UJIAN TENGAH SEMESTER

PENGOLAHAN CITRA

MK401

POLITEKNIK NEGERI Batam

Disusun oleh :

Ricky Silitonga (4211901034)

PROGRAM STUDI TEKNIK MEKATRONIKA

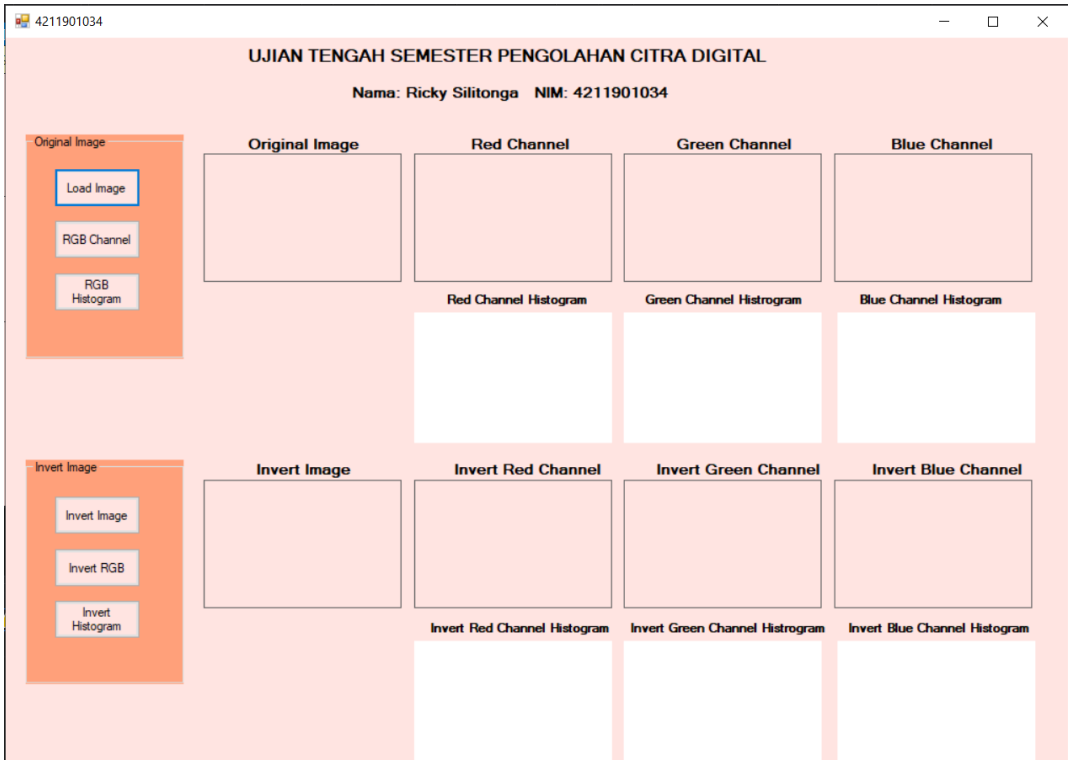
JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI BATAM

2020

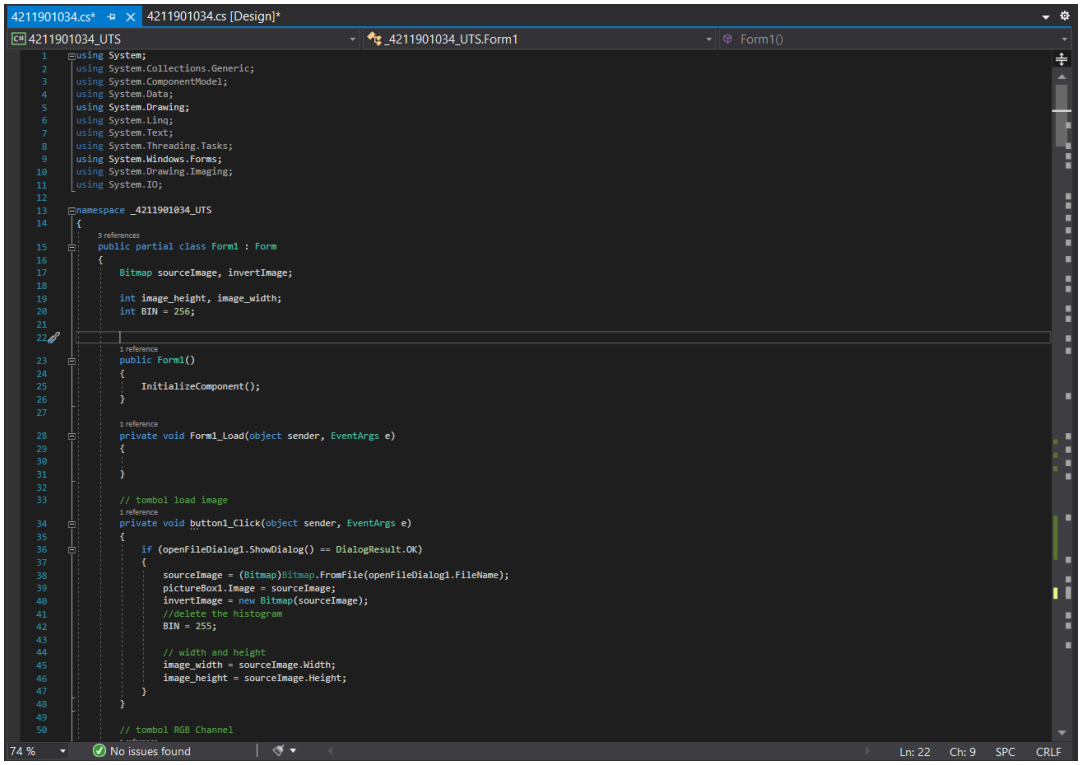
UTS PENGOLAHAN CITRA

1. Screen shoot GUI dan harus memuat NIM dan Nama

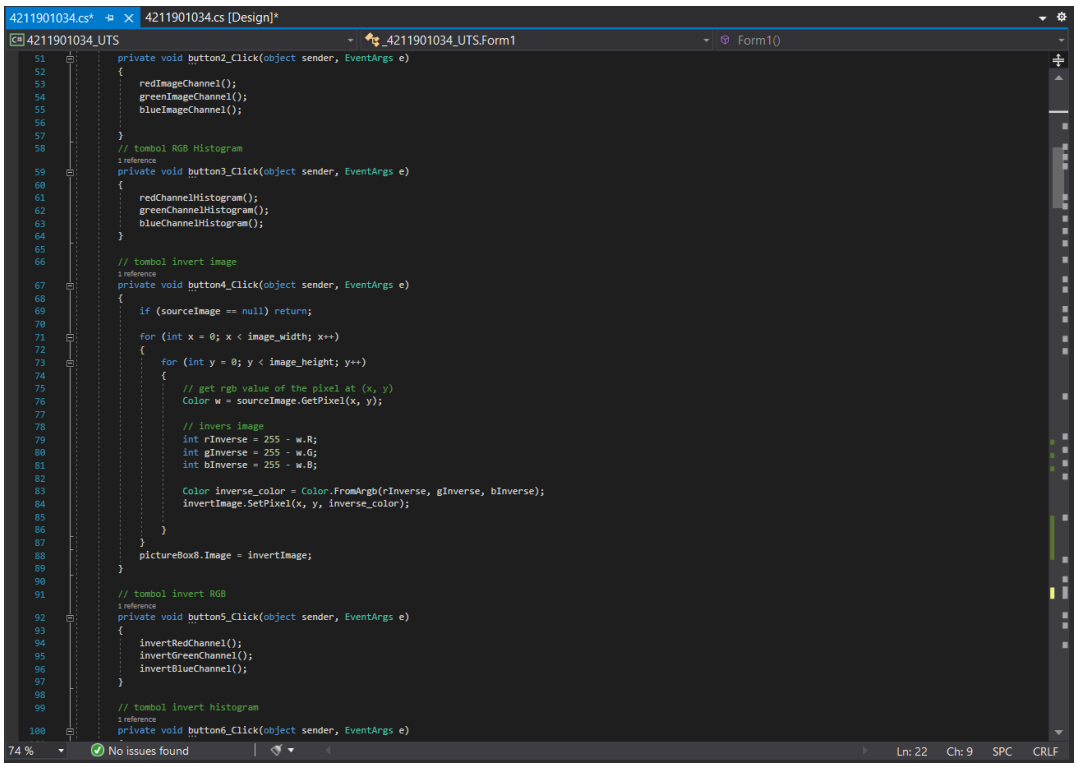


Graphical User Interface (GUI)

