

**UJIAN TENGAH SEMESTER
PENGOLAHAN CITRA
MK401**



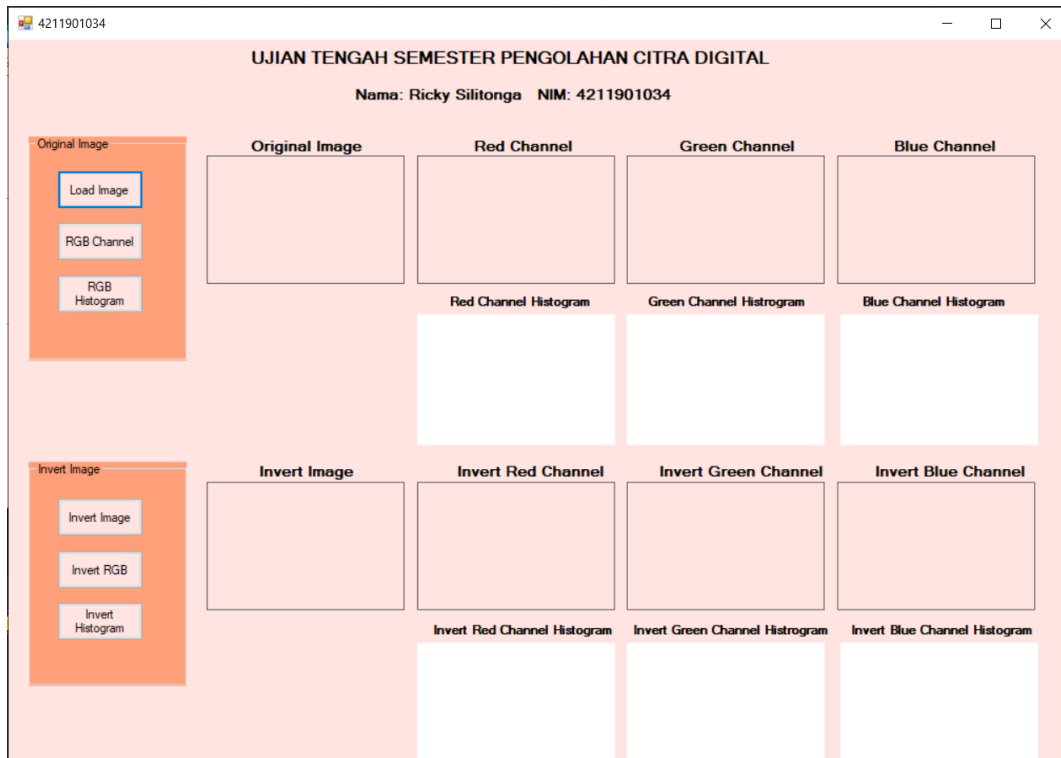
POLITEKNIK NEGERI Batam

**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

```
4211901034.cs* x 4211901034 [Design]*
4211901034_UTS - 4211901034_UTS.Form1 Form10
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.Drawing.Imaging;
11 using System.IO;
12
13 namespace _4211901034_UTS
14 {
15     3 references
16     public partial class Form1 : Form
17     {
18         Bitmap sourceImage, invertImage;
19         int image_height, image_width;
20         int BIN = 256;
21
22
23     1 reference
24     public Form1()
25     {
26         InitializeComponent();
27
28     1 reference
29     private void Form1_Load(object sender, EventArgs e)
30     {
31
32     }
33
34     // tombol load image
35     1 reference
36     private void button1_Click(object sender, EventArgs e)
37     {
38         if (openFileDialog1.ShowDialog() == DialogResult.OK)
39         {
40             sourceImage = (Bitmap)Bitmap.FromFile(openFileDialog1.FileName);
41             pictureBox1.Image = sourceImage;
42             invertImage = new Bitmap(sourceImage);
43             //delete the histogram
44             BIN = 255;
45
46             // width and height
47             image_width = sourceImage.Width;
48             image_height = sourceImage.Height;
49         }
50
51     }
52
53     // tombol RGB Channel
```

```
51 private void button2_Click(object sender, EventArgs e)
52 {
53     redImageChannel();
54     greenImageChannel();
55     blueImageChannel();
56 }
57
58 // tombol RGB Histogram
59 1 reference
60 private void button3_Click(object sender, EventArgs e)
61 {
62     redChannelHistogram();
63     greenChannelHistogram();
64     blueChannelHistogram();
65 }
66
67 // tombol invert image
68 1 reference
69 private void button4_Click(object sender, EventArgs e)
70 {
