

3

MANIPULASI CITRA RGB

Pada pokok bahasan ini, mahasiswa akan mempelajari tentang cara memanipulasi citra RGB ke bentuk lain.

Pokok Bahasan :

1. Citra RGB, Citra Gray, Citra Biner dan Citra Inversi
2. Resampling Citra dan Kuantisasi Citra
3. Transformasi intensitas citra

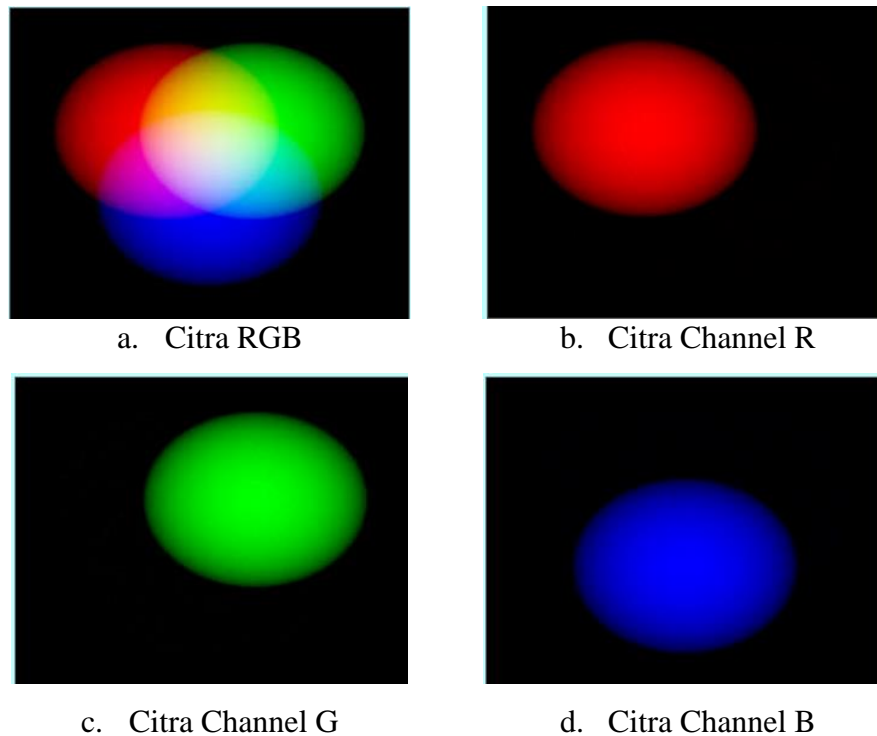
Latihan :

1. Membuat aplikasi mengubah citra RGB ke tiga channel citra yaitu citra channel R, G dan B
2. Membuat aplikasi mengubah citra RGB ke citra gray, citra biner dan citra inversi
3. Membuat aplikasi meresampling dan mengkuantisasi citra
4. Membuat aplikasi transformasi intensitas citra

3.1 Citra RGB

Capaian pembelajaran : mampu memahami citra RGB

Citra RGB adalah citra yang mempunyai komponen warna *R(red)*, *G(green)*, dan *B(blue)* masing-masing mempunyai nilai 8 bit (0 sampai 255). Dalam citra RGB, setiap pixel diwakili oleh nilai RGB tersebut. Gambar 3.1 menunjukkan sebuah Citra RGB dan konversi citra ke masing-masing channel.



Gambar 3.1 Citra RGB dan masing-masing channel R, G, dan B

3.2 Citra Gray

Capaian pembelajaran : mampu memanipulasi citra RGB ke citra Gray.