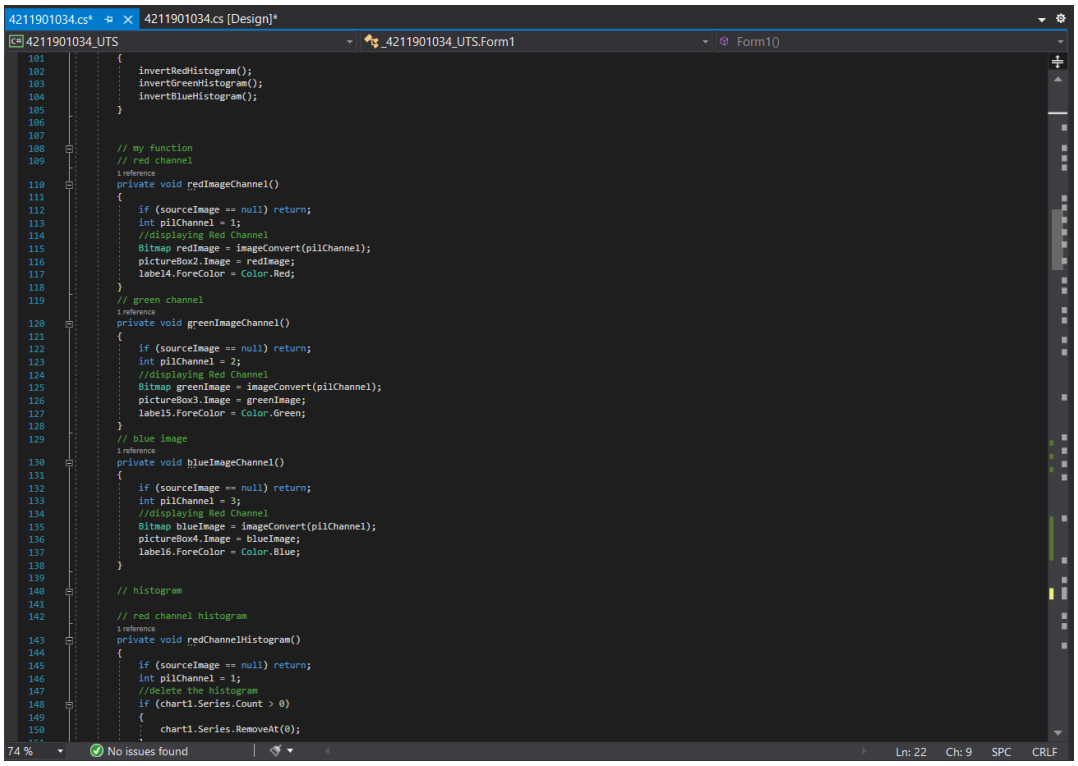
2. Screen shoot source seluruh code anda harus memuat nim



```
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_UTS
{
    3 references
    public partial class Form1 : Form
    {
        Bitmap sourceImage, invertImage;

        int image_height, image_width;
        int BIN = 256;

        1 reference
        public Form1()
        {
            InitializeComponent();
        }

        1 reference
        private void Form1_Load(object sender, EventArgs e)
        {

        }

        // tombol load image
        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            if (openFileDialog1.ShowDialog() == DialogResult.OK)
            {
                sourceImage = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);
                pictureBox1.Image = sourceImage;
                invertImage = new Bitmap(sourceImage);
                //delete the histogram
                BIN = 255;

                // width and height
                image_width = sourceImage.Width;
                image_height = sourceImage.Height;
            }
        }

        // tombol RGB Channel
```

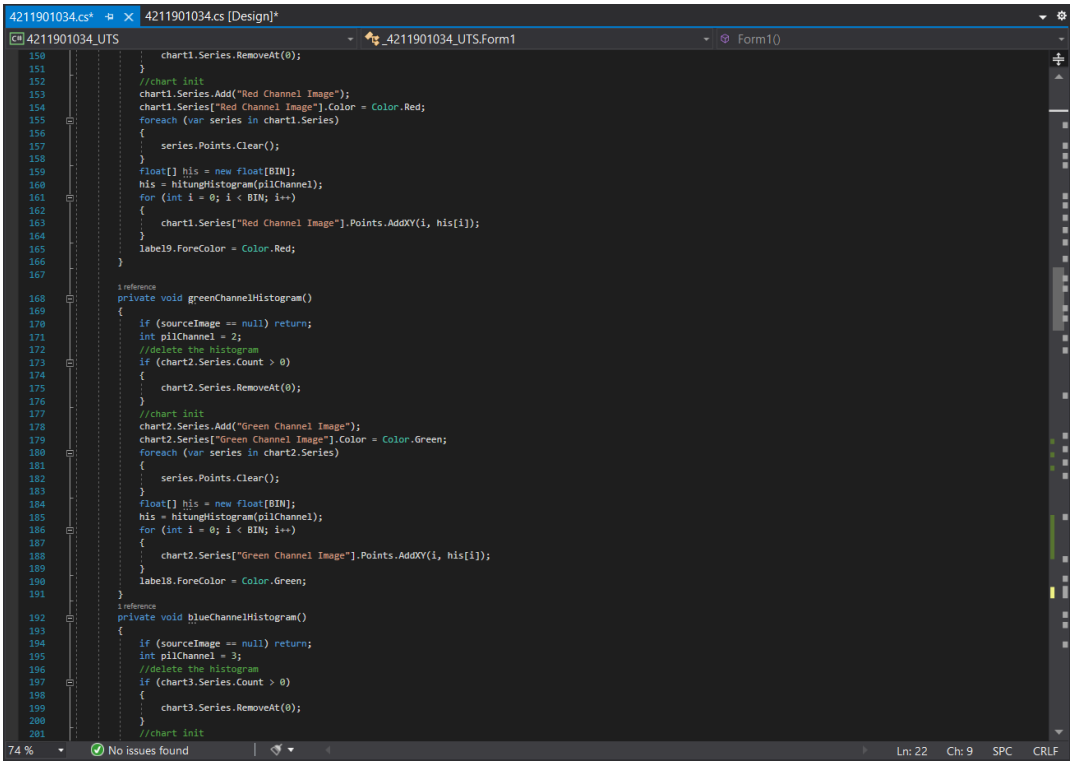


```
        private void button2_Click(object sender, EventArgs e)
        {
            redImageChannel();
            greenImageChannel();
            blueImageChannel();

        }
        // tombol RGB Histogram
        1 reference
        private void button3_Click(object sender, EventArgs e)
        {
            redChannelHistogram();
            greenChannelHistogram();
            blueChannelHistogram();
        }

        // tombol invert image
        1 reference
        private void button4_Click(object sender, EventArgs e)
        {
            if (sourceImage == null) return;

            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 = sourceImage.GetPixel(x, y);

                    // invers image
                    int rInverse = 255 - w.R;
                    int gInverse = 255 - w.G;
                    int bInverse = 255 - w.B;

                    Color inverse_color = Color.FromArgb(rInverse, gInverse, bInverse);
                    invertImage.SetPixel(x, y, inverse_color);

                }
            }
            pictureBox8.Image = invertImage;
        }

        // tombol invert RGB
        1 reference
        private void button5_Click(object sender, EventArgs e)
        {
            invertRedChannel();
            invertGreenChannel();
            invertBlueChannel();
        }

        // tombol invert histogram
        1 reference
        private void button6_Click(object sender, EventArgs e)
```