71     if (sourceImage == null) return;
72     for (int x = 0; x < image_width; x++)
73     {
74         for (int y = 0; y < image_height; y++)
75         {
76             // get rgb value of the pixel at (x, y)
77             Color w = sourceImage.GetPixel(x, y);
78
79             // Invers image
80             int rInverse = 255 - w.R;
81             int gInverse = 255 - w.G;
82             int bInverse = 255 - w.B;
83
84             Color inverse_color = Color.FromArgb(rInverse, gInverse, bInverse);
85             invertImage.SetPixel(x, y, inverse_color);
86         }
87     }
88     pictureBox8.Image = invertImage;
89 }
90
91 // tombol invert RGB
92 1 reference
93 private void button5_Click(object sender, EventArgs e)
94 {
95     invertRedChannel();
96     invertGreenChannel();
97     invertBlueChannel();
98 }
99
100 // tombol invert histogram
101 1 reference
102 private void button6_Click(object sender, EventArgs e)
```

```
4211901034.cs* x 4211901034 [Design]*
4211901034.UTS - 4211901034.UTS.Form1 - Form10
101 {
102     InvertRedHistogram();
103     InvertGreenHistogram();
104     InvertBlueHistogram();
105 }
106
107 // my function
108 // red channel
109 1 reference
110 private void redImageChannel()
111 {
112     if (sourceImage == null) return;
113     int pilChannel = 1;
114     //displaying Red Channel
115     Bitmap redImage = imageConvert(pilChannel);
116     pictureBox2.Image = redImage;
117     label4.ForeColor = Color.Red;
118 }
119 // green channel
120 1 reference
121 private void greenImageChannel()
122 {
123     if (sourceImage == null) return;
124     int pilChannel = 2;
125     //displaying Red Channel
126     Bitmap greenImage = imageConvert(pilChannel);
127     pictureBox3.Image = greenImage;
128     label5.ForeColor = Color.Green;
129 }
130 // blue image
131 1 reference
132 private void blueImageChannel()
133 {
134     if (sourceImage == null) return;