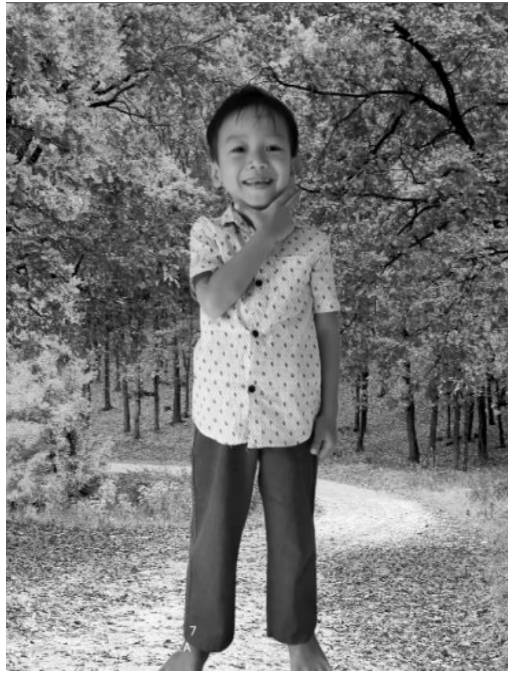
Citra gray adalah citra yang hanya diwakili oleh satu nilai pada tiap pixel. Nilai intensitas pixel pada image gray adalah 8 bit (0-255). Nilai gray dapat ditentukan oleh persamaan sebagai berikut :

$$Gray = \frac{(R + G + B)}{3}$$

Atau

$$Gray = 0.5 R + 0.419 G + 0.081 B$$

Gambar 3.2 menunjukkan citra hasil konversi dari RGB ke Gray dengan menggunakan persamaan di atas.



Gambar 3.2 Citra gray

3.3 Citra Biner

Capaian pembelajaran : mampu memanipulasi citra RGB ke citra Sepia.

Citra Biner adalah citra yang hanya memiliki dua buah nilai intensitas yaitu 0 dan 255. Untuk mendapatkan citra biner, dapat dilakukan suatu pengambangan (*threshoding*) pada suatu citra gray seperti berikut:

$$Binary Image = \begin{cases} 255, & \text{if intensity} > TH \\ 0, & \text{if intensity} \leq TH \end{cases}$$

Gambar 3.3 menunjukkan citra biner dengan *threshold* sebesar 100.



Gambar 3.3 Citra Biner dengan TH = 100

3.4 Citra Inversi

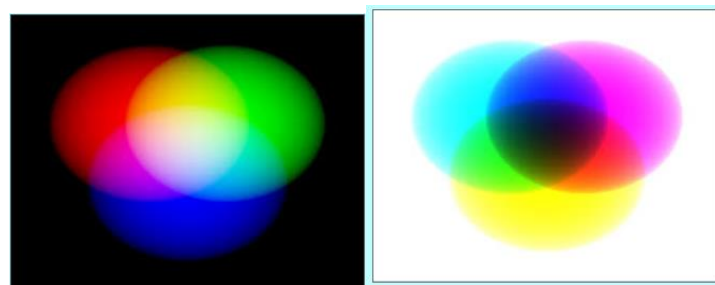
Capaian pembelajaran : mampu memanipulasi citra RGB ke citra Inversi.

Citra inversi (citra negatif) adalah citra yang mempunyai nilai pixel kebalikan dari pixel sebelumnya. Hasil invers terhadap nilai intensitas dapat dipakai untuk meningkatkan mutu citra dengan detail warna putih atau keabuan yang ada dalam daerah gelap citra, terutama ketika daerah hitamnya dominan seperti ditunjukkan pada gambar 3.4. Persamaan invers citra ditunjukkan oleh persamaan berikut:

$$R_{invers} = 255 - R$$

$$G_{invers} = 255 - G$$

$$B_{invers} = 255 - B$$



a. Original

b. inversi

Gambar 3.4 Citra inversi

3.5 Image Resampling

Capaian pembelajaran : mampu memahami proses resampling citra

Resampling Image adalah proses mendefinisikan kembali resolusi dari suatu image. Salah satu metode yang digunakan adalah dengan mencari rata-rata beberapa pixel dan menjadikannya satu pixel yang baru. Semakin besar resolusi suatu citra maka semakin bagus gambar tersebut, sebaliknya jika semakin kecil maka gambar akan terpecah dan menjadi kaku. Resampling image gambar 3.5 banyak digunakan untuk menyamarkan suatu objek agar tidak terlihat disekitar object lainnya.



