

5

HISTOGRAM DAN APLIKASINYA

Pada pokok bahasan ini, mahasiswa akan mempelajari tentang cara menghitung dan menampilkan histogram sebuah citra dan aplikasi histogram pada pengolahan citra.

Pokok Bahasan :

1. Histogram
2. Aplikasi histogram

Latihan :

1. Membuat aplikasi untuk menghitung histogram dari citra menampilkannya
2. Mengaplikasikan histogram untuk memperbaiki kualitas citra

5.1 Histogram

Capaian pembelajaran : mampu memahami proses penghitungan histogram suatu citra dan menampilkannya

Histogram suatu citra/gambar RGB menunjukkan distribusi nilai-nilai pixel dari citra atau gambar tersebut. Histogram $H(xg)$ ini menyatakan jumlah titik yang mempunyai nilai pixel xg . Untuk menghasilkan histogram xg ini atau nilai derajat keabuan xg dari setiap titik (x,y) . Tambahkan $H(xg)$ secara counter $H(xg)=H(xg)+1$.

Karena histogram menyajikan distribusi setiap nilai derajat keabuan, maka dapat dengan histogram ini kita bisa melihat distribusi terang-gelapnya sebuah gambar. Gambar yang histogramnya cenderung ke kiri adalah gambar yang gelap, sedangkan gambar yang histogramnya cenderung ke kanan adalah gambar yang terang.

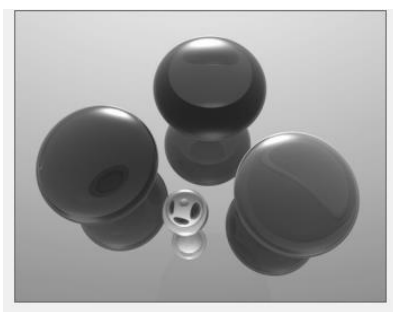
Fungsi Distribusi Kumulatif atau Cumulative Density Function (CDF) $C(xg)$ dari sebuah gambar derajat keabuan menunjukkan jumlah titik yang nilai derajat keabuannya mulai nol sampai xg . Fungsi ini menyatakan model kurva histogram yang selalu naik. Fungsi distribusi kumulatif $C(xg)$ dapat dirumuskan dengan:

$$C(xg) = \sum_{i=0}^{xg} H(i)$$

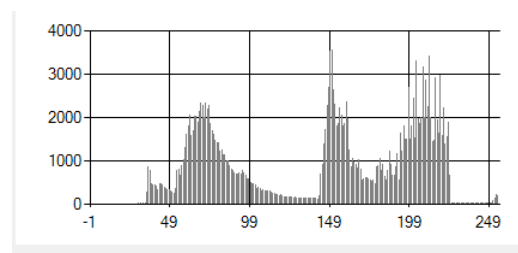
Atau disederhanakan dengan fungsi counter sbb:

$$C(0) = H(0)$$

$$C(xg) = C(xg - 1) + H(xg)$$



a. Citra Gray



b. Histogram

Gambar 5.1 Citra gray dan histogramnya

5.2 Aplikasi histogram

Capaian pembelajaran : mampu mengaplikasikan suatu histogram pada suatu citra.

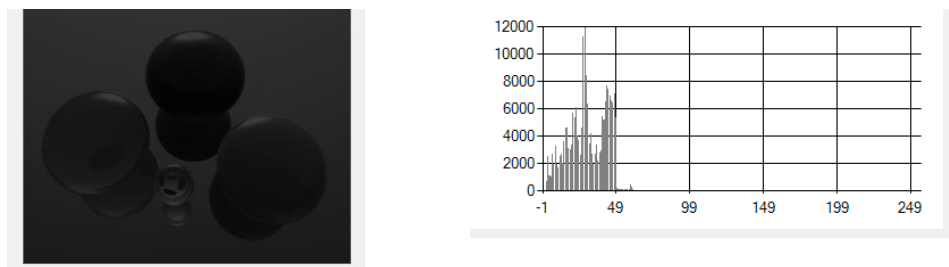
A. Histogram Ekualisasi

Histogram ekualisasi adalah sebuah proses untuk memperbaiki terang/gelap citra dengan memanfaatkan distribusi nilai-nilai pixel dari citra tersebut. Untuk dapat melakukan histogram equalization ini diperlukan suatu fungsi distribusi kumulatif yang merupakan kumulatif dari histogram. Histogram ekualisasi dapat dihitung dengan proses berikut :