135     int pilChannel = 3;
136     //displaying Red Channel
137     Bitmap blueImage = imageConvert(pilChannel);
138     pictureBox4.Image = blueImage;
139     label6.ForeColor = Color.Blue;
140 }
141 // histogram
142 // red channel histogram
143 1 reference
144 private void redChannelHistogram()
145 {
146     if (sourceImage == null) return;
147     int pilChannel = 1;
148     //delete the histogram
149     if (chart1.Series.Count > 0)
150     {
151         chart1.Series.RemoveAt(0);
152     }
153 }
```

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

```
4211901034.cs* x 4211901034 [Design]*
4211901034.UTS - 4211901034.UTS.Form1 - Form10
150 chart1.Series.RemoveAt(0);
151 }
152 //chart init
153 chart1.Series.Add("Red Channel Image");
154 chart1.Series["Red Channel Image"].Color = Color.Red;
155 foreach (var series in chart1.Series)
156 {
157     series.Points.Clear();
158 }
159 float[] his = new float[BIN];
160 his = hitungHistogram(pilChannel);
161 for (int i = 0; i < BIN; i++)
162 {
163     chart1.Series["Red Channel Image"].Points.AddXY(i, his[i]);
164 }
165 label9.ForeColor = Color.Red;
166 }
167
168 1 reference
169 private void greenChannelHistogram()
170 {
171     if (sourceImage == null) return;
172     int pilChannel = 2;
173     //delete the histogram
174     if (chart2.Series.Count > 0)
175     {
176         chart2.Series.RemoveAt(0);
177     }
178     //chart init
179     chart2.Series.Add("Green Channel Image");
180     chart2.Series["Green Channel Image"].Color = Color.Green;
181     foreach (var series in chart2.Series)
182     {
183         series.Points.Clear();
184     }
185     float[] his = new float[BIN];
186     his = hitungHistogram(pilChannel);
187     for (int i = 0; i < BIN; i++)
188     {
189         chart2.Series["Green Channel Image"].Points.AddXY(i, his[i]);
190     }
191     label8.ForeColor = Color.Green;
192 }