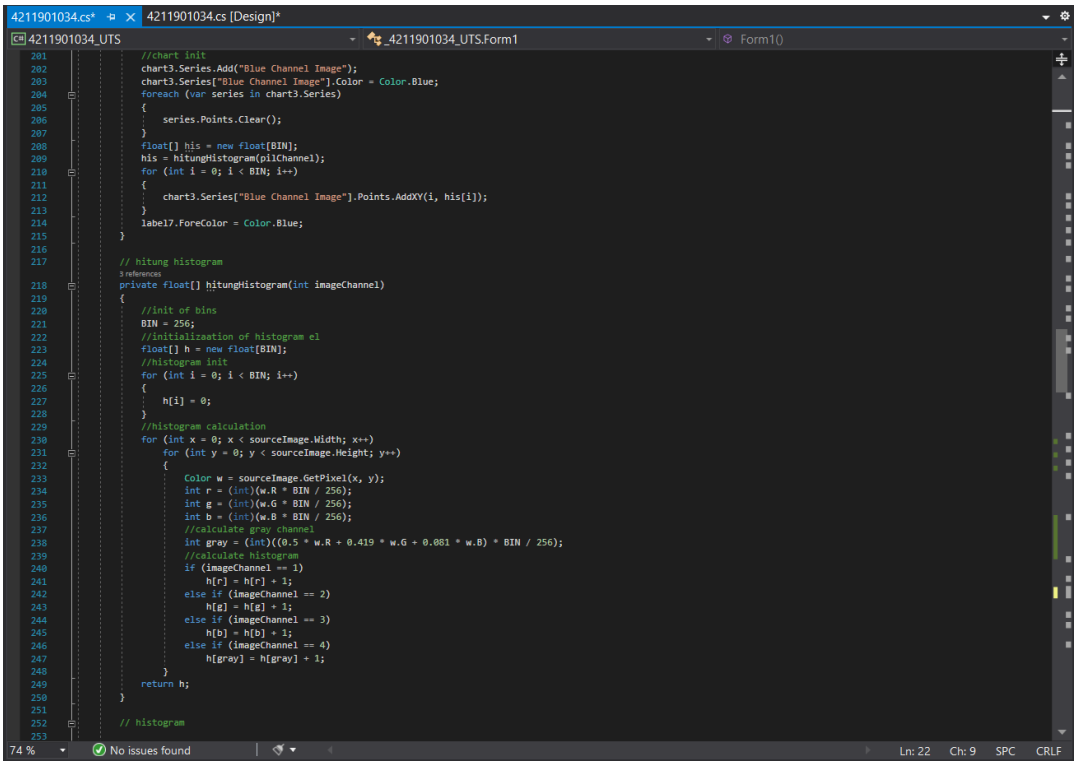
```
{
    invertRedHistogram();
    invertGreenHistogram();
    invertBlueHistogram();
}


// my function
// red channel
1 reference
private void redImageChannel()
{
    if (sourceImage == null) return;
    int pilChannel = 1;
    //displaying Red Channel
    Bitmap redImage = imageConvert(pilChannel);
    pictureBox2.Image = redImage;
    label4.ForeColor = Color.Red;
}
// green channel
1 reference
private void greenImageChannel()
{
    if (sourceImage == null) return;
    int pilChannel = 2;
    //displaying Red Channel
    Bitmap greenImage = imageConvert(pilChannel);
    pictureBox3.Image = greenImage;
    label5.ForeColor = Color.Green;
}
// blue image
1 reference
private void blueImageChannel()
{
    if (sourceImage == null) return;
    int pilChannel = 3;
    //displaying Red Channel
    Bitmap blueImage = imageConvert(pilChannel);
    pictureBox4.Image = blueImage;
    label6.ForeColor = Color.Blue;
}

// histogram

// red channel histogram
1 reference
private void redChannelHistogram()
{
    if (sourceImage == null) return;
    int pilChannel = 1;
    //delete the histogram
    if (chart1.Series.Count > 0)
    {
        chart1.Series.RemoveAt(0);
```

74 % No issues found Ln: 22 Ch: 9 SPC CRLF

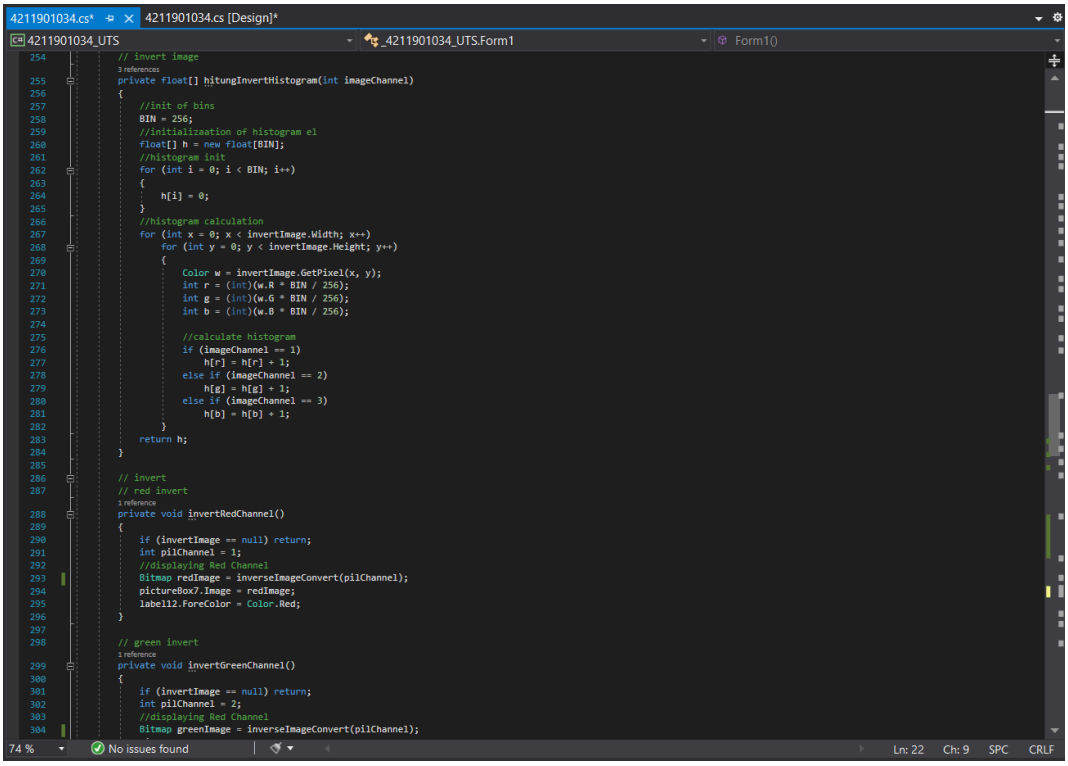


```
        chart1.Series.RemoveAt(0);
    }
    //chart init
    chart1.Series.Add("Red Channel Image");
    chart1.Series["Red Channel Image"].Color = Color.Red;
    foreach (var series in chart1.Series)
    {
        series.Points.Clear();
    }
    float[] his = new float[BIN];
    his = hitungHistogram(pilChannel);
    for (int i = 0; i < BIN; i++)
    {
        chart1.Series["Red Channel Image"].Points.AddXY(i, his[i]);
    }
    label9.ForeColor = Color.Red;
}

1 reference
private void greenChannelHistogram()
{
    if (sourceImage == null) return;
    int pilChannel = 2;
    //delete the histogram
    if (chart2.Series.Count > 0)
    {
        chart2.Series.RemoveAt(0);
    }
    //chart init
    chart2.Series.Add("Green Channel Image");
    chart2.Series["Green Channel Image"].Color = Color.Green;
    foreach (var series in chart2.Series)
    {
        series.Points.Clear();
    }
    float[] his = new float[BIN];
    his = hitungHistogram(pilChannel);
    for (int i = 0; i < BIN; i++)
    {
        chart2.Series["Green Channel Image"].Points.AddXY(i, his[i]);
    }
    label8.ForeColor = Color.Green;
}
1 reference
private void blueChannelHistogram()
{
    if (sourceImage == null) return;
    int pilChannel = 3;
    //delete the histogram
    if (chart3.Series.Count > 0)
    {
        chart3.Series.RemoveAt(0);
    }
    //chart init
```