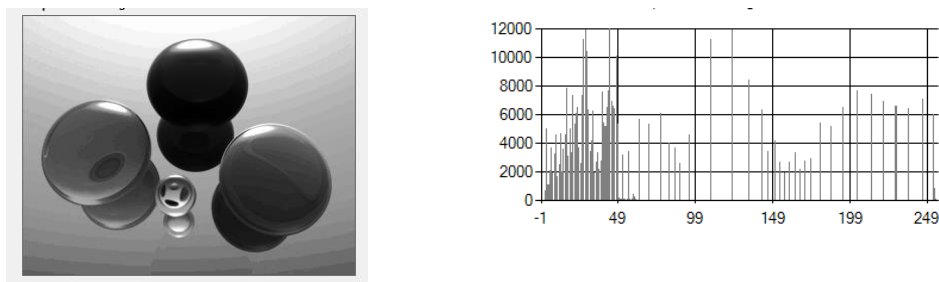
1. Hitung histogram $h(x)$
2. Hitung histogram cumulative $C(x)$
3. Nilai pixel yang baru didefinisikan dengan

$$x' = \frac{255C(x)}{n_x - n_y}$$

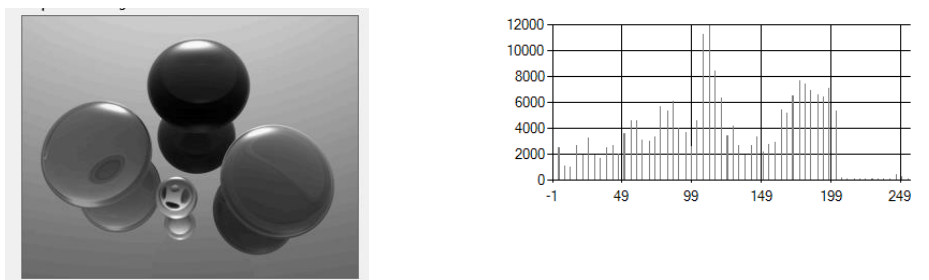
Dengan n_x : lebar gambar dan n_y : tinggi citra



a. Citra asal dan histogramnya



b. Citra hasil ekualisasi dan histogramnya



c. Citra hasil auto level dan histogramnya

Gambar 5.2 Aplikasi histogram

B. Histogram Auto Level

Auto-Level adalah proses untuk membuat semua nilai derajat keabuan mulai 0 sampai dengan 255 terisi. Pada gambar yang terlalu gelap, biasanya nilai maksimumnya tidak mencapai 255. Pada gambar yang terlalu terang, biasanya nilai minimumnya tidak di nol. Dengan proses auto level ini, maka nilai maksimumnya dibuat 255 dan nilai minimumnya dibuat nol sehingga membuat gambar bisa semakin jelas.

Proses auto level menggunakan rumus sebagai berikut:

$$x' = \frac{255}{x_{max} - x_{min}} (x - x_{min})$$

5.3 Latihan

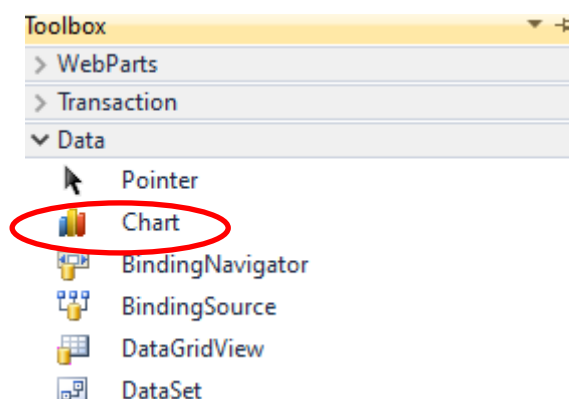
Tujuan

Memahami cara menghitung histogram suatu citra dan mengaplikasikannya untuk mengolah citra.

