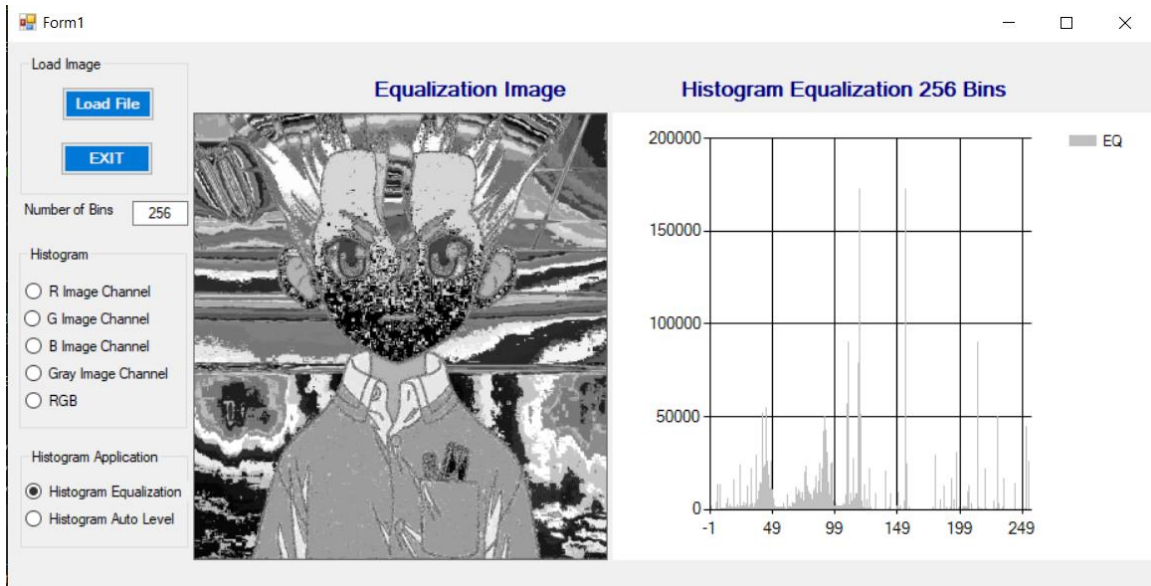
Gambar dan Histogram image low brightness setelah equalization
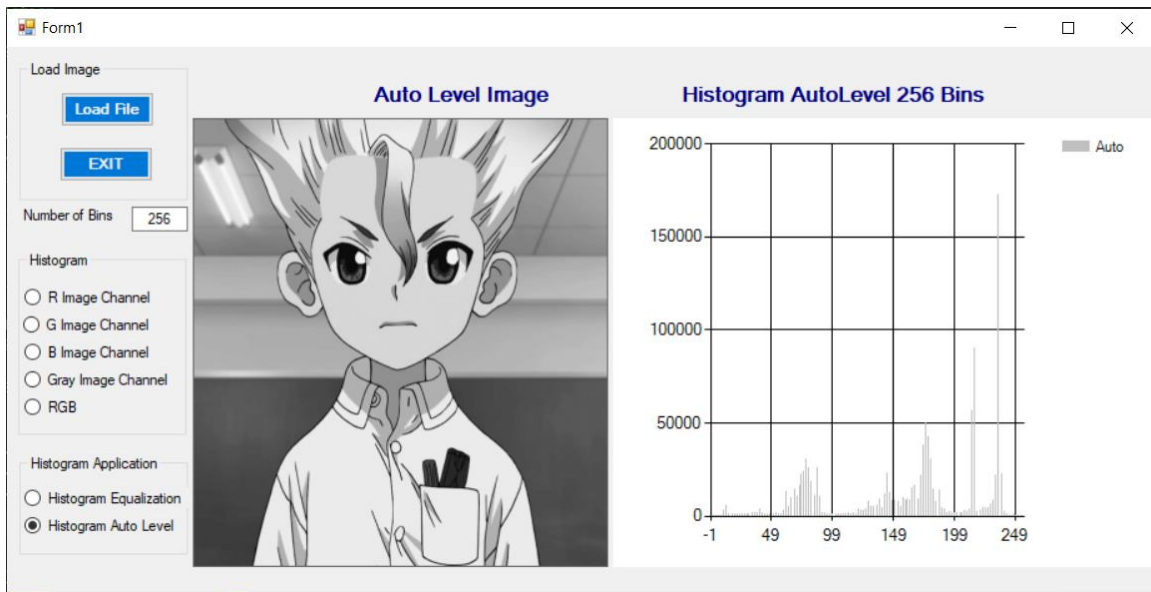
Gambar dan Histogram image low brightness setelah auto level

Sourcecode Button G, B, Gray dan Fungsi Convert

```csharp
private void radioButton2_CheckedChanged(object sender, EventArgs e)
    {
        // g image channel
        if (sourceImage == null) return;
        if (radioButton2.Checked == false) return;
        radioButtonHisAppClear();
        int pilChannel = 2;
        //delete the histogram
        if (chart1.Series.Count > 0)
        {
            chart1.Series.RemoveAt(0);
        }
        //chart init
        chart1.Series.Add("Green Channel Image");
        chart1.Series["Green Channel Image"].Color = Color.Green;
        foreach (var series in chart1.Series)
        {
            series.Points.Clear();
        }
        float[] his = new float[BIN];
        his = hitungHistogram(pilChannel);
        for (int i = 0; i < BIN; i++)
        {
            chart1.Series["Green Channel Image"].Points.AddXY(i, his[i]);
        }
        //displaying Green Channel
        Bitmap greenImage = imageConvert(pilChannel);
        pictureBox1.Image = greenImage;
        label2.Text = "Green Channel Image";
        label2.ForeColor = Color.Green;
        label3.ForeColor = Color.Green;
        label3.Text = string.Format("Green Channel Image Histogram {0} Bins",
BIN);
    }


private void radioButton3_CheckedChanged(object sender, EventArgs e)
    {
        // b image channel
        if (sourceImage == null) return;
        if (radioButton3.Checked == false) return;
        radioButtonHisAppClear();
        int pilChannel = 3;
        //delete the histogram
        if (chart1.Series.Count > 0)
        {
            chart1.Series.RemoveAt(0);
        }
        //chart init
        chart1.Series.Add("Blue Channel Image");
        chart1.Series["Blue Channel Image"].Color = Color.Blue;
        foreach (var series in chart1.Series)
        {
            series.Points.Clear();
        }
```

```csharp
            float[] his = new float[BIN];
            his = hitungHistogram(pilChannel);
            for (int i = 0; i < BIN; i++)
            {
                chart1.Series["Blue Channel Image"].Points.AddXY(i, his[i]);
            }
            //displaying Blue Channel
            Bitmap blueImage = imageConvert(pilChannel);
            pictureBox1.Image = blueImage;
            label2.Text = "Blue Channel Image";
            label2.ForeColor = Color.Blue;
            label3.ForeColor = Color.Blue;
            label3.Text = string.Format("Blue Channel Image Histogram {0} Bins", BIN);
        }

        private void radioButton4_CheckedChanged(object sender, EventArgs e)
        {
            // gray image channel
            if (sourceImage == null) return;
            if (radioButton4.Checked == false) return;