193 1 reference
194 private void blueChannelHistogram()
195 {
196     if (sourceImage == null) return;
197     int pilChannel = 3;
198     //delete the histogram
199     if (chart3.Series.Count > 0)
200     {
201         chart3.Series.RemoveAt(0);
202     }
203     //chart init
204     chart3.Series.Add("Blue Channel Image");
205     chart3.Series["Blue Channel Image"].Color = Color.Blue;
206     foreach (var series in chart3.Series)
207     {
208         series.Points.Clear();
209     }
210     float[] his = new float[BIN];
211     his = hitungHistogram(pilChannel);
212     for (int i = 0; i < BIN; i++)
213     {
214         chart3.Series["Blue Channel Image"].Points.AddXY(i, his[i]);
215     }
216     label7.ForeColor = Color.Blue;
217 }
```

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

```
4211901034.cs* x 4211901034.cs [Design]*
4211901034_UTS - 4211901034_UTS.Form1 - Form10

201 //chart init
202 chart3.Series.Add("Blue Channel Image");
203 chart3.Series["Blue Channel Image"].Color = Color.Blue;
204 foreach (var series in chart3.Series)
205 {
206     series.Points.Clear();
207 }
208 float[] his = new float[BIN];
209 his = hitungHistogram(pilChannel);
210 for (int i = 0; i < BIN; i++)
211 {
212     chart3.Series["Blue Channel Image"].Points.AddXY(i, his[i]);
213 }
214 label7.ForeColor = Color.Blue;
215 }
216
217 // hitung histogram
218 3 references
219 private float[] hitungHistogram(int imageChannel)
220 {
221     //init of bins
222     BIN = 256;
223     //initialization of histogram el
224     float[] h = new float[BIN];
225     //histogram init
226     for (int i = 0; i < BIN; i++)
227     {
228         h[i] = 0;
229     }
230     //histogram calculation
231     for (int x = 0; x < sourceImage.Width; x++)
232     for (int y = 0; y < sourceImage.Height; y++)
233     {
234         Color w = sourceImage.GetPixel(x, y);
235         int r = (int)(w.R * BIN / 256);