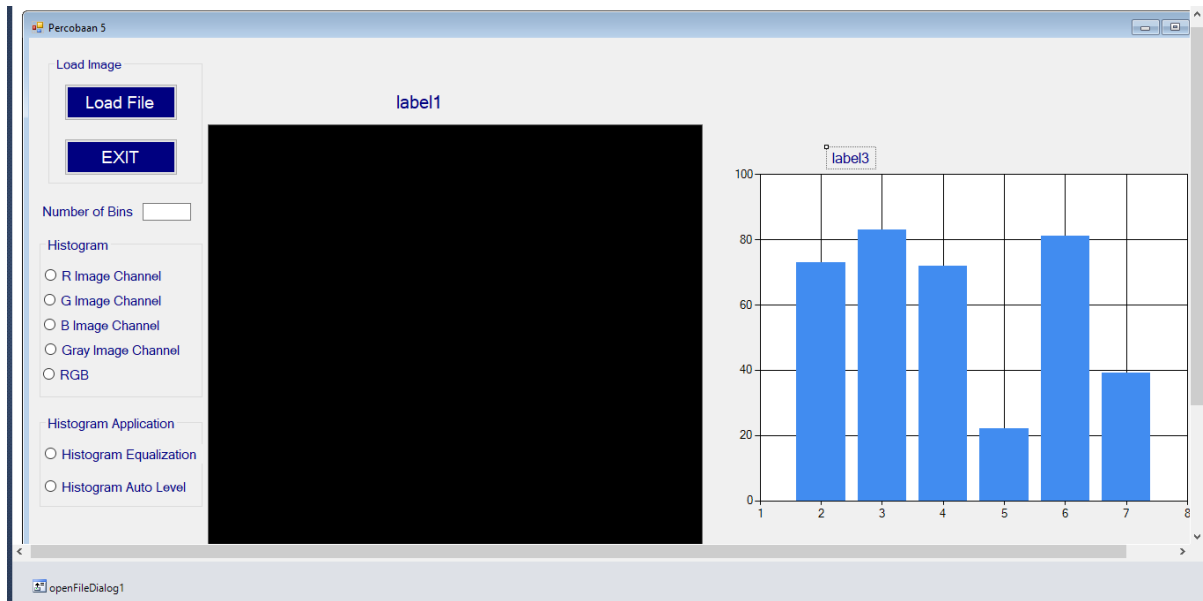
Prosedur

1. Tambahkah proyek baru dengan toolbox sebagai berikut :

- 1 **pictureBox** dan 1 buah **label**
- 1 buah **openFileDialog1**
- 2 buah tombol : **Load File** dan **Exit**
- 3 buah label : **Label1**, **Number of Bins** dan **label 3**
- 3 buah **groupbox** dan beri nama : **Load Image**, **Histogram** dan **Histogram Application**
- 7 buah **radioButton** dan beri label seperti gambar 5.3
- 1 buah **chart** untuk menampilkan histogram yang ada di



sehingga tampilannya seperti yang terlihat pada gambar 5.3.



Gambar 5.3. Tampilan Form untuk Latihan BAB 5

2. Setting properties dari **pictureBox** seperti pada latihan BAB 1
3. Setting properties dari **openFileDialog1** seperti pada latihan BAB 1
4. Tambahkan global variable sebagai berikut :

```
//global variable
Bitmap sourceImage;
int BIN = 256;           // jumlah histogram BIN
```

5. Tambahkan fungsi berikut di bawah InitializeComponent();

```
public Percobaan5()
{
    InitializeComponent();

    //tambahkan
    chart1.Series.Clear();
    label1.Text = "Original Image";
    textBox1.Text = "256";
    textBox1.TextAlign = HorizontalAlignment.Center;
}
```

6. Tambahkan beberapa fungsi berikut :

```
private void radioButtonHisClear()
{
    radioButton1.Checked = false;
    radioButton2.Checked = false;
    radioButton3.Checked = false;
    radioButton4.Checked = false;
    radioButton5.Checked = false;
}

private void radioButtonHisAppClear()
{
    radioButton6.Checked = false;
    radioButton7.Checked = false;
}
```

7. Tambahkan fungsi untuk mengecek textbox supaya nilainya diantara 0 -255

```
private void checkTextBox()
{
    if (int.Parse(textBox1.Text) < 0)
        textBox1.Text = "0";
    else if (int.Parse(textBox1.Text) > 256)
        textBox1.Text = "256";
}
```

8. Tambahkan fungsi untuk mengkonversi image:

```
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++)
        {
            //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);

            //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
            }
            else if (imageChannel == 3) //blue
            {
                // tambah coding sendiri
            }
            else if (imageChannel == 4) //gray
            {
                // tambah coding sendiri
            }
        }
    return convImage;
}
```

9. Tambahkan fungsi untuk menghitung histogram :

```
private float[] hitungHistogram(int imageChannel)
{
    //init of bins
    checkTextBox();
    BIN = int.Parse(textBox1.Text);

    //initializaation of histogram el
    float[] h = new float[BIN];

    //histogram init
    for (int i = 0; i < BIN; i++)
    {
        h[i] = 0;
    }

    //histogram calculation
    for (int x = 0; x < sourceImage.Width; x++)
        for (int y = 0; y < sourceImage.Height; y++)
        {
            Color w = sourceImage.GetPixel(x, y);

            int r = (int)(w.R * BIN / 256);
            int g = (int)(w.G * BIN / 256);
            int b = (int)(w.B * BIN / 256);

            //calculate gray channel
            int gray = (int)((0.5 * w.R + 0.419 * w.G + 0.081 * w.B) * BIN / 256);

            //calculate histogram
            if (imageChannel == 1)
                h[r] = h[r] + 1;
            else if (imageChannel == 2)
                h[g] = h[g] + 1;
            else if (imageChannel == 3)
                h[b] = h[b] + 1;
            else if (imageChannel == 4)
                h[gray] = h[gray] + 1;
        }
    return h;
}
```