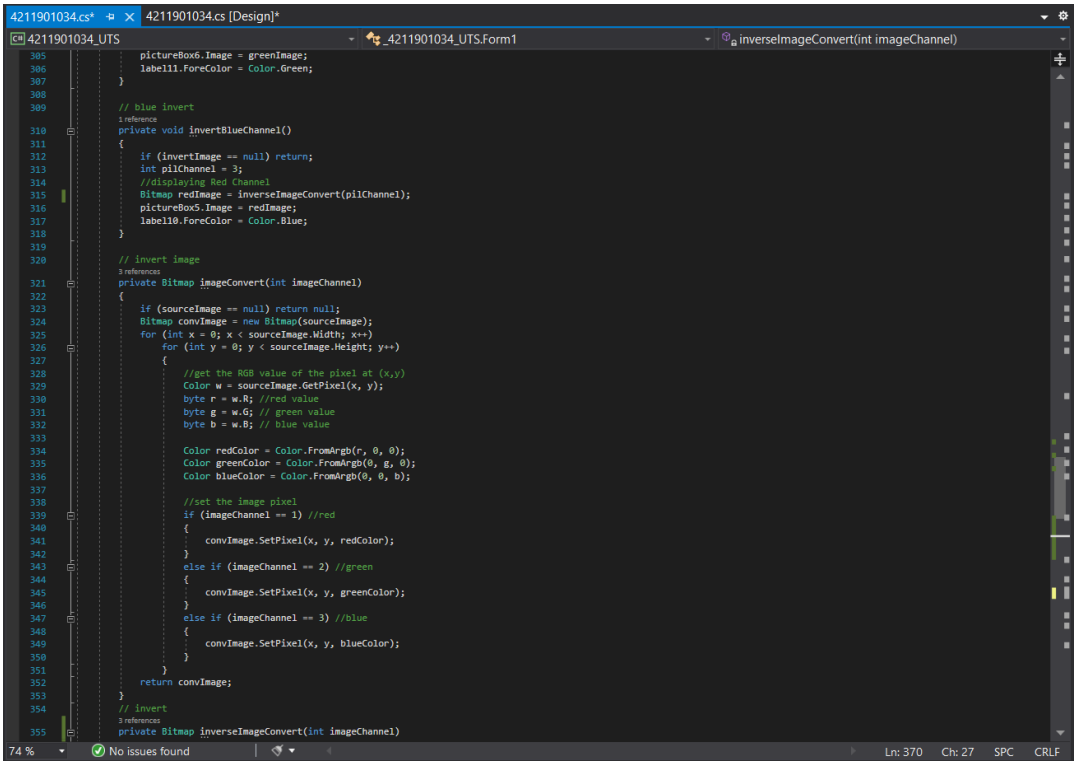
74 % No issues found Ln: 22 Ch: 9 SPC CRLF



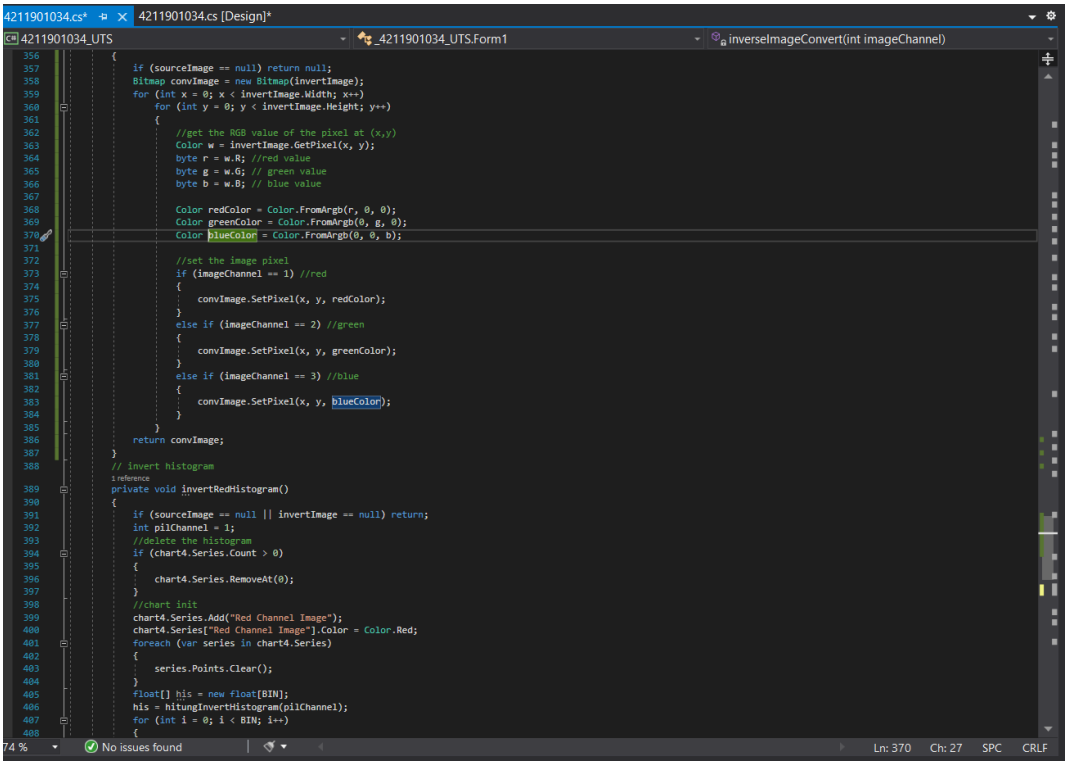
```
4211901034.cs* | 4211901034.cs [Design]*
4211901034_UTS | _4211901034_UTS.Form1 | Form1()

        //chart init
        chart3.Series.Add("Blue Channel Image");
        chart3.Series["Blue Channel Image"].Color = Color.Blue;
        foreach (var series in chart3.Series)
        {
            series.Points.Clear();
        }
        float[] his = new float[BIN];
        his = hitungHistogram(pilChannel);
        for (int i = 0; i < BIN; i++)
        {
            chart3.Series["Blue Channel Image"].Points.AddXY(i, his[i]);
        }
        label17.ForeColor = Color.Blue;
    }

    // hitung histogram
    1 reference
    private float[] hitungHistogram(int imageChannel)
    {
        //init of bins
        BIN = 256;
        //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;
    }

    // histogram

74 %  No issues found  Ln: 22  Ch: 9  SPC  CRLF
```