236         int g = (int)(w.G * BIN / 256);
237         int b = (int)(w.B * BIN / 256);
238         //calculate gray channel
239         int gray = (int)((0.5 * w.R + 0.419 * w.G + 0.081 * w.B) * BIN / 256);
240         //calculate histogram
241         if (imageChannel == 1)
242             h[r] = h[r] + 1;
243         else if (imageChannel == 2)
244             h[g] = h[g] + 1;
245         else if (imageChannel == 3)
246             h[b] = h[b] + 1;
247         else if (imageChannel == 4)
248             h[gray] = h[gray] + 1;
249     }
250     return h;
251 }
252 // histogram
253
74 % No issues found Ln: 22 Ch: 9 SPC CRLF
```

```
4211901034.cs* x 4211901034.cs [Design]*
4211901034_UTS - 4211901034_UTS.Form1 - Form10

254 // invert image
255 2 references
256 private float[] hitungInverHistogram(int imageChannel)
257 {
258     //init of bins
259     BIN = 256;
260     //initialization of histogram el
261     float[] h = new float[BIN];
262     //histogram init
263     for (int i = 0; i < BIN; i++)
264     {
265         h[i] = 0;
266     }
267     //histogram calculation
268     for (int x = 0; x < invertImage.Width; x++)
269     for (int y = 0; y < invertImage.Height; y++)
270     {
271         Color w = invertImage.GetPixel(x, y);
272         int r = (int)(w.R * BIN / 256);
273         int g = (int)(w.G * BIN / 256);
274         int b = (int)(w.B * BIN / 256);
275         //calculate histogram
276         if (imageChannel == 1)
277             h[r] = h[r] + 1;
278         else if (imageChannel == 2)
279             h[g] = h[g] + 1;
280         else if (imageChannel == 3)
281             h[b] = h[b] + 1;
282     }
283     return h;
284 }
285
286 // invert
287 // red invert
288 1 reference
289 private void invertRedChannel()
290 {
291     if (invertImage == null) return;
292     int pilChannel = 1;
293     //displaying Red Channel