10. Double klik tombol **Load File**, kemudian tuliskan sub rutin program berikut :

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        sourceImage = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);
        pictureBox1.Image = sourceImage;
        //delete the histogram
        if (chart1.Series.Count > 0)
        {
            chart1.Series.RemoveAt(0);
        }
        radioButtonHisClear();
        radioButtonHisAppClear();
        checkTextBox();
        BIN = int.Parse(textBox1.Text);
    }
}
```

11. Double klik radioButton1 **R Image Channel** dan tambahkan program berikut:

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

    if (radioButton1.Checked == false) return;
    radioButtonHisAppClear();

    int pilChannel = 1;

    //delete the histogram
    if (chart1.Series.Count > 0)
    {
        chart1.Series.RemoveAt(0);
    }

    //chart init
    chart1.Series.Add("Red Channel Image");
    chart1.Series["Red Channel Image"].Color = Color.Red;

    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["Red Channel Image"].Points.AddXY(i, his[i]);
    }

    //displaying Red Channel
    Bitmap redImage = imageConvert(pilChannel);
    pictureBox1.Image = redImage;

    label1.Text = "Red Channel Image";
    label1.ForeColor = Color.Red;
    label3.ForeColor = Color.Red;
    label3.Text = string.Format("Red Channel Image Histogram {0} Bins", BIN);
}
```


12. Tambahkan program untuk menghitung **histogram RGB** :

```
private float[] hitungHistogramRGB()
{
    //initializaation of histogram
    checkTextBox();
    BIN = int.Parse(textBox1.Text);

    float[] h = new float[BIN*3];

    //histogram init
    for (int i = 0; i < BIN*3; i++)
    {
        h[i] = 0;
    }

    //histogram calculation
    for (int x = 0; x < sourceImage.Width; x++)
        for (int y = 0; y < sourceImage.Height; y++)
        {
            Color w = sourceImage.GetPixel(x, y);

            int r = (int)(w.R * BIN / 256);
            int g = (int)(w.G * BIN / 256);
            int b = (int)(w.B * BIN / 256);

            //calculate histogram

            h[r] = h[r] + 1; //0 - BIN
            h[g + BIN] = h[g + BIN] + 1; //BIN - 2*BIN
            h[b + 2*BIN] = h[b + 2*BIN] + 1; //2*BIN - 3*BIN
        }
    return h;
}
```

13. Tambahkan fungsi untuk menampilkan **histogram RGB**

```
private void histogramRGBDisplay()
{
    //delete the histogram
    if (chart1.Series.Count > 0)
    {
        chart1.Series.RemoveAt(0);
    }

    //chart init
    chart1.Series.Add("RGB Image");
    chart1.Series["RGB Image"].Color = Color.Maroon;

    foreach (var series in chart1.Series)
    {
        series.Points.Clear();
    }

    float[] his = new float[BIN*3];

    his = hitungHistogramRGB();

    for (int i = 0; i < BIN*3; i++)
    {
        chart1.Series["RGB Image"].Points.AddXY(i, his[i]);
    }
}
```

```

label1.ForeColor = Color.Maroon;
label1.Text = "RGB Image";
label3.ForeColor = Color.Maroon;
label3.Text = string.Format("RGB Image Histogram {0} Bins", BIN*3);
}

```

14. Tambahkan program berikut untuk menghitung **histogram equalization** :

```

private float[] histogramEqualization()
{
    Bitmap grayEqIm = new Bitmap(sourceImage);
    float[] h = new float[BIN];
    float[] c = new float[BIN];

    //histogram init
    for (int i = 0; i < BIN; i++)
    {
        h[i] = 0;
    }
    //histogram calculation
    for (int x = 0; x < sourceImage.Width; x++)
        for (int y = 0; y < sourceImage.Height; y++)
        {
            Color w = sourceImage.GetPixel(x, y);
            byte r = w.R;
            byte g = w.G;
            byte b = w.B;

            //calculate gray channel
            byte gray = (byte)(0.5 * r + 0.419 * g + 0.081 * b);
            h[gray] = h[gray] + 1;
        }

    c[0] = h[0];

    //calculate the cummulative histogram
    for (int i = 1; i < 256; i++)
        c[i] = c[i - 1] + h[i];

    //image equalization
    for (int x = 0; x < sourceImage.Width; x++)
        for (int y = 0; y < sourceImage.Height; y++)
        {
            Color w = sourceImage.GetPixel(x, y);
            byte r = w.R;
            byte g = w.G;
            byte b = w.B;

            byte gray = (byte)(0.5 * r + 0.419 * g + 0.081 * b);
            byte grayEq = (byte)(255 * c[gray] / (sourceImage.Width *
sourceImage.Height));

            Color gEq = Color.FromArgb(grayEq, grayEq, grayEq);
            grayEqIm.SetPixel(x, y, gEq);
            h[grayEq] = h[grayEq] + 1;
        }
    //displaying Equalization Image
    pictureBox1.Image = grayEqIm;

    return h;
}

