

**LAPORAN PROJECT AKHIR
PENGOLAHAN CITRA DIGITAL**



KELOMPOK I7GEN13HX

IK22-A

1. Muhammad Rivaldi J. (221011041)
2. Muh. Abubakar T. (221011021)
3. Osama Iyad Al Ghozy (221011031)
4. Michael Yonathan (221011064)
5. Wildan An Millah M. (221011116)
6. Edwin Tangaran (221011118)

**PROGRAM STUDI ILMU KOMPUTER
JURUSAN TEKNOLOGI PRODUKSI DAN INDUSTRI
INSTITUT TEKNOLOGI BACHARUDDIN JUSUF HABIBIE PAREPARE
2024**

Ringkasan:

Program kami menggunakan bahasa pemrograman C#, Program ini adalah aplikasi sederhana untuk mengedit gambar menggunakan beberapa teknik dasar seperti pengaburan Gaussian, resize, konversi ke citra negatif, konversi ke citra biner, deteksi tepi (Canny Edge Detection), dan deteksi objek (Contour Detection) menggunakan EmguCV di Windows Forms.

Fitur Utama:

1. **Load Image:** Memuat gambar dari file dengan format .png, .jpg, .bmp, atau .jpeg ke dalam PictureBox pertama.
2. **Gaussian Blur:** Menggunakan kernel Gaussian 3x3 untuk mengurangi noise pada gambar.
3. **Resize Image:** Melakukan resize gambar menggunakan nearest neighbor interpolation.
4. **Negative Image:** Mengubah gambar menjadi citra negatif (mengubah warna menjadi negatif).
5. **Convert to Binary Image:** Mengonversi gambar menjadi citra biner berdasarkan nilai grayscale.
6. **Edge Detection (Canny):** Menggunakan algoritma Canny untuk mendeteksi tepi pada gambar.
7. **Object Detection (Contour):** Menggunakan teknik deteksi kontur untuk menemukan objek dalam gambar.
8. **Clear Image:** Membersihkan PictureBox kedua untuk menampilkan gambar baru atau menghapus gambar yang sedang ditampilkan.

Komponen UI:

- **PictureBox 1:** Menampilkan gambar asli yang dimuat atau diproses.
- **PictureBox 2:** Menampilkan hasil pengolahan gambar sesuai dengan tombol yang ditekan.

Teknologi(Template/Library):

- **EmguCV:** Digunakan untuk operasi pengolahan citra seperti Gaussian Blur, Canny Edge Detection, dan Contour Detection.
- **Windows Forms (C#):** Digunakan sebagai platform untuk membangun antarmuka pengguna dan mengintegrasikan operasi pengolahan citra.

Penggunaan:

- Pengguna dapat memuat gambar dari file.
- Pengguna dapat memilih berbagai operasi pengolahan citra yang tersedia.
- Hasil dari setiap operasi ditampilkan secara real-time di PictureBox kedua.

Kode Program:

- Library/Reference

```
1  using System;
2  using System.Drawing;
3  using System.Drawing.Imaging;
4  using System.Windows.Forms;
5  using Emgu.CV;
6  using Emgu.CV.CvEnum;
7  using Emgu.CV.OCR;
8  using Emgu.CV.Structure;
9  using Emgu.CV.Util;
10
```

- Method untuk mengupload gambar

```
20  private void button1_Click(object sender, EventArgs e)
21  {
22      OpenFileDialog ofd = new OpenFileDialog();
23      ofd.DefaultExt = "bmp";
24      ofd.Filter = "image files|*.png; *.jpg; *.bmp; *.jpeg;";
25      if (ofd.ShowDialog() == DialogResult.OK)
26      {
27          pictureBox1.Image = new Bitmap(ofd.FileName);
28          Bitmap bmp = new Bitmap(pictureBox1.Image);
29      }
30 }
```