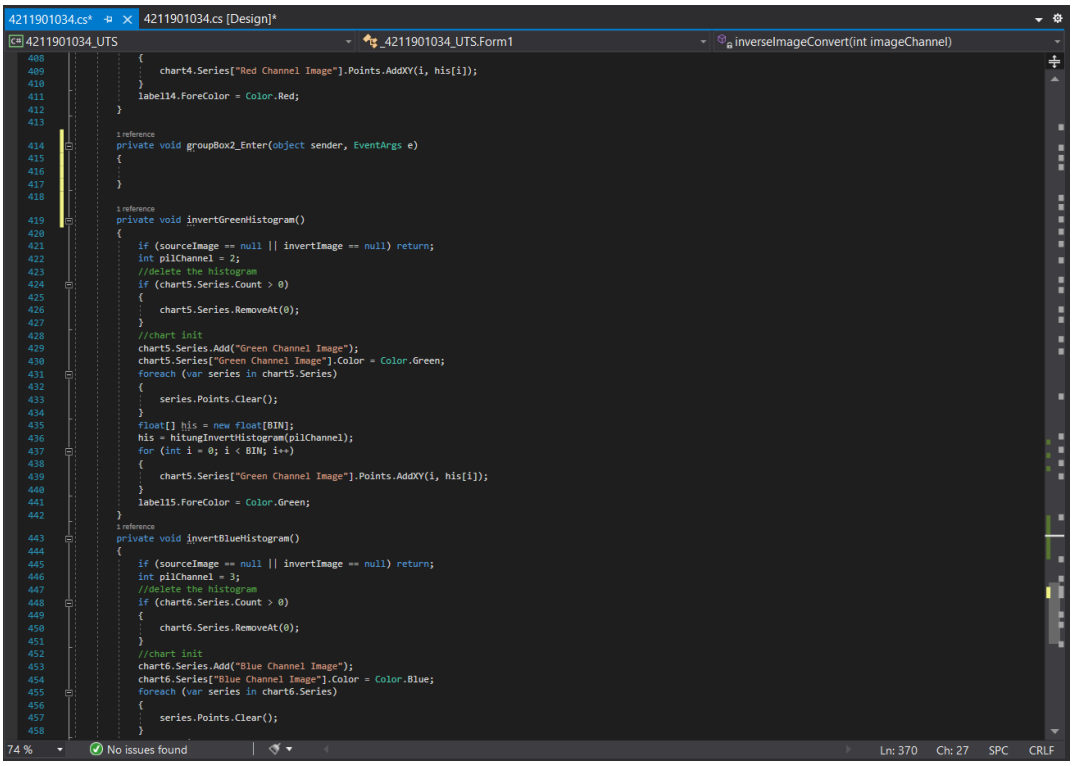
```
4211901034.cs* | 4211901034.cs [Design]*
4211901034_UTS | _4211901034_UTS.Form1 | Form1()

    // invert image
    3 references
    private float[] hitungInvertHistogram(int imageChannel)
    {
        //init of bins
        BIN = 256;
        //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 < invertImage.Width; x++)
            for (int y = 0; y < invertImage.Height; y++)
            {
                Color w = invertImage.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
                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;
            }
        return h;
    }

    // invert
    // red invert
    1 reference
    private void invertRedChannel()
    {
        if (invertImage == null) return;
        int pilChannel = 1;
        //displaying Red Channel
        Bitmap redImage = inverseImageConvert(pilChannel);
        pictureBox7.Image = redImage;
        label12.ForeColor = Color.Red;
    }

    // green invert
    1 reference
    private void invertGreenChannel()
    {
        if (invertImage == null) return;
        int pilChannel = 2;
        //displaying Red Channel
        Bitmap greenImage = inverseImageConvert(pilChannel);

74 %  No issues found  Ln: 22  Ch: 9  SPC  CRLF
```

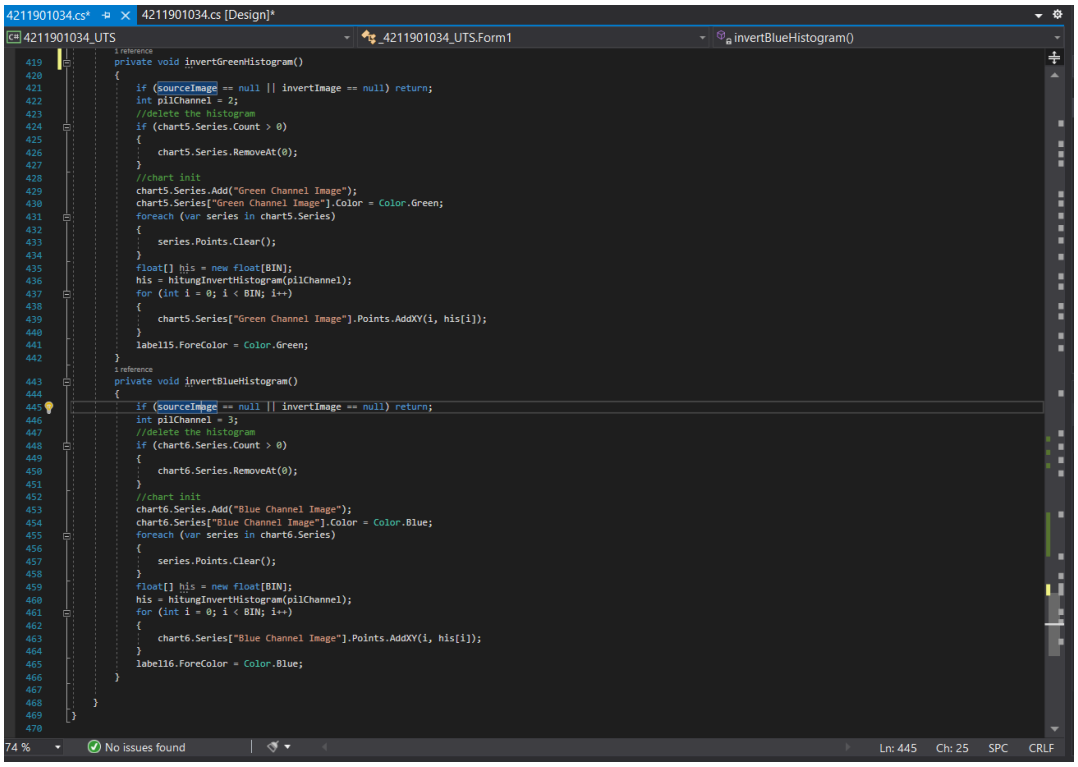


```
4211901034.cs* | 4211901034.cs [Design]*
4211901034_UTS | _4211901034_UTS.Form1 | inverseImageConvert(int imageChannel)

305             pictureBox6.Image = greenImage;
306             label11.ForeColor = Color.Green;
307         }
308
309         // blue invert
            1 reference
310         private void invertBlueChannel()
311         {
312             if (invertImage == null) return;
313             int pilChannel = 3;
314             //displaying Red Channel
315             Bitmap redImage = inverseImageConvert(pilChannel);
316             pictureBox5.Image = redImage;
317             label10.ForeColor = Color.Blue;
318         }
319
320         // invert image
            3 references
321         private Bitmap imageConvert(int imageChannel)
322         {
323             if (sourceImage == null) return null;
324             Bitmap convImage = new Bitmap(sourceImage);
325             for (int x = 0; x < sourceImage.Width; x++)
326                 for (int y = 0; y < sourceImage.Height; y++)
327                 {
328                     //get the RGB value of the pixel at (x,y)
329                     Color w = sourceImage.GetPixel(x, y);
330                     byte r = w.R; //red value
331                     byte g = w.G; // green value
332                     byte b = w.B; // blue value
333
334                     Color redColor = Color.FromArgb(r, 0, 0);
335                     Color greenColor = Color.FromArgb(0, g, 0);
336                     Color blueColor = Color.FromArgb(0, 0, b);
337
338                     //set the image pixel
339                     if (imageChannel == 1) //red
340                     {
341                         convImage.SetPixel(x, y, redColor);
342                     }
343                     else if (imageChannel == 2) //green
344                     {
345                         convImage.SetPixel(x, y, greenColor);
346                     }
347                     else if (imageChannel == 3) //blue
348                     {
349                         convImage.SetPixel(x, y, blueColor);
350                     }
351                 }
352             return convImage;
353         }
354         // invert
            3 references
355         private Bitmap inverseImageConvert(int imageChannel)

74 %  No issues found  Ln: 370  Ch: 27  SPC  CRLF
```