```

15. Double klik radioButton5 **RGB** dan tambahkan program berikut :

```
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
    if (sourceImage == null) return;
    if (radioButton5.Checked == false) return;
    radioButtonHisAppClear();

    histogramRGBDisplay();

    pictureBox1.Image = sourceImage;
}
```

16. Double klik radioButton6 **Histogram Equalization** dan tambahkan program berikut:

```
private void radioButton6_CheckedChanged(object sender, EventArgs e)
{
    if (sourceImage == null) return;
    if (radioButton6.Checked == false) return;
    radioButtonHisClear();

    //delete the histogram
    if (chart1.Series.Count > 0)
    {
        chart1.Series.RemoveAt(0);
    }

    //chart init
    chart1.Series.Add("EQ");
    chart1.Series["EQ"].Color = Color.Silver;

    foreach (var series in chart1.Series)
    {
        series.Points.Clear();
    }

    float[] his = new float[BIN];

    his = histogramEqualization();

    for (int i = 0; i < BIN; i++)
    {
        chart1.Series["EQ"].Points.AddXY(i, his[i]);
    }

    label1.ForeColor = Color.Navy;
    label1.Text = "Equalization Image";
    label3.ForeColor = Color.Navy;
    label3.Text = string.Format("Histogram Equalization {0} Bins", BIN);
}
```

17. Tambahkan fungsi berikut untuk menghitung **Histogram Auto Level** :

```
private float[] histogramAutoLevel()
{
    Bitmap grayAuto = new Bitmap(sourceImage);
    float[] h = new float[BIN];
    int grayMin = 255;
    int grayMax = 0;

    //histogram init
    for (int i = 0; i < BIN; i++)
    {
        h[i] = 0;
    }

    //histogram calculation
    for (int x = 0; x < sourceImage.Width; x++)
        for (int y = 0; y < sourceImage.Height; y++)
        {
            Color w = sourceImage.GetPixel(x, y);
            int r = (int)(w.R * BIN / 256);
            int g = (int)(w.G * BIN / 256);
            int b = (int)(w.B * BIN / 256);

            //calculate gray channel
            int gray = (int)((0.5 * w.R + 0.419 * w.G + 0.081 * w.B) * BIN / 256);
            if (grayMin > gray) grayMin = gray;
            if (grayMax < gray) grayMax = gray;
        }

    //histogram AutoLevel calculation
    for (int x = 0; x < sourceImage.Width; x++)
        for (int y = 0; y < sourceImage.Height; y++)
        {
            Color w = sourceImage.GetPixel(x, y);
            int r = (int)(w.R * BIN / 256);
            int g = (int)(w.G * BIN / 256);
            int b = (int)(w.B * BIN / 256);

            //calculate gray channel
            int gray = (int)((0.5 * w.R + 0.419 * w.G + 0.081 * w.B) * BIN / 256);
            int grayEq = (byte)(255 * (gray - grayMin) / (grayMax - grayMin) * BIN / 256);

            Color gEq = Color.FromArgb(grayEq, grayEq, grayEq);
            grayAuto.SetPixel(x, y, gEq);

            h[grayEq] = h[grayEq] + 1;
        }

    //displaying Auto Level Image
    pictureBox1.Image = grayAuto;

    return h;
}
```