a. Original

b. resampling 1/16 kali

Gambar 3.5 Citra hasil resampling

3.6 Image Quantization

Capaian pembelajaran : mampu memahami proses quantisasi suatu citra

Kuantisasi citra gambar 3.6 adalah proses mengubah jumlah nilai keabuan, misalnya menjadi 8 nilai derajat keabuan. Untuk melakukan kuantisasi n step nilai dapat menggunakan rumus kuantisasi berikut ini:

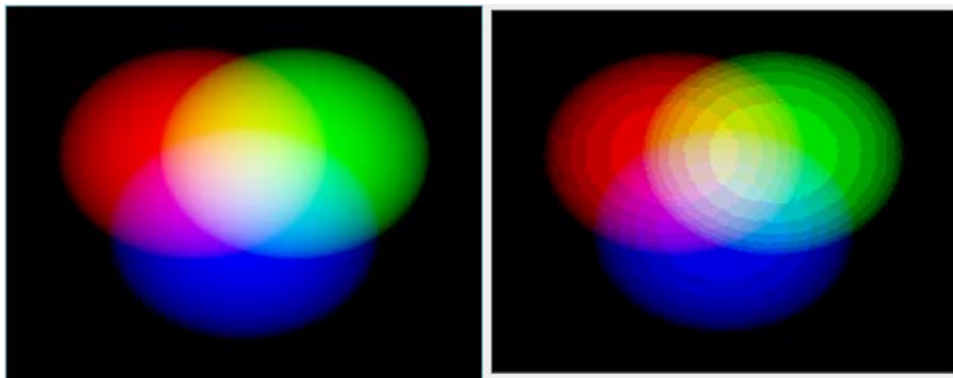
$$xk = n * (int)(\frac{x}{n})$$

$n = 1 \Rightarrow$ 256 step nilai: 0 1 2 3 4 5 6 7 8 9 ... 255

$n = 2 \Rightarrow$ 128 step nilai: 0 2 4 6 8 10 12 14 ... 254

$n = 4 \Rightarrow$ 64 step nilai: 0 4 8 12 16 20 24 ... 252

$n = 16 \Rightarrow$ 16 step nilai: 0 16 32 48 64 80 ... 240



a. Original

b. image kuantisasi 16 x

Gambar 3.6 Citra hasil kuantisasi

3.7 Image Brightness

Brightness adalah proses untuk menambah/mengurangi kecerahan citra. Jika intensitas pixel dikurangi dengan nilai tertentu maka citra akan menjadi lebih gelap, dan sebaliknya jika intensitas pixelnya ditambah dengan nilai tertentu maka akan lebih terang

Brightness dapat dirumuskan sebagai berikut :

$$R_{new} = R + k$$

$$G_{new} = G + k$$

$$B_{new} = B + k$$

Dimana R_{new} , G_{new} dan B_{new} adalah intensitas citra setelah mendapat perubahan intensitas sebesar k . Gambar 3.5 menunjukkan gambar hasil penambahan intensitas sebesar 100.



a. Original



b. brightness bertambah 100

Gambar 3.7 Citra hasil penambahan brightness

3.8 Image Contrast

Contrast adalah perbedaan pencahayaan dan / atau warna yang membuat obyek (atau perwakilannya dalam gambar atau layar) dibedakan. Dalam persepsi visual dari dunia nyata, Contrast ditentukan oleh perbedaan warna dan kecerahan obyek dan objek lain dalam bidang yang sama pandang.