- Method untuk mengaplikasikan Gaussian Blur

```
32  private Bitmap ApplyGaussianBlur(Bitmap image)
33  {
34      // Mendefinisikan kernel (3x3 kernel gaussian, sigma = 1.0)
35      double[,] kernel = {
36          { 1, 2, 1 },
37          { 2, 4, 2 },
38          { 1, 2, 1 }
39      };
40
41      int kernelSize = 3;
42      double kernelSum = 16.0; // Total jumlah dalam kernel
43      int kernelRadius = kernelSize / 2;
```

```
44      Bitmap blurredImage = new Bitmap(image.Width, image.Height);
45
46      for (int y = kernelRadius; y < image.Height - kernelRadius; y++)
47      {
48          for (int x = kernelRadius; x < image.Width - kernelRadius; x++)
49          {
50              double r = 0, g = 0, b = 0;
51
52              for (int ky = -kernelRadius; ky <= kernelRadius; ky++)
53              {
54                  for (int kx = -kernelRadius; kx <= kernelRadius; kx++)
55                  {
56                      int pixelX = x + kx;
57                      int pixelY = y + ky;
58
59                      Color pixelColor = image.GetPixel(pixelX, pixelY);
60
61                      r += pixelColor.R * kernel[ky + kernelRadius, kx + kernelRadius];
62                      g += pixelColor.G * kernel[ky + kernelRadius, kx + kernelRadius];
63                      b += pixelColor.B * kernel[ky + kernelRadius, kx + kernelRadius];
64                  }
65              }
66
67              // Menormalisasi hasilnya
68              int newR = Math.Min(255, Math.Max(0, (int)(r / kernelSum)));
69              int newG = Math.Min(255, Math.Max(0, (int)(g / kernelSum)));
70              int newB = Math.Min(255, Math.Max(0, (int)(b / kernelSum)));
71
```

```
72
73          blurredImage.SetPixel(x, y, Color.FromArgb(newR, newG, newB));
74      }
75  }
76
77  return blurredImage;
78 }
79
```

- Method untuk memperbaiki citra dengan mengurangi Noise

```

1 reference
85 private void button4_Click(object sender, EventArgs e)
86 {
87     //Mengurangi noise dengan gaussian blur
88     if (pictureBox1.Image != null)
89     {
90         // memuat gambar dari PictureBox
91         Bitmap originalImage = new Bitmap(pictureBox1.Image);
92
93         // memakai Gaussian Blur untuk mengurangi noise
94         Bitmap fixedImage = ApplyGaussianBlur(originalImage);
95
96         // Menampilkan hasil
97         pictureBox2.Image = fixedImage;
98         originalImage.Dispose();
99     }
100 }

```

- Method untuk melakukan Resize

```

1 reference
102 private void button2_Click(object sender, EventArgs e)
103 {
104     // Melakukan Resize
105     if (pictureBox1.Image != null)
106     {
107         // memuat gambar
108         Bitmap originalImage = new Bitmap(pictureBox1.Image);
109
110         // membagi 2 ukuran aslinya
111         int newWidth = originalImage.Width / 2;
112         int newHeight = originalImage.Height / 2;
113
114         // buat bitmap dengan ukuran/dimensi yang telah dibagi 2
115         Bitmap resizedImage = new Bitmap(newWidth, newHeight);

```

```

116 // Resize menggunakan nearest neighbor interpolation
117 for (int y = 0; y < newHeight; y++)
118 {
119     for (int x = 0; x < newWidth; x++)
120     {
121         int originalX = x * 2;
122         int originalY = y * 2;
123
124         // mengambil warna pixel dari gambar asli dan melakukan resize
125         Color pixelColor = originalImage.GetPixel(originalX, originalY);
126         resizedImage.SetPixel(x, y, pixelColor);
127     }
128 }
129
130 pictureBox2.Image = resizedImage;
131 originalImage.Dispose();
132 }
133 }
134 }

```

- Method untuk mengubah citra menjadi citra negative

```

136 1 reference
137 private void button3_Click(object sender, EventArgs e)
138 {
139     //Mengubah menjadi citra negatif
140     if (pictureBox1.Image != null)
141     {
142         Bitmap originalImage = new Bitmap(pictureBox1.Image);
143
144         // Buat Bitmap dengan dimensi yang sama
145         Bitmap negativeImage = new Bitmap(originalImage.Width, originalImage.Height);
146
147         // Perulangan untuk tiap pixel
148         for (int y = 0; y < originalImage.Height; y++)
149         {
150             for (int x = 0; x < originalImage.Width; x++)
151             {
152                 Color originalColor = originalImage.GetPixel(x, y);
153
154                 // rumus
155                 int newR = 255 - originalColor.R;
156                 int newG = 255 - originalColor.G;
157                 int newB = 255 - originalColor.B;
158
159                 // Mengubah pixel warna menjadi negatif
160                 Color negativeColor = Color.FromArgb(newR, newG, newB);
161                 negativeImage.SetPixel(x, y, negativeColor);
162             }
163         }
164
165         // Menampilkan hasil di pictureBox2
166         pictureBox2.Image = negativeImage;
167         originalImage.Dispose();
168     }
169 }