18. Double klik radioButton7 **Histogram Auto Level** dan tambahkan program berrikut :

```
private void radioButton7_CheckedChanged(object sender, EventArgs e)
{
    if (sourceImage == null) return;
    if (radioButton7.Checked == false) return;
    radioButtonHisClear();

    //delete the histogram
    if (chart1.Series.Count > 0)
    {
        chart1.Series.RemoveAt(0);
    }

    //chart init
    chart1.Series.Add("Auto");
    chart1.Series["Auto"].Color = Color.Silver;

    foreach (var series in chart1.Series)
    {
        series.Points.Clear();
    }

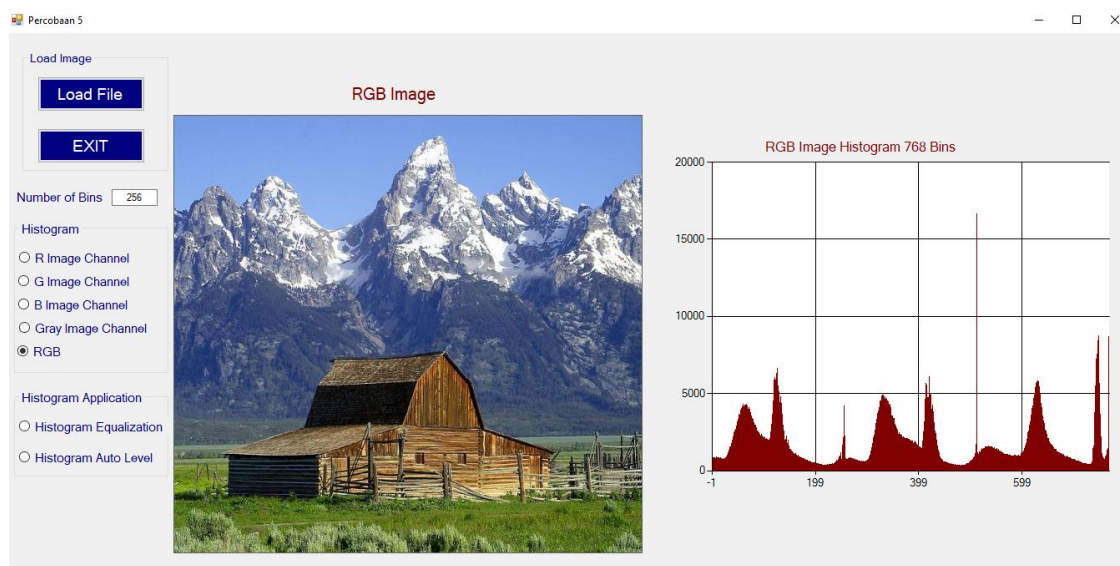
    float[] his = new float[BIN];

    his = histogramAutoLevel();

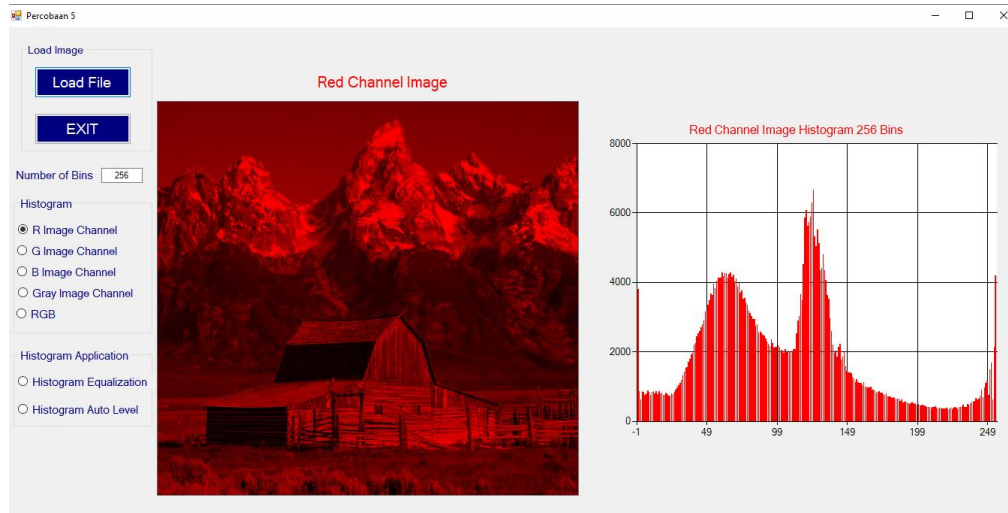
    for (int i = 0; i < BIN; i++)
    {
        chart1.Series["Auto"].Points.AddXY(i, his[i]);
    }

    label1.Text = "Auto Level Image";
    label1.ForeColor = Color.Navy;
    label3.ForeColor = Color.Navy;
    label3.Text = string.Format("Histogram AutoLevel {0} Bins", BIN);
}
}
```