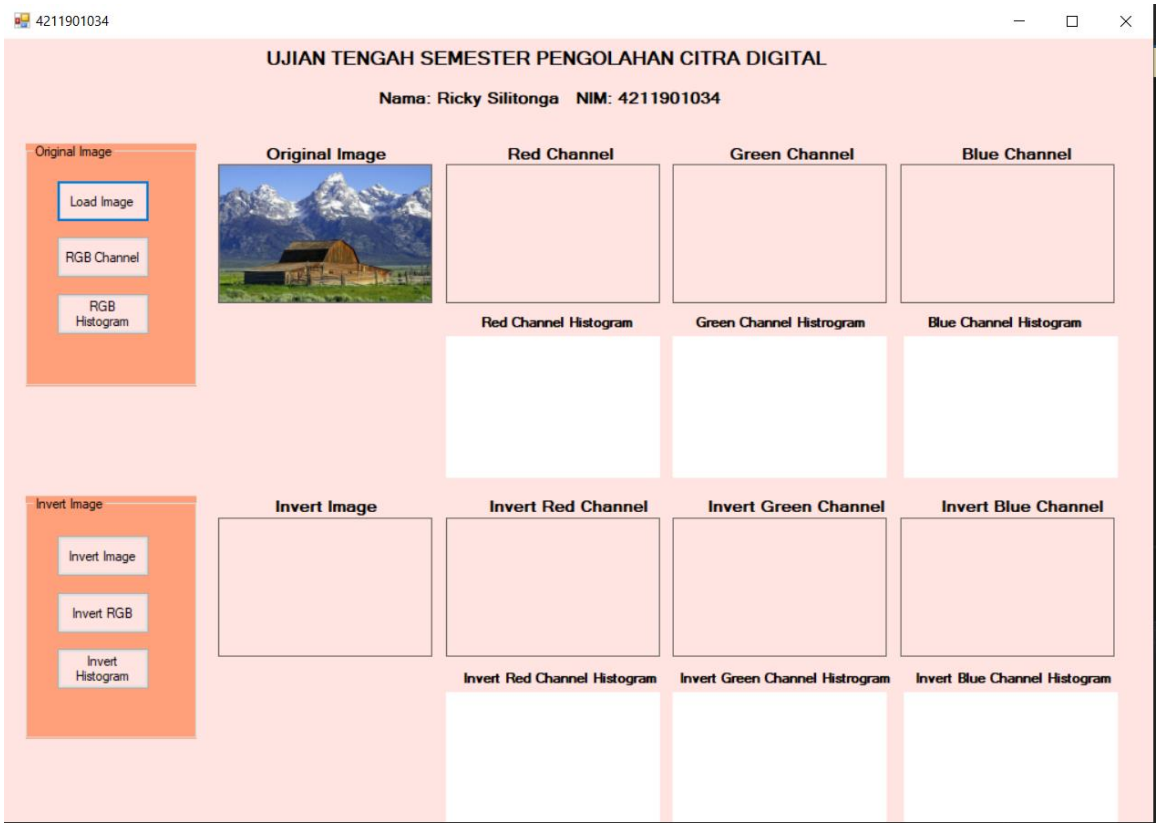


```
4211901034.cs* | 4211901034.cs [Design]*
4211901034_UTS | _4211901034_UTS.Form1 | inverseImageConvert(int imageChannel)

356         {
357             if (sourceImage == null) return null;
358             Bitmap convImage = new Bitmap(invertImage);
359             for (int x = 0; x < invertImage.Width; x++)
360                 for (int y = 0; y < invertImage.Height; y++)
361                 {
362                     //get the RGB value of the pixel at (x,y)
363                     Color w = invertImage.GetPixel(x, y);
364                     byte r = w.R; //red value
365                     byte g = w.G; // green value
366                     byte b = w.B; // blue value
367
368                     Color redColor = Color.FromArgb(r, 0, 0);
369                     Color greenColor = Color.FromArgb(0, g, 0);
370                     Color blueColor = Color.FromArgb(0, 0, b);
371
372                     //set the image pixel
373                     if (imageChannel == 1) //red
374                     {
375                         convImage.SetPixel(x, y, redColor);
376                     }
377                     else if (imageChannel == 2) //green
378                     {
379                         convImage.SetPixel(x, y, greenColor);
380                     }
381                     else if (imageChannel == 3) //blue
382                     {
383                         convImage.SetPixel(x, y, blueColor);
384                     }
385                 }
386             return convImage;
387         }
388         // invert histogram
            1 reference
389         private void invertRedHistogram()
390         {
391             if (sourceImage == null || invertImage == null) return;
392             int pilChannel = 1;
393             //delete the histogram
394             if (chart4.Series.Count > 0)
395             {
396                 chart4.Series.RemoveAt(0);
397             }
398             //chart init
399             chart4.Series.Add("Red Channel Image");
400             chart4.Series["Red Channel Image"].Color = Color.Red;
401             foreach (var series in chart4.Series)
402             {
403                 series.Points.Clear();
404             }
405             float[] his = new float[BIN];
406             his = hitungInvertHistogram(pilChannel);
407             for (int i = 0; i < BIN; i++)
408             {

74 %  No issues found  Ln: 370  Ch: 27  SPC  CRLF
```