```

- Method untuk mengubah citra menjadi citra Biner

```

170 1 reference
171 private void button5_Click(object sender, EventArgs e)
172 //Memanggil fungsi convert ke citra biner
173 {
174     if (pictureBox1.Image != null)
175     {
176         // Memuat gambar asli dari pictureBox1
177         Bitmap originalImage = new Bitmap(pictureBox1.Image);
178         // ubah jadi biNary
179         Bitmap binaryImage = ConvertToBinaryImage(originalImage);
180         pictureBox2.Image = binaryImage;
181         // membersihkan
182         originalImage.Dispose();
183     }
184 }

```

```

185 1 reference
186 private Bitmap ConvertToBinaryImage(Bitmap image)
187 //Konversi ke Citra Biner
188 {
189     Bitmap binaryImage = new Bitmap(image.Width, image.Height);
190
191     for (int y = 0; y < image.Height; y++)
192     {
193         for (int x = 0; x < image.Width; x++)
194         {
195             Color originalColor = image.GetPixel(x, y);
196             // Menjadi grayscale pakai metode luminosity
197             int gray = (int)(0.3 * originalColor.R + 0.59 * originalColor.G + 0.11 * originalColor.B);
198             // Menentukan hitam dan putih
199             Color binaryColor = gray < 190 ? Color.Black : Color.White;
200             binaryImage.SetPixel(x, y, binaryColor);
201         }
202     }
203     return binaryImage;
204 }

```

- Method untuk mendeteksi tepi citra

```

206 private void button6_Click(object sender, EventArgs e)
207     //Deteksi Tepi
208 {
209     if (pictureBox1.Image != null)
210     {
211         // Membuat Bitmap baru dari PictureBox
212         Bitmap bmp = new Bitmap(pictureBox1.Image);
213
214         // Konversi Bitmap ke Image<Bgr, byte>
215         using (Image<Bgr, byte> emguImage = bmp.ToImage<Bgr, byte>())
216         {
217             // Mendapatkan Mat dari Image<Bgr, byte>
218             Mat pic = emguImage.Mat;
219             Mat gaussianBlur = new Mat();
220             Mat sobelX = new Mat();
221             Mat sobelY = new Mat();
222             Mat sobelXY = new Mat();
223
224             // Lakukan Gaussian Blur terlebih dahulu
225             CvInvoke.GaussianBlur(pic, gaussianBlur, new Size(3, 3), 5.0);
226
227             // Operasi Sobel untuk mendapatkan gradien dalam sumbu X, Y, dan kedua-duanya
228             CvInvoke.Sobel(gaussianBlur, sobelX, DepthType.Default, 1, 0, 3);
229             CvInvoke.Sobel(gaussianBlur, sobelY, DepthType.Default, 0, 1, 3);
230             CvInvoke.Sobel(gaussianBlur, sobelXY, DepthType.Default, 1, 1, 3);
231
232             // Buat variabel untuk menyimpan hasil Canny
233             Mat cannyPic = new Mat();
234
235             // Hitung nilai ambang batas bawah dan atas untuk Canny
236             var average = pic.ToImage<Gray, byte>().GetAverage();
237             var lowerThreshold = Math.Max(0, (1.0 - 0.33) * average.Intensity);
238             var upperThreshold = Math.Min(255, (1.0 + 0.33) * average.Intensity);
239
240             // Lakukan operasi Canny
241             CvInvoke.Canny(gaussianBlur, cannyPic, lowerThreshold, upperThreshold, 3);
242
243             // Konversi Mat hasil Canny ke Bitmap untuk ditampilkan di PictureBox2
244             Bitmap cannyBitmap = cannyPic.ToBitmap();
245
246             // Tampilkan hasil Canny di PictureBox2
247             pictureBox2.Image = cannyBitmap;
248
249             // Bersihkan sumber daya
250             pic.Dispose();
251             gaussianBlur.Dispose();
252             sobelX.Dispose();
253             sobelY.Dispose();
254             sobelXY.Dispose();
255             cannyPic.Dispose(); // Pastikan untuk membebaskan sumber daya Mat cannyPic
256         }
257     }
258 }