a. Original



b. contrast bertambah 50

Gambar 3.7 Citra hasil penambahan contrast

3.9 Latihan

Tujuan

Memahami cara membaca file gambar dengan microsoft visual C#

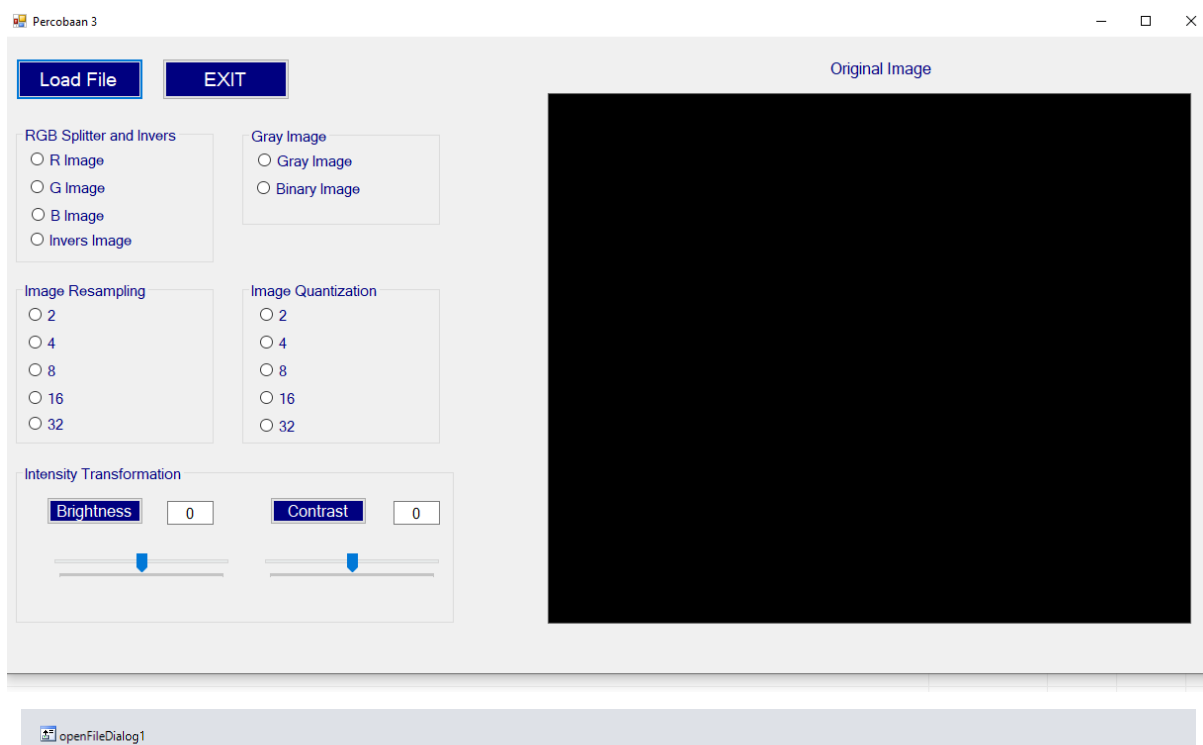
Mampu memanipulasi data citra

Prosedur

1. Tambahkah proyek baru dengan toolbox sebagai berikut :

- 1 buah **pictureBox** dan 1 buah **label** dengan nama : **original Image**
- **openFileDialog1**
- 4 buah **button** : **Load File**, **Exit**, **Brightness** dan **Contrast**
- 5 buah **groupBox** : **RB Splitter dan Invers**, **Gray Image**, **Image Resampling**, **Image Quantization**, **Intensity Transformation**.
- 16 buah **radioButton**
- 2 buah **trackBar**

sehingga tampilannya seperti yang terlihat pada gambar 3.5.



Gambar 3.5. Tampilan VC# setelah membuat proyek baru

2. Setting properties dari **pictureBox** seperti percobaan 1
3. Setting properties dari **openFileDialog1** seperti percobaan 1

4. Tambahkan global variable sebagai berikut :

```
//global variable
Bitmap sourceImage, processingImage;
int imageHeight, imageWidth;

// number of processing image
int imProcNo;

//level of resampling image
int resampleLevel;

//level of intensity quantization
int quantizationLevel;
```

5. Tambahkan inialisasi trackbar dan textbox di bawah fungsi **InitializeComponent()**sbb:

```
public Percobaan3()
{
    InitializeComponent();

    //trackBar dan textBox initialization
    trakbarInitialization();
    textBoxInitialization();
}

//trackBar initialization
private void trakbarInitialization()
{
    //Brightness TrackBar
    trackBar1.Minimum = -255;
    trackBar1.Maximum = 255;

    //Contrants TrackBar
    trackBar2.Minimum = -100;
    trackBar2.Maximum = 100;

    //trackBar init pos
    trackBar1.Value = 0;
    trackBar2.Value = 0;
}

//trackBar dan textBox initialization
private void textBoxInitialization()
{
    textBox1.Text = "0";
    textBox2.Text = "0";
}
```