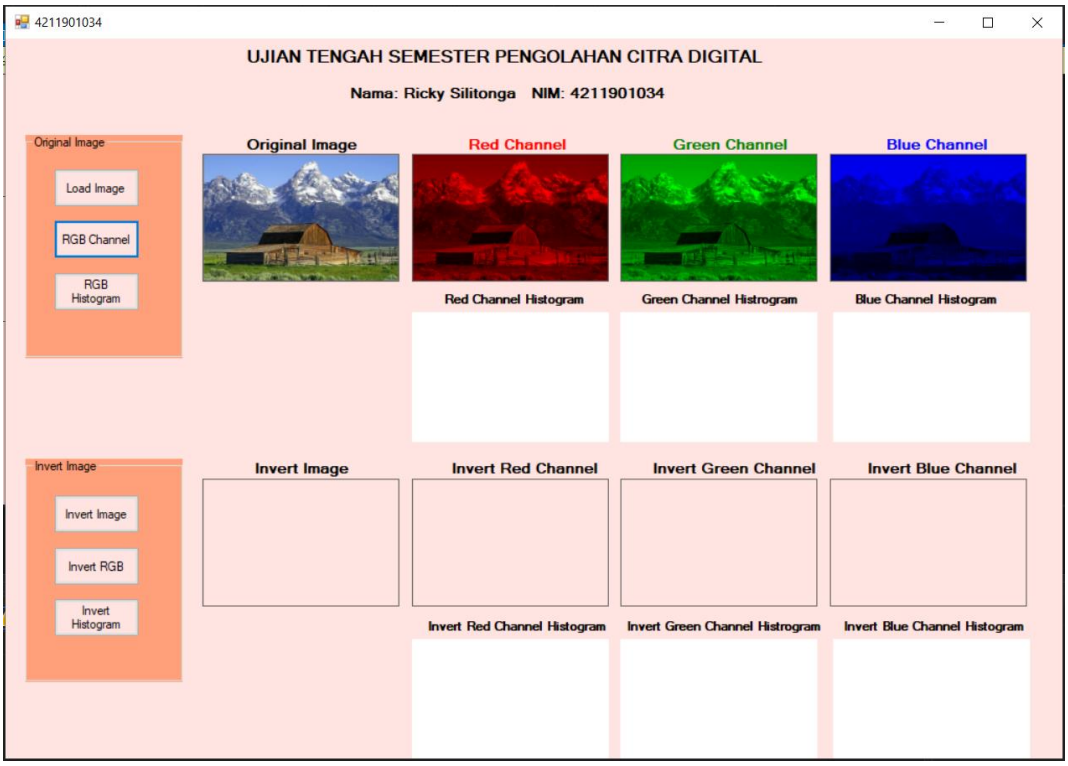

4211901034.cs*
4211901034.cs [Design]*
4211901034_UTS
_4211901034_UTS.Form1
inverseImageConvert(int imageChannel)
{
chart4.Series["Red Channel Image"].Points.AddXY(i, his[i]);
}
label14.ForeColor = Color.Red;
}
1 reference
private void groupBox2_Enter(object sender, EventArgs e)
{
}
1 reference
private void invertGreenHistogram()
{
if (sourceImage == null || invertImage == null) return;
int pilChannel = 2;
//delete the histogram
if (chart5.Series.Count > 0)
{
chart5.Series.RemoveAt(0);
}
//chart init
chart5.Series.Add("Green Channel Image");
chart5.Series["Green Channel Image"].Color = Color.Green;
foreach (var series in chart5.Series)
{
series.Points.Clear();
}
float[] his = new float[BIN];
his = hitungInvertHistogram(pilChannel);
for (int i = 0; i < BIN; i++)
{
chart5.Series["Green Channel Image"].Points.AddXY(i, his[i]);
}
label15.ForeColor = Color.Green;
}
1 reference
private void invertBlueHistogram()
{
if (sourceImage == null || invertImage == null) return;
int pilChannel = 3;
//delete the histogram
if (chart6.Series.Count > 0)
{
chart6.Series.RemoveAt(0);
}
//chart init
chart6.Series.Add("Blue Channel Image");
chart6.Series["Blue Channel Image"].Color = Color.Blue;
foreach (var series in chart6.Series)
{
series.Points.Clear();
}
74 %
No issues found
Ln: 370
Ch: 27
SPC
CRLF


4211901034.cs*
4211901034.cs [Design]*
4211901034_UTS
_4211901034_UTS.Form1
invertBlueHistogram()
1 reference
private void invertGreenHistogram()
{
if (sourceImage == null || invertImage == null) return;
int pilChannel = 2;
//delete the histogram
if (chart5.Series.Count > 0)
{
chart5.Series.RemoveAt(0);
}
//chart init
chart5.Series.Add("Green Channel Image");
chart5.Series["Green Channel Image"].Color = Color.Green;
foreach (var series in chart5.Series)
{
series.Points.Clear();
}
float[] his = new float[BIN];
his = hitungInvertHistogram(pilChannel);
for (int i = 0; i < BIN; i++)
{
chart5.Series["Green Channel Image"].Points.AddXY(i, his[i]);
}
label15.ForeColor = Color.Green;
}
1 reference
private void invertBlueHistogram()
{
if (sourceImage == null || invertImage == null) return;
int pilChannel = 3;
//delete the histogram
if (chart6.Series.Count > 0)
{
chart6.Series.RemoveAt(0);
}
//chart init
chart6.Series.Add("Blue Channel Image");
chart6.Series["Blue Channel Image"].Color = Color.Blue;
foreach (var series in chart6.Series)
{
series.Points.Clear();
}
float[] his = new float[BIN];
his = hitungInvertHistogram(pilChannel);
for (int i = 0; i < BIN; i++)
{
chart6.Series["Blue Channel Image"].Points.AddXY(i, his[i]);
}
label16.ForeColor = Color.Blue;
}
}
}
74 %
No issues found
Ln: 445
Ch: 25
SPC
CRLF

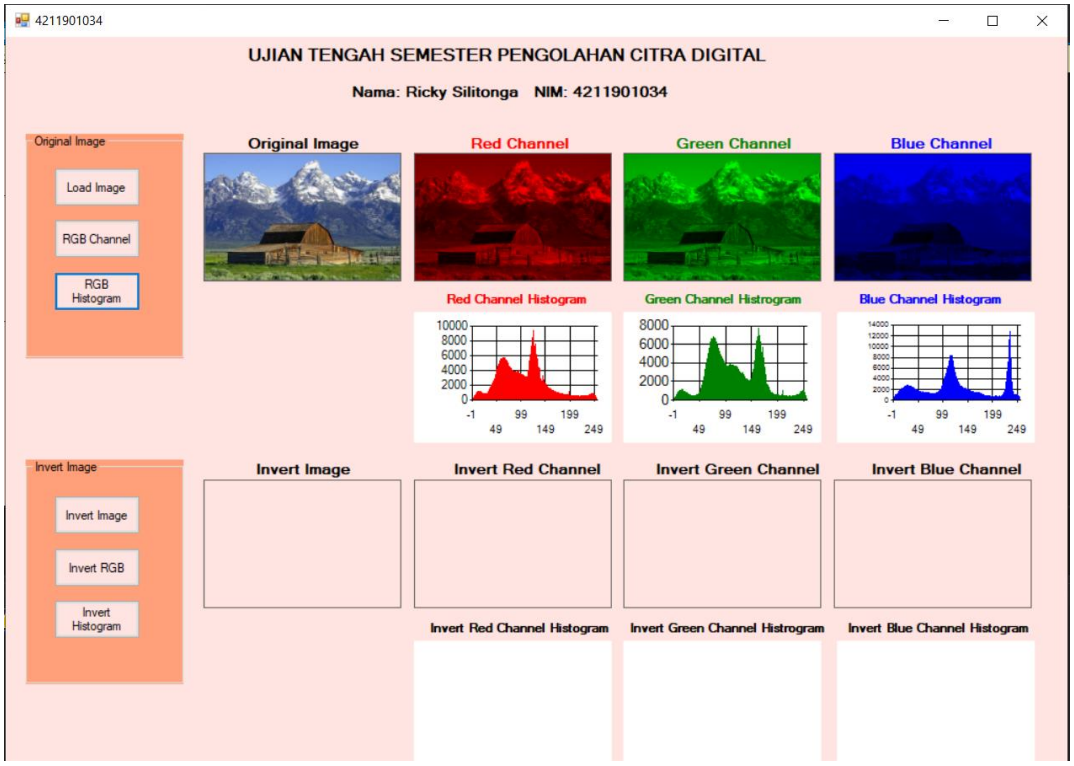
3. Screen shoot semua hasil operasi yang dilakukan semua tombol yang ada