294     Bitmap redImage = InverseImageConvert(pilChannel);
295     pictureBox7.Image = redImage;
296     label12.ForeColor = Color.Red;
297 }
298
299 // green invert
300 1 reference
301 private void invertGreenChannel()
302 {
303     if (invertImage == null) return;
304     int pilChannel = 2;
305     //displaying Red Channel
306     Bitmap greenImage = InverseImageConvert(pilChannel);
307 }
```

```
4211901034.cs* x 4211901034.cs [Design]*
4211901034_UTS 4211901034_UTS.Form1 inverselImageConvert(int imageChannel)
305 pictureBox6.Image = greenImage;
306 label11.ForeColor = Color.Green;
307
308
309 // blue invert
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
321 3 references
322 private Bitmap imageConvert(int imageChannel)
323 {
324     if (sourceImage == null) return null;
325     Bitmap convImage = new Bitmap(sourceImage);
326     for (int x = 0; x < sourceImage.Width; x++)
327         for (int y = 0; y < sourceImage.Height; y++)
328         {
329             //get the RGB value of the pixel at (x,y)
330             Color w = sourceImage.GetPixel(x, y);
331             byte r = w.R; //red value
332             byte g = w.G; //green value
333             byte b = w.B; //blue value
334
335             Color redColor = Color.FromArgb(r, 0, 0);
336             Color greenColor = Color.FromArgb(0, g, 0);
337             Color blueColor = Color.FromArgb(0, 0, b);
338
339             //set the image pixel
340             if (imageChannel == 1) //red
341             {
342                 convImage.SetPixel(x, y, redColor);
343             }
344             else if (imageChannel == 2) //green
345             {
346                 convImage.SetPixel(x, y, greenColor);
347             }
348             else if (imageChannel == 3) //blue
349             {
350                 convImage.SetPixel(x, y, blueColor);
351             }
352         }
353     return convImage;