```

- Method untuk mendeteksi Objek

```

259 private void button7_Click(object sender, EventArgs e)
260     //Deteksi Objek
261 {
262     if (pictureBox1.Image != null)
263     {
264         Bitmap bmp = new Bitmap(pictureBox1.Image);
265
266         using (Image<Bgr, byte> emguImage = bmp.ToImage<Bgr, byte>())
267         {
268             Mat objek = emguImage.Mat;
269             VectorOfVectorOfPoint contours = new VectorOfVectorOfPoint();
270
271             Mat thresholdPic = new Mat();
272             Mat hierarchy = new Mat();
273
274             Image<Gray, byte> grayPhone = objek.ToImage<Gray, byte>();
275

```

```

276 //Untuk citra yang berwarna lebih baik pake trshold tinggi begitu pun sebaliknya
277 CvInvoke.Threshold(grayPhone, thresholdPic, 175, 255, Emgu.CV.CvEnum.ThresholdType.Binary);
278
279 CvInvoke.FindContours(thresholdPic, contours, hierarchy, Emgu.CV.CvEnum.RetrType.Tree, Emgu.CV.CvEnum.ChainApproxMethod.ChainApproxNone);
280
281 Mat contourImage = new Mat(objek.Size, DepthType.Cv8U, 3);
282 CvInvoke.DrawContours(contourImage, contours, -1, new MCvScalar(0, 255, 0), -1);
283
284 Bitmap contourBitmap = contourImage.ToBitmap();
285
286 pictureBox2.Image = contourBitmap;
287
288
289

```

- Method untuk me-reset

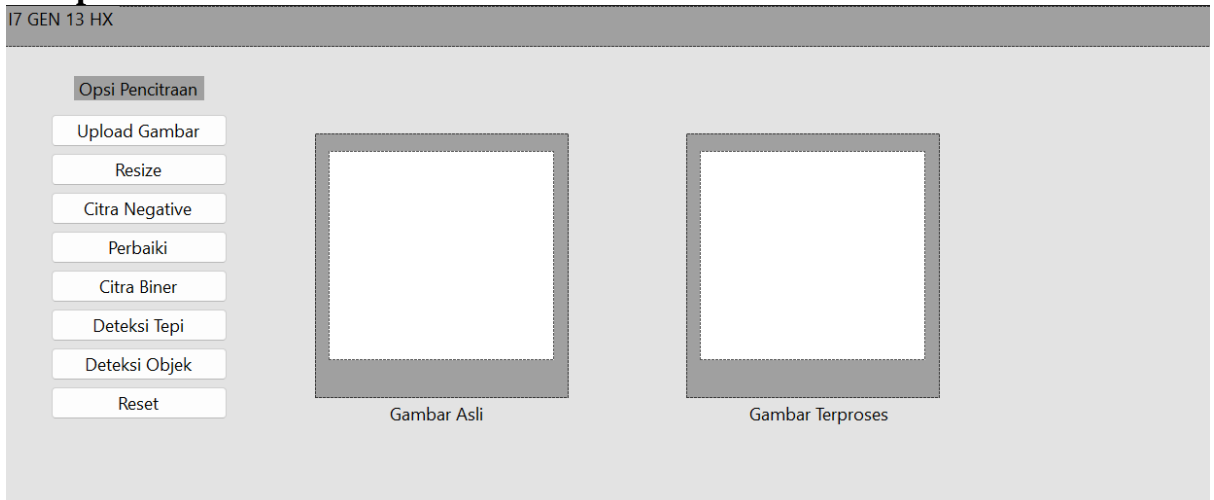
```

291 1 reference
292 private void button8_Click(object sender, EventArgs e)
293 {
294     // Bersihkan gambar di pictureBox2 dengan menghapus referensi Image-nya
295     pictureBox2.Image = null;
296 }
297
298