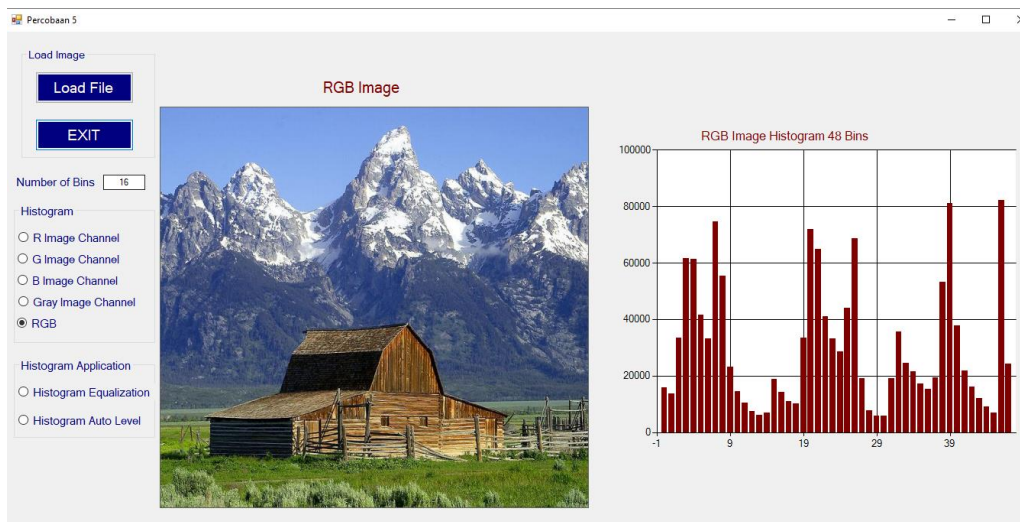
19. Jalankan program dan load salah satu file gambar. Berikut adalah hasil tampilan histogram untuk beberapa fungsi :