354 }
355 // invert
356 3 references
357 private Bitmap inverseImageConvert(int imageChannel)
```

```
4211901034.cs* x 4211901034.cs [Design]*
4211901034_UTS 4211901034_UTS.Form1 inverselImageConvert(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
389 1 reference
390 private void invertRedHistogram()
391 {
392     if (sourceImage == null || invertImage == null) return;
393     int pilChannel = 1;
394     //delete the histogram
395     if (chart4.Series.Count > 0)
396     {
397         chart4.Series.RemoveAt(0);
398     }
399     //chart init
400     chart4.Series.Add("Red Channel Image");
401     chart4.Series["Red Channel Image"].Color = Color.Red;
402     foreach (var series in chart4.Series)
403     {
404         series.Points.Clear();
405     }
406     float[] his = new float[BIN];
407     his = hitungInverHistogram(pilChannel);
408     for (int i = 0; i < BIN; i++)
```

```
4211901034.cs* x 4211901034 [Design]*
4211901034_UTS
    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* x 4211901034 [Design]*
4211901034_UTS
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;
}

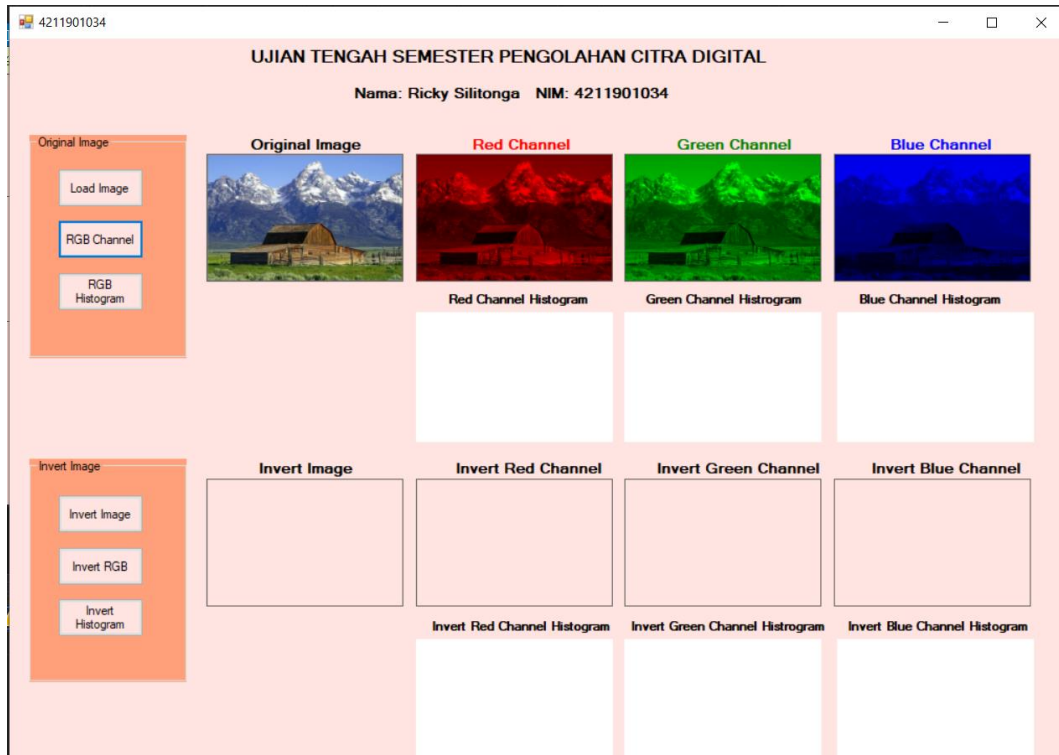
1 reference
private void invertBlueHistogram()
{
    if (sourceImage == null || invertImage == null) return;
    int pilChannel = 3;
    //delete the histogram
    if (chart6.Series.Count > 0)
    {
        chart6.Series.RemoveAt(0);
    }
    //chart init
    chart6.Series.Add("Blue Channel Image");
    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

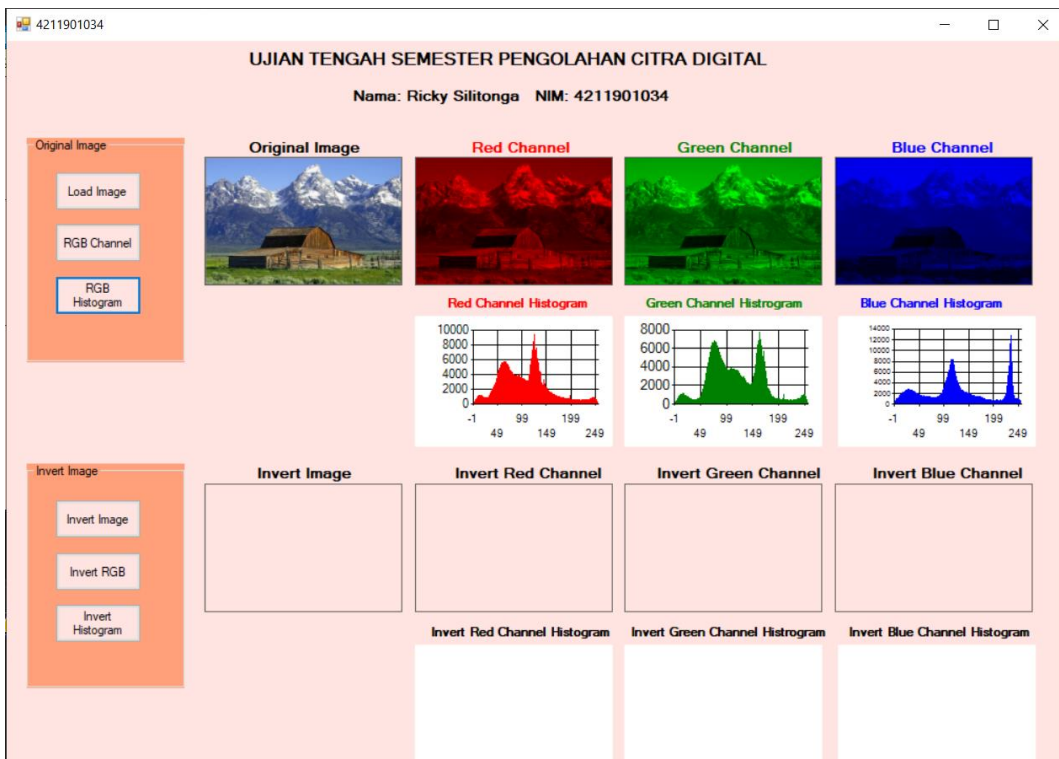
3. Screen shoot semua hasil operasi yang dilakukan semua tombol yang ada



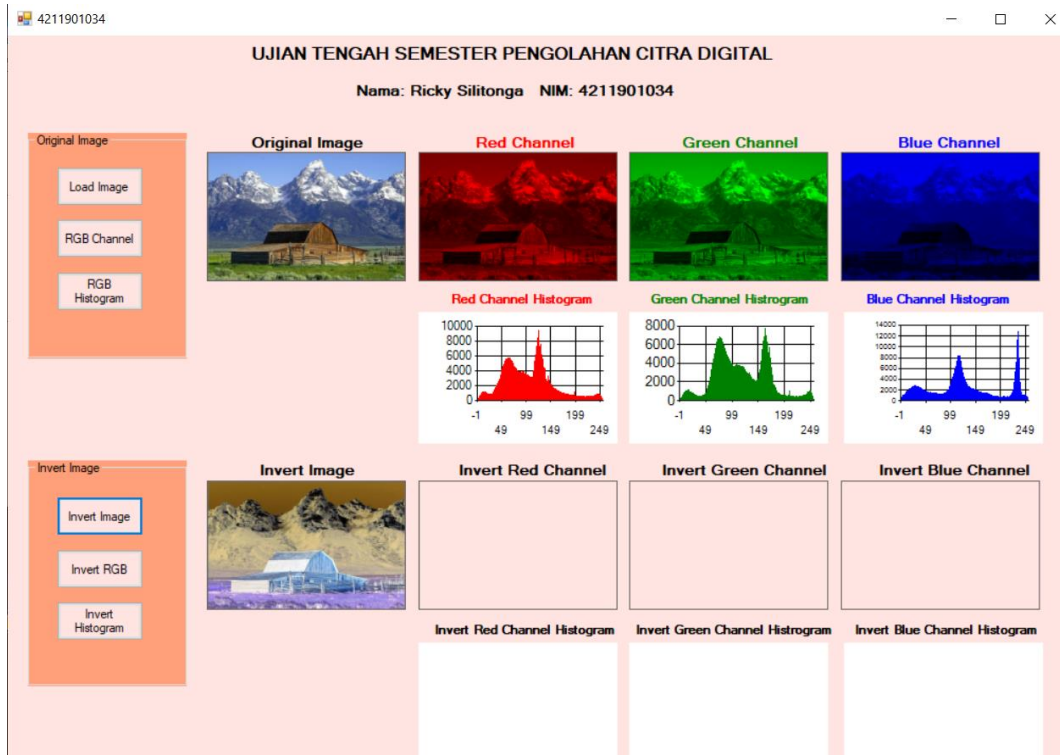
Load Image berhasil



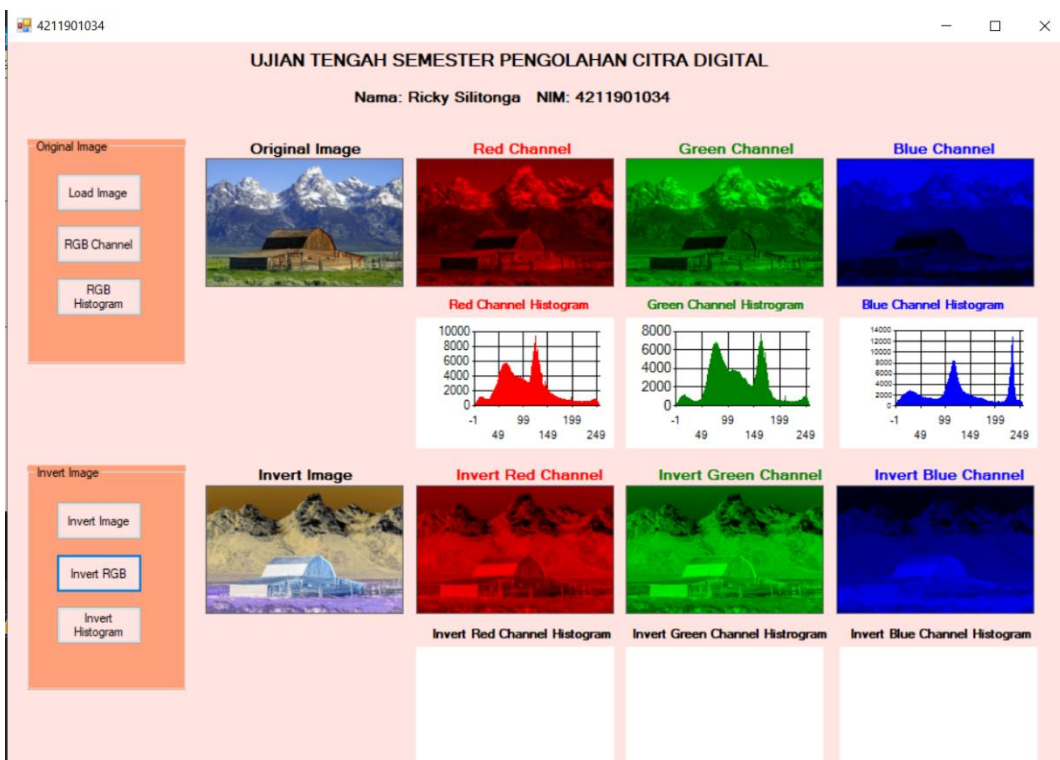
RGB Channel di klik



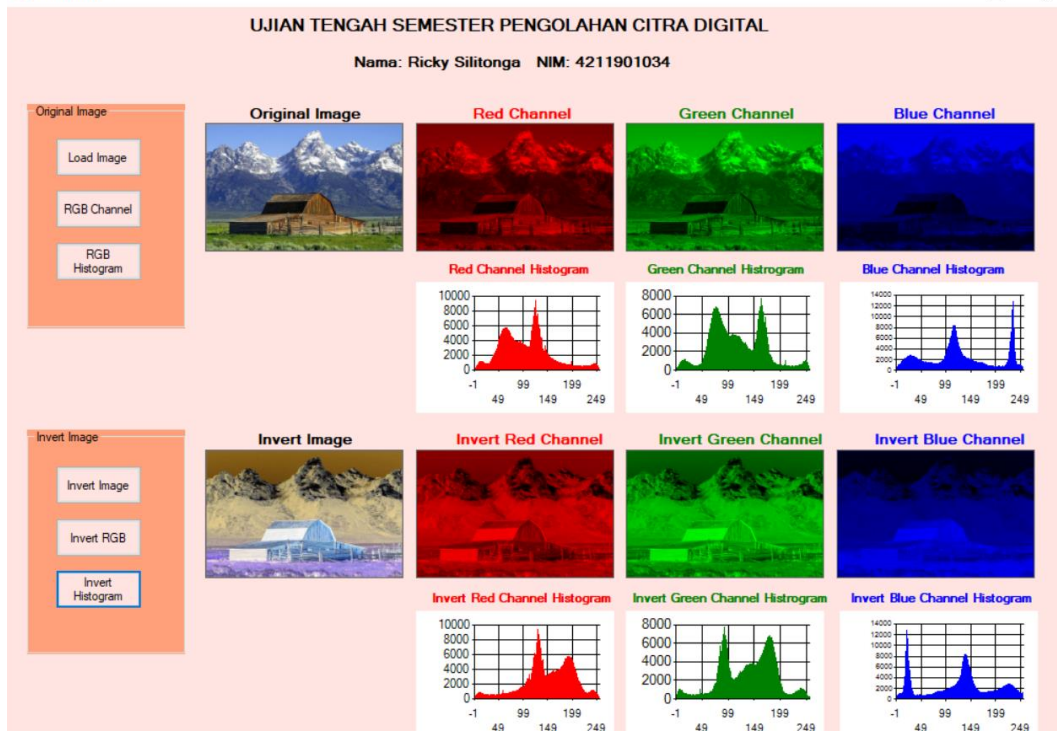
RGB Histogram di klik



Invert Image di klik



Invert RGB di klik



Invert Histogram di klik