```

Demo Aplikasi:

- Tampilan Awal



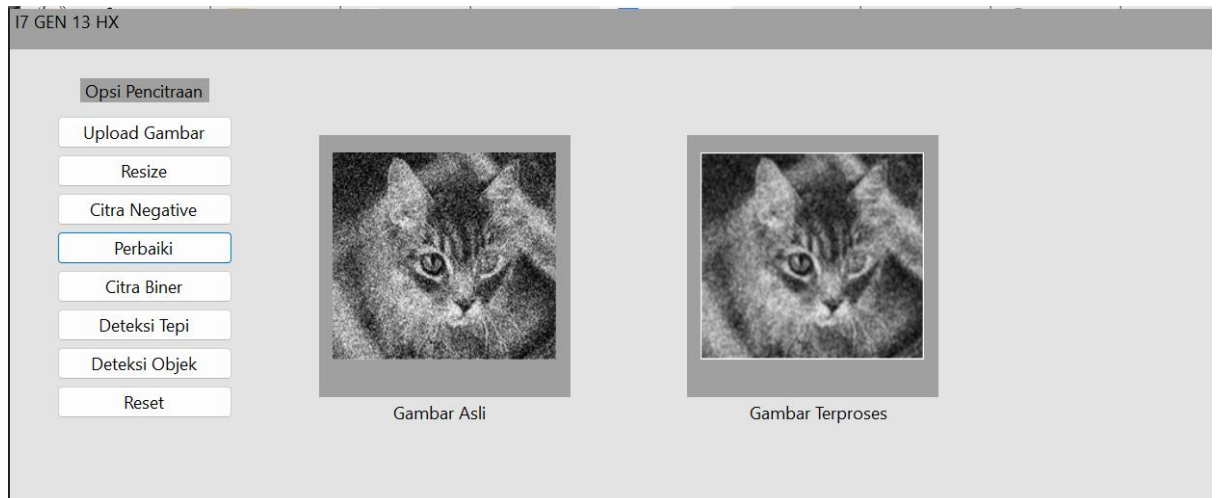
- Fitur Resize



- **Fitur Citra Negatif**

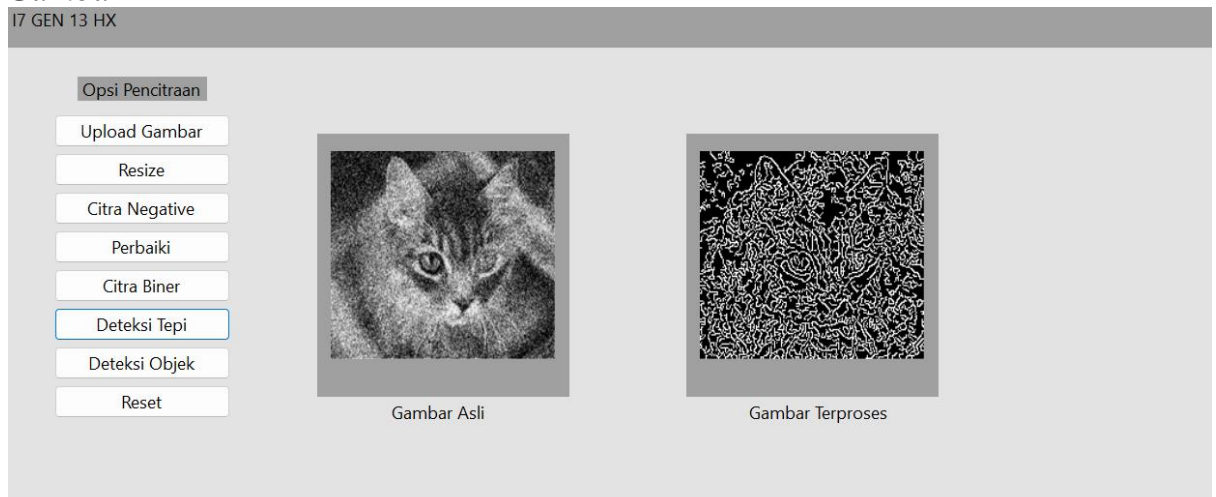


- **Fitur Perbaiki Citra**



- **Fitur Pendeteksian Tepi**

Gambar 1



Gambar 2

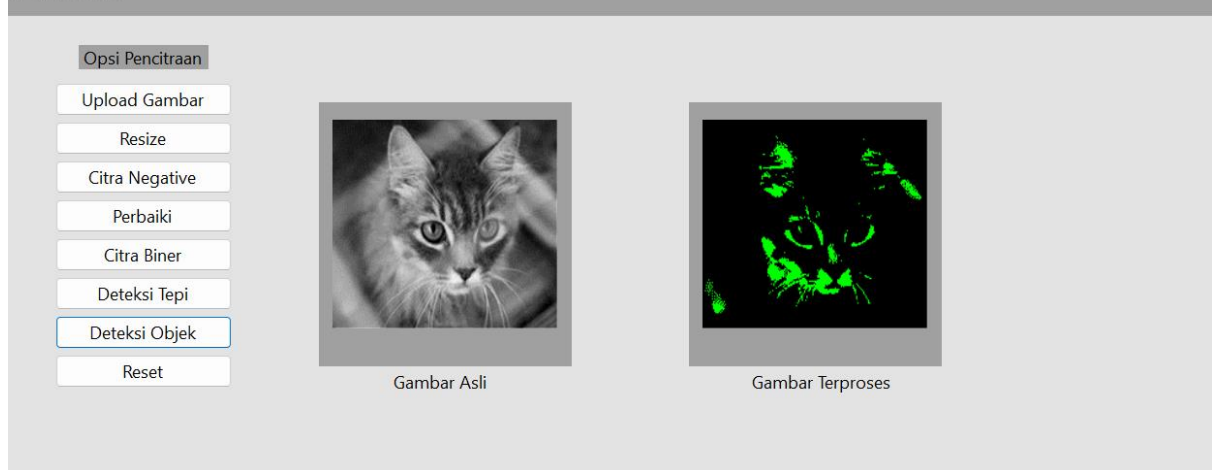