            radioButtonHisAppClear();
            int pilChannel = 4;
            //delete the histogram
            if (chart1.Series.Count > 0)
            {
                chart1.Series.RemoveAt(0);
            }
            //chart init
            chart1.Series.Add("Gray Channel Image");
            chart1.Series["Gray Channel Image"].Color = Color.Silver;
            foreach (var series in chart1.Series)
            {
                series.Points.Clear();
            }
            float[] his = new float[BIN];
            his = hitungHistogram(pilChannel);
            for (int i = 0; i < BIN; i++)
            {
                chart1.Series["Gray Channel Image"].Points.AddXY(i, his[i]);
            }
            //displaying gray Channel
            Bitmap grayImage = imageConvert(pilChannel);
            pictureBox1.Image = grayImage;
            label2.Text = "Gray Channel Image";
            label2.ForeColor = Color.Navy;
            label3.ForeColor = Color.Navy;
            label3.Text = string.Format("Gray Channel Image Histogram {0} Bins", BIN);
        }


        private Bitmap imageConvert(int imageChannel)
        {
            if (sourceImage == null) return null;
            Bitmap convImage = new Bitmap(sourceImage);
            for (int x = 0; x < sourceImage.Width; x++)
                for (int y = 0; y < sourceImage.Height; y++)
                {
```

```csharp
            //get the RGB value of the pixel at (x,y)
            Color w = sourceImage.GetPixel(x, y);
            byte r = w.R; //red value
            byte g = w.G; // green value
            byte b = w.B; // blue value
                        //calculate gray channel
            byte gray = (byte)(0.5 * r + 0.419 * g + 0.081 * b);
            //set the color of each channel
            //red channel image
            Color redColor = Color.FromArgb(r, 0, 0);
            Color greenColor = Color.FromArgb(0, g, 0);
            Color blueColor = Color.FromArgb(0, 0, b);
            Color grayColor = Color.FromArgb(gray, gray,gray );
            //for green, blue and gray channel image,
            //please add yourself the coding for them
            // tambah coding sendiri
            //set the image pixel
            if (imageChannel == 1) //red
            {
                convImage.SetPixel(x, y, redColor);
            }
            else if (imageChannel == 2) //green
            {
                // tambah coding sendiri
                convImage.SetPixel(x, y, greenColor);
            }
            else if (imageChannel == 3) //blue
            {
                // tambah coding sendiri
                convImage.SetPixel(x, y, blueColor);
            }
            else if (imageChannel == 4) //gray
            {
                // tambah coding sendiri
                convImage.SetPixel(x, y, grayColor);
            }
        }
    return convImage;
}
```