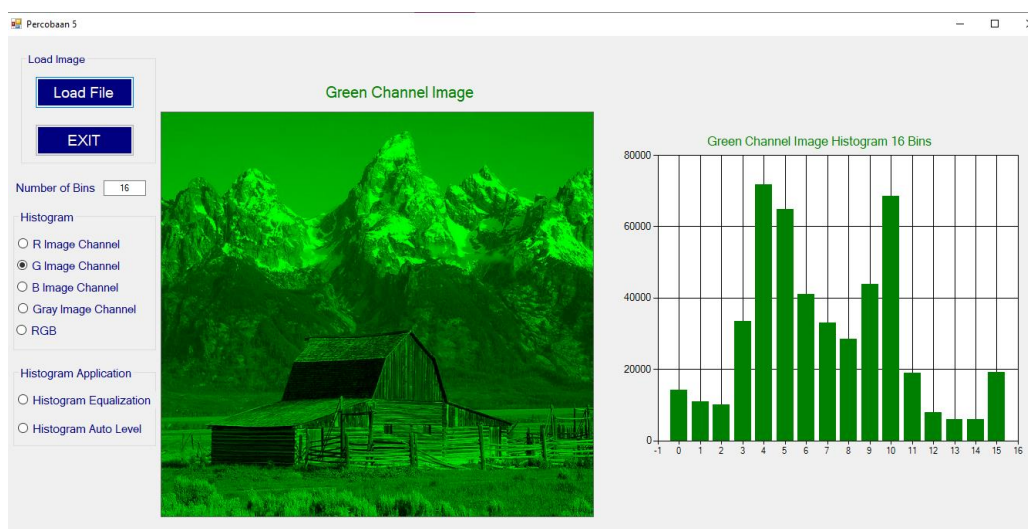
RGB Image $256 \times 3 = 768$ BIN



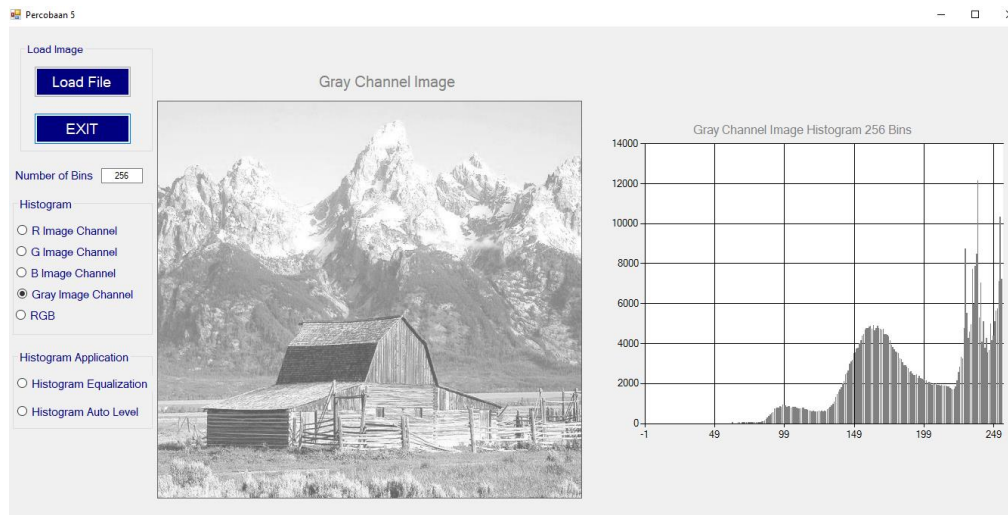
R Channel Image 256 BIN



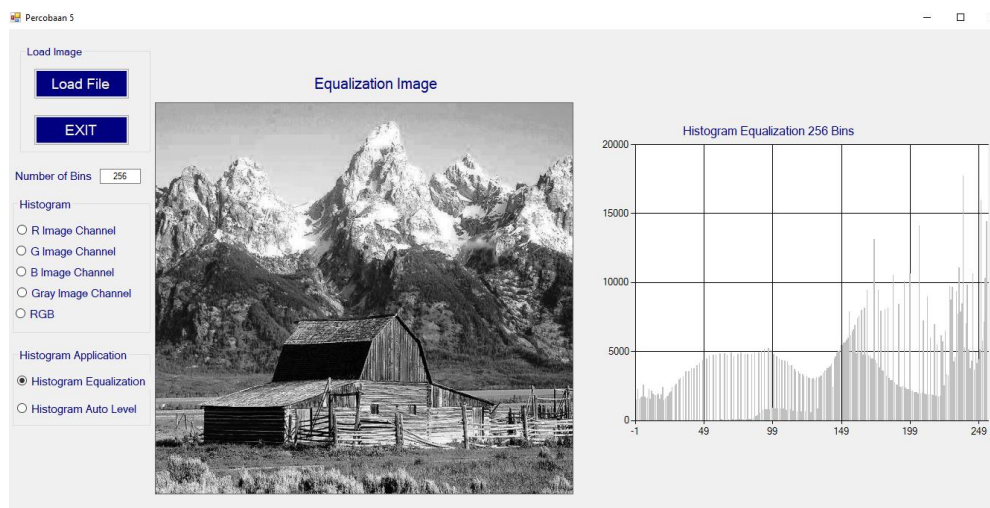
RGB Image $16 \times 3 = 48$ BIN



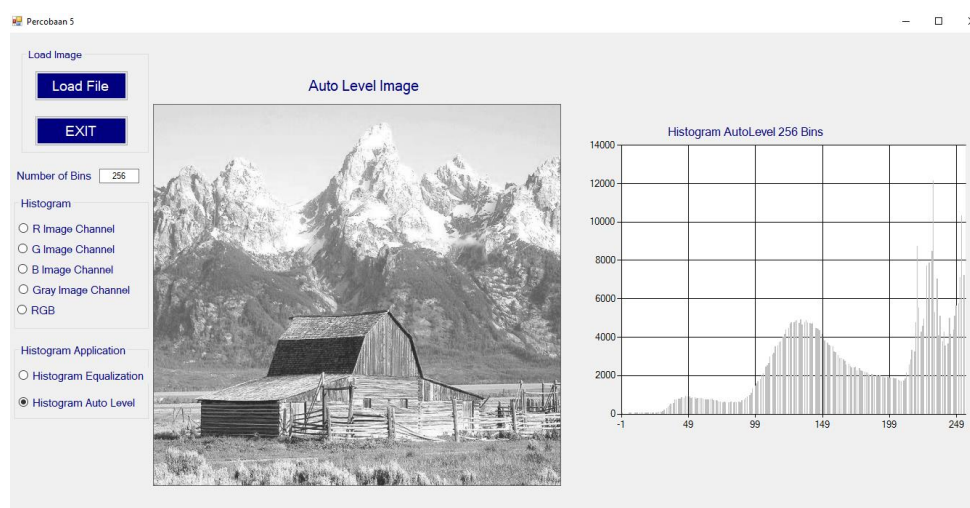
Green Image 16 BIN



Gambar dan Histogram dari original image



Gambar dan Histogram setelah equalization



Gambar dan Histogram setelah auto level

5.4 Tugas dan Pertanyaan

1. Tambahkan kode program untuk radioButton **G Image Channel**, **B Image Channel** dan **Gray Image Channel**?
2. Tambahkan kode program `private Bitmap imageConvert(int imageChannel)` untuk menghitung konversi citra ke G, B dan Gray?
3. Editlah suatu image sehingga menjadi lebih gelap atau lebih terang, kemudian gunakan histogram equalization dan auto level pada citra tersebut.