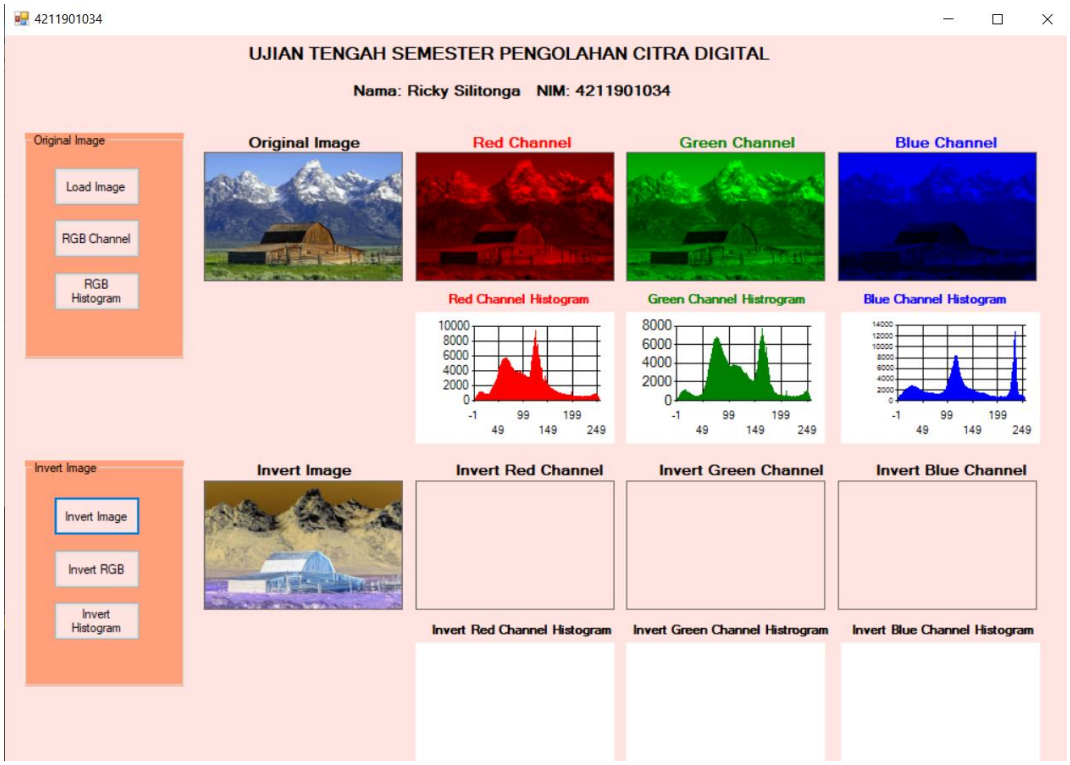
Load Image berhasil



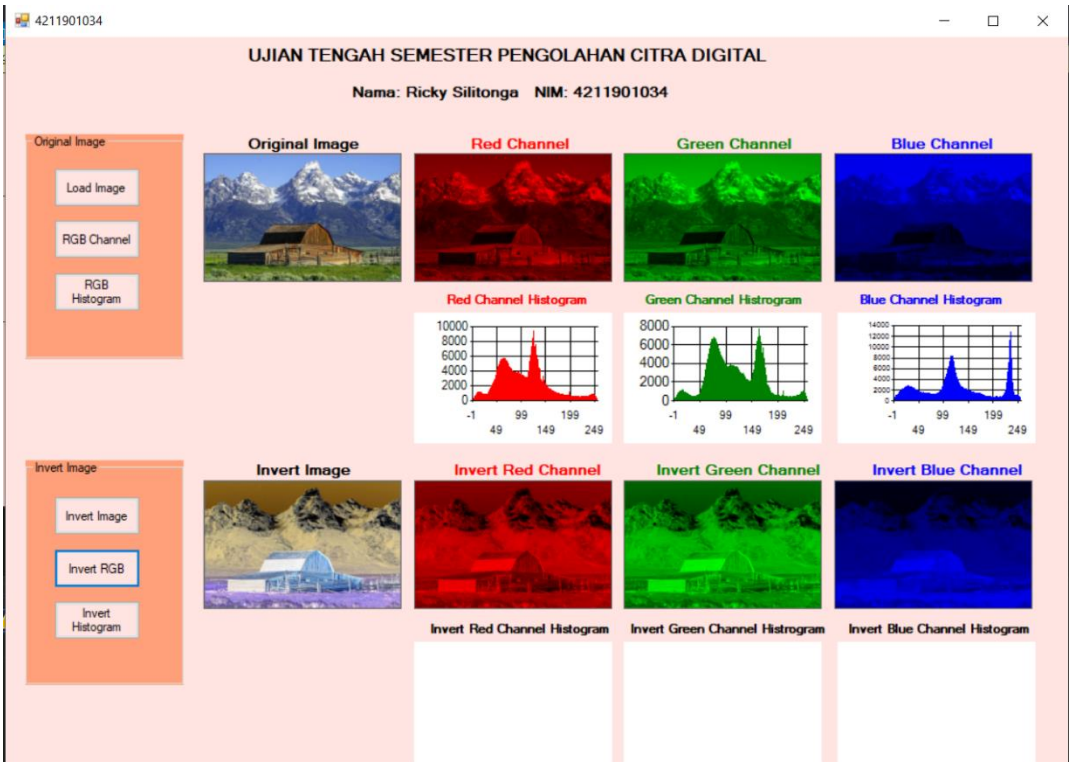
RGB Channel di klik



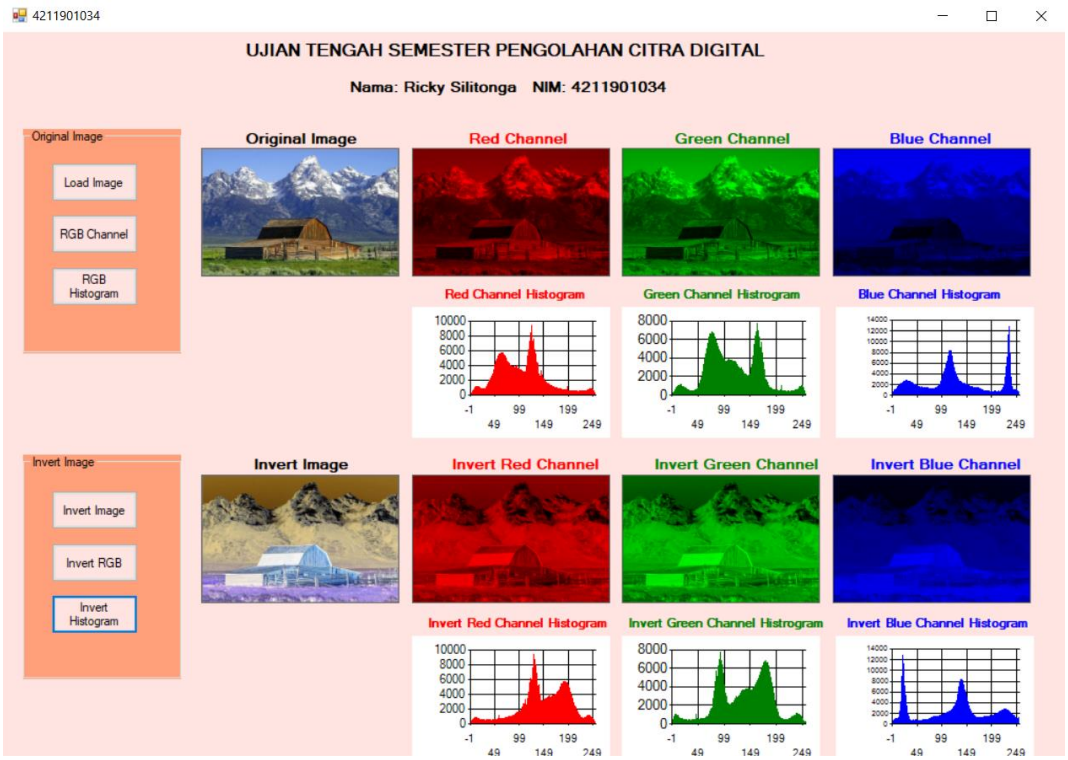
RGB Histogram di klik



Invert Image di klik



Invert RGB di klik



Invert Histogram di klik