17 GEN 13 HX



- **Fitur Pendeteksian Objek**

Gambar 1

17 GEN 13 HX



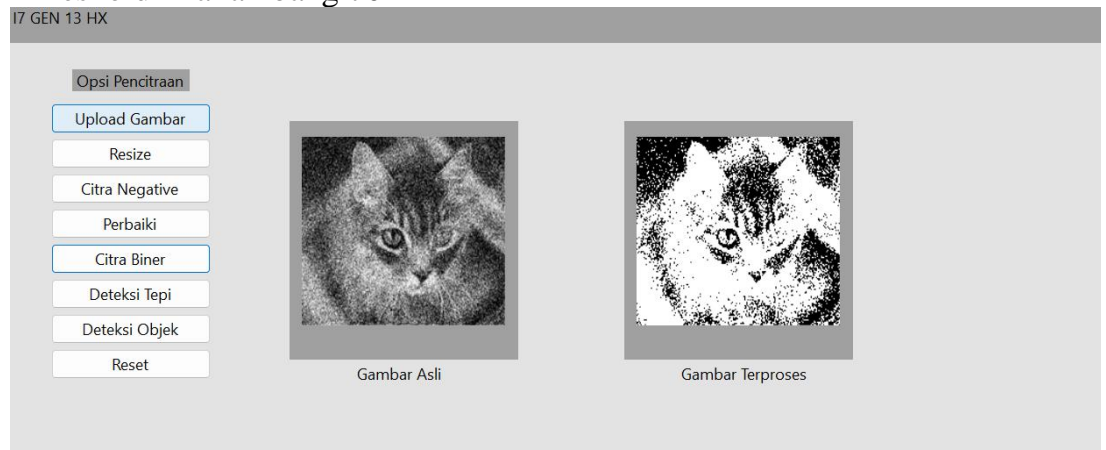
Gambar 2

17 GEN 13 HX

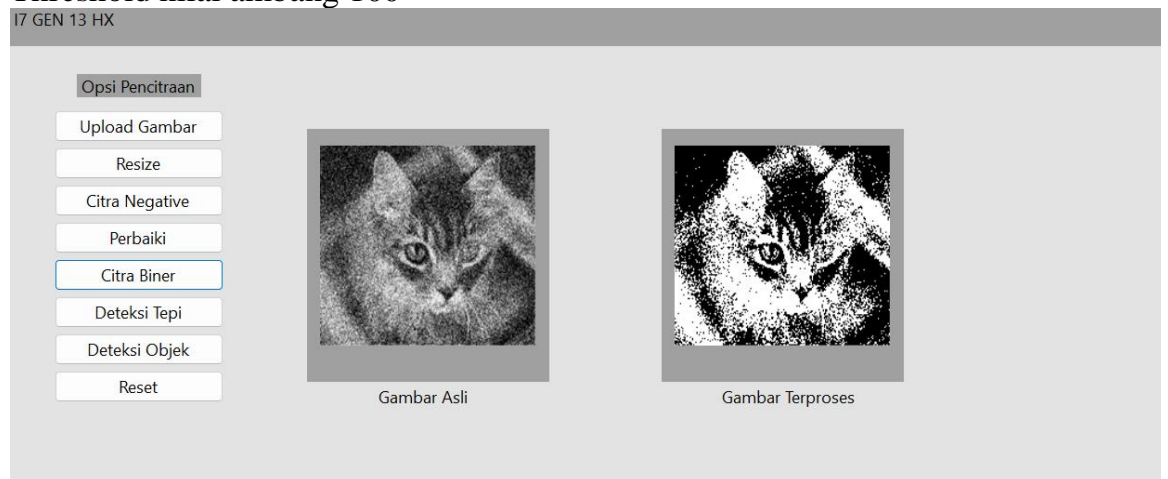


- **Fitur Citra ke Biner**

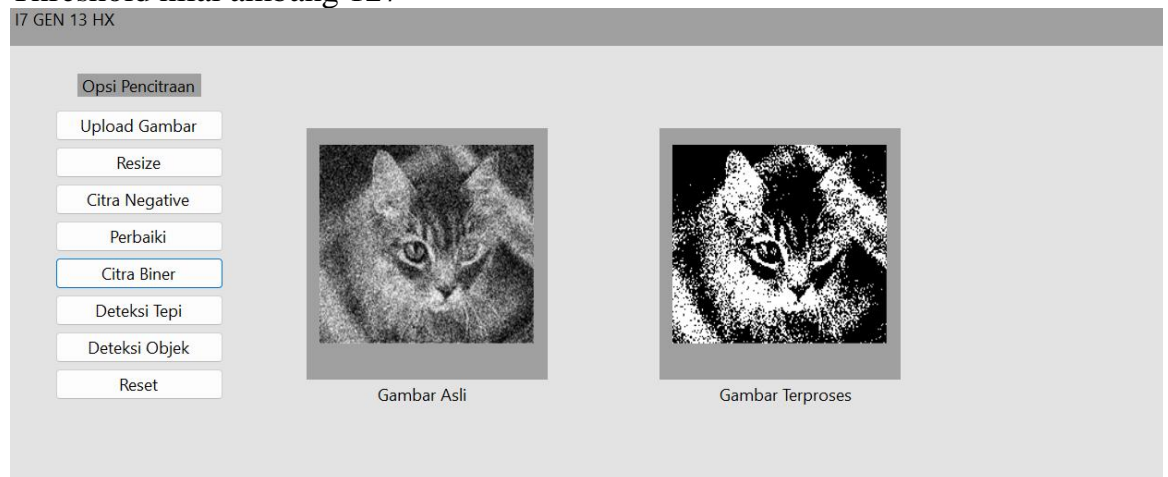
1. **Threshold nilai ambang 70**



2. **Threshold nilai ambang 100**



3. **Threshold nilai ambang 127**



4. Threshold nilai ambang 190

17 GEN 13 HX

Opsi Pencitraan

Upload Gambar

Resize

Citra Negative

Perbaiki

Citra Biner

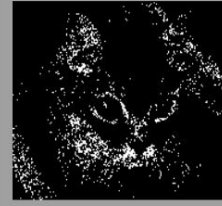
Deteksi Tepi

Deteksi Objek

Reset



Gambar Asli



Gambar Terproses