6. Tambahkan juga fungsi berikut untuk mereset kondisi semua toolbox ke posisi awal

```
private void resetCondition()
{
    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;

    trackBar1.Value = 0; trackBar2.Value = 0;
    textBox1.Text = "0"; textBox2.Text = "0";
}
```

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

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        resetCondition();

        //loading source image
        sourceImage = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);
        processingImage = new Bitmap (sourceImage);

        //menampilkan sourceImage di pictureBox1
        pictureBox1.Image = sourceImage;

        //menampilkan label image sebagai "Original Image"
        label1.Text = "Original Image";

        //mencari tinggi dan lebar image
        imageHeight = sourceImage.Height;
        imageWidth = sourceImage.Width;
    }
}
```

8. Double klik radioButton **R Image** dan tambahkan rutin berikut :

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

    //change the label to R image
    label1.Text = "Red Image";

    //setting the image
    setImageProcessing(imProcNo);
}
```

9. Tambahkan fungsi berikut sebagai fungsi untuk melakukan image processing :

```
private void setImageProcessing(int procNo)
{
    for (int x = 0; x < imageWidth; x++)
        for (int y = 0; y < imageHeight; y++)
        {
            //get the RGB value of the pixel at (x,y)
            Color w = sourceImage.GetPixel(x, y);

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

                //change the image to R image
                processingImage.SetPixel(x, y, redColor);
            }
            //G Image
            else if (procNo == 2)
            {
                //Tambahkan koding sendiri
            }
            //B Image
            else if (procNo == 3)
            {
                //Tambahkan koding sendiri
            }
            //Invers Image
            else if (procNo == 4)
            {
                //Tambahkan koding sendiri
            }
            //Gray Image && binary image
            else if (procNo == 5 || procNo == 6)
            {
                int r = w.R; //red value
                int g = w.G; //green value
                int b = w.B; //blue value

                int grayValue = (int)(0.5 * r + 0.419 * g + 0.081 * b);

                //because the maximum intensity is 255
                if (grayValue > 255) grayValue = 255;

                //Binary Image
                if (procNo == 6)
                {
                    //thresholding for making binary image, for ex. TH = 100;
                    int TH = 100;
                    if (grayValue > TH) grayValue = 255; //white pixels
                    else grayValue = 0; //black pixels
                }

                Color grayColor = Color.FromArgb(grayValue, grayValue, grayValue);
                processingImage.SetPixel(x, y, grayColor);
            }
        }

    pictureBox1.Image = processingImage;
}
```

10. Double klik tombol **Gray Image** tambahkan rutin berikut :

```
private void radioButton5_CheckedChanged(object sender, EventArgs e)
{
    if (sourceImage == null) return;
    imProcNo = 5;
    //change the label to R image
    label1.Text = "Gray Image";

    //setting the image
    setImageProcessing(imProcNo);
}
```

11. Double klik tombol **Binary Image** tambahkan rutin berikut :

```
private void radioButton6_CheckedChanged(object sender, EventArgs e)
{
    if (sourceImage == null) return;
    imProcNo = 6;
    //change the label to R image
    label1.Text = "Binary Image";

    //setting the image
    setImageProcessing(imProcNo);
}
```

12. Tambahan fungsi berikut untuk setting resample image:

```
private void setResampleLevel(int iLevel)
{
    resampleLevel = iLevel;
}
```

13. Tambahkan fungsi berikut untuk setting quantization image:

```
private void setQuantizationLevel(int iLevel)
{
    quantizationLevel = iLevel;
}
```

14. Double klik radioButton **Image Resampling 2** dan tambahkan rutin berikut:

```
private void radioButton7_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton7.Checked == false) return;
    //change the label to Resample image
    label1.Text = "Resample Image 1/4"; //1/4 = 1/2 * 1/2
    setResampleLevel(2);
    imageResample();
}
```

15. Double klik radioButton **Image Quantization 2** dan tambahkan rutin berikut:

```
private void radioButton12_CheckedChanged(object sender, EventArgs e)
{
    if (radioButton12.Checked == false) return;
    setQuantizationLevel(2);
    imageQuantization();
}
```

16. Tambahkan fungsi berikut untuk melakukan image resampling

```
private void imageResample()
{
    if (sourceImage == null) return;

    //resampling to new Width and new Height
    int ht = (int)(imageHeight / resampleLevel);
    int wd = (int)(imageWidth / resampleLevel);

    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 < resampleLevel; k++)
            {
                for (l = 0; l < resampleLevel; l++)
                {
                    Color w = sourceImage.GetPixel(j * resampleLevel + l, i *
resampleLevel + 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 / (resampleLevel * resampleLevel));
            new_valueG = (int)(new_valueG / (resampleLevel * resampleLevel));
            new_valueB = (int)(new_valueB / (resampleLevel * resampleLevel));

            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 < resampleLevel; k++)
            {
                for (l = 0; l < resampleLevel; l++)
                {
                    processingImage.SetPixel(j * resampleLevel + l, i *
resampleLevel + k, colorRed);
                }
            }
        }
    }

    pictureBox1.Image = processingImage;
}
```

17. Tambahkan fungsi berikut untuk melakukan **image kuantization**

```
private void imageQuantization()
{
    if (sourceImage == null) return;

    for (int x = 0; x < imageWidth; x++)
        for (int y = 0; y < imageHeight; y++)
        {
            Color w = sourceImage.GetPixel(x, y);
            int r = w.R;
            int g = w.G;
            int b = w.B;

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

            Color wb = Color.FromArgb(rk, gk, bk);
            processingImage.SetPixel(x, y, wb);
        }

    pictureBox1.Image = processingImage;
}
```

18. Tambahkan fungsi berikut untuk mengatur **brightness**

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

    for (int x = 0; x < imageWidth; x++)
        for (int y = 0; y < imageHeight; y++)
        {
            Color w = processingImage.GetPixel(x, y);
            int R = (int)(brightness + w.R);
            //because intensity is between 0-255
            if (R > 255) R = 255; if (R < 0) R = 0;

            int G = (int)(brightness + w.G);
            //because intensity is between 0-255
            if (G > 255) G = 255; if (G < 0) G = 0;

            int B = (int)(brightness + w.B);
            //because intensity is between 0-255
            if (B > 255) B = 255; if (B < 0) B = 0;

            //setting the new color
            Color wBaru = Color.FromArgb(R, G, B);

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

19. Double klik button **Brightness** dan tambahkan fungsi berikut :

```
private void button3_Click(object sender, EventArgs e)
{
    if (processingImage == null) return;

    int brightness = int.Parse(textBox1.Text);

    if (brightness < 0 || brightness > 255) return;

    //setting brightness
    setBrightness(brightness);

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

20. Tambahkan fungsi berikut untuk mengatur **contrast**

```
private void setContrast(double contrast)
{
    //inisialisasi processing image
    Bitmap cImage = new Bitmap(processingImage);

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

    for (int x = 0; x < imageWidth; x++)
        for (int y = 0; y < imageHeight; y++)
        {
            Color w = processingImage.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;
}
```

21. Double klik trackBar **Contrast** dan tambahkan fungsi berikut :

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

    //setting contrast
    setContrast(contrast);

    //menampilkan nilai contrast pada textbox
    textBox2.Text = string.Format("{0}", trackBar2.Value);
}
```

22. Jalankan program dan load salah satu file gambar.

3.10 Tugas dan Pertanyaan

1. Tambahkan kode program untuk radioButton **G Image, B Image dan Invers Image**.
2. Tambahkan kode program pada fungsi `setImageProcessing(int procNo)` untuk menampilkan GImage, BImage dan Invers Image
3. Tambahkan kode program untuk **radioButton Image Resampling 4, 8, 16 dan 32**
4. Tambahkan kode program untuk **radioButton Image Quantization 4, 8, 16 dan 32**
5. Tambahkan kode program untuk **trackBar Brightness**
6. Tambahkan kode program untuk **button Contrast**