Image Edge Detection for Batak Ulos Motif Recognition using Canny Operators

Sarinah Sitanggang[1], Paska Marto Hasugian[2]
Informatics Engineering Study Program, STMIK Pelita Nusantara, Medan, Indonesia.

Article Info

Keywords:

Edge detection,
batak ulos,
canny operator

ABSTRACT

Batak ulos is one of the handicrafts originating from North Sumatra. In Batak Ulos, there are various kinds of motifs that are characteristic of these ulos. One way to find out the type of ulos is by knowing the motives found on the ulos. For that we need a system that can detect ulos and then can recognize these ulos. The system was built using the canny edge detection method proposed by Jhon Canny in 1986, and is known as the optimal edge detection operator. Canny operator is one of edge detection which is very good in detecting image edges. Canny operators have met the criteria in detecting, namely detecting very well, responding well and localizing well and clearly. The system built also uses the C # programming language in Microsoft Visual Studio 2010.

Corresponding Author:

Sarinah Sitanggang,
Informatics Engineering Study Program,
STMIK Pelita Nusantara, Medan, Indonesia,
Jl. Iskandar Muda No.1, Merdeka, Kec. Medan Baru, Kota Medan, Sumatera Utara 20154.
Email: sitanggang.sarinah@gmail.com

1. INTRODUCTION

Image or image is a visual form of information from an object. Image is a combination of points, lines, fields, and colors to create an object[1]–[3]. From an image, very important information can be obtained, be it information about the area, tourist objects, history and others[4], [5]. By looking at an image, someone can give an impression and message to an object of the image being observed. For this reason, many image objects are used by the community as information media because it is believed that people are more interested in seeing an image than reading[6], [7].

Batak ulos is one of the handicrafts developed from generation to generation from North Sumatra. In Batak Ulos, there are various forms of motifs and colors. Batak ulos are always included in every activity such as weddings, births, and sorrows. Each ulos has different meanings and meanings, for that at certain events or activities not all types of ulos can be used. Currently, many Batak people cannot distinguish the meaning and meaning as well as the types of ulos, this can be influenced by the development of an increasingly modern era as a result many people have forgotten about culture and traditions. In addition, many people are not interested in seeking information and learning about Batak culture. This can be caused by the Batak people who were born outside the land of Batak and also young people.

Currently there is no application that can be used in the introduction of ulos motifs, for that it is necessary to build a system that can be used to detect ulos motifs, so that the public will easily recognize and distinguish these types of ulos.

Image edge detection can be used to detect all edges or lines that make up an image object. One of the operators of edge detection is canny edge detection[8][9]. Canny operators have met three criteria as proposed by Jhon Canny in 1986[10]. These criteria are good detection, good location, one response to single edge[11], [12]. One of the studies using the canny operator is "Introduction to Batik Motif Using Canny Edge Detection and K-Nearest Neighbor" by Johannes Widagdho Yodha, Achmad Wahid Kurniawan, November 2014. According to this study, it was written that the canny operator was able to detect edges with a minimum error rate. The canny operator will detect the ulos, but before going to the image detection stage, the image is first converted into a grayscale image, then the results of the grayscale image are segmented, then the canny operator is applied to detect the ulos image and the system to be built uses a programming language. C # in Microsoft visual studio 2010.

2. RESEARCH METHOD

The purpose of the framework is to assist in compiling the form of design that will be carried out in completion of the research. The framework is the steps that will be carried out in solving the problems found. From this framework, the research implementation process can be seen. The following are the stages of the framework in this study:

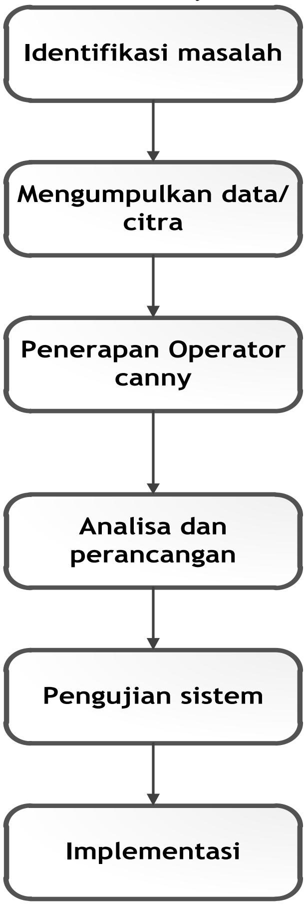


Figure 1. Research framework

Based on the framework described in Figure 1 The following is an explanation of each of these steps:

1. Identify the problem
   Problem identification is one of the most important processes in completing a study. The problem in this research is, how to recognize ulos based on the motives that exist in these ulos.
2. Collecting data / images
   At this stage, the process involves data collection, including the following:
   a. Observation

Observation is one of the activities carried out to observe an object to be studied, then from that observation, the information needed to compile the research to be made will be obtained. In this study, the authors made observations in Pangururan Samosir, the authors observed several types of ulos which have almost the same motive but have different meanings and meanings.

b. Interview
Interviews are the process of collecting data to obtain information by asking respondents directly. The interview was carried out directly to one of the ulos weaving craftsmen who had participated in many ulos exhibitions, by asking the question what is the characteristic feature of the ulos batak, what is the meaning and meaning of the ulos batak motif and what makes up to now there are still many that have not know the difference of each existing motive on these ulos.

c. Documentation
The documentation method is used to obtain the data needed to meet the needs in compiling research and has not been obtained from the observation and interview process. In this study, the information about batak ulos must be accompanied by a picture of the ulos motif. To take a picture of the ulos motif, the writer uses a camera, and from the picture the name and shape of the ulos motif needed for research needs will be explained.

3. Analysis
Analysis is a stage of the research process where the data that has been collected will be managed to be processed. The stages in the analysis process in this study include:

a. Performs review purposes
The review process is carried out directly to find out the types of ulos batak motifs that exist.

b. Take into account the results of the review
After finding the data from the results of the review, a comparison was made of each ulos motif to consider the research.

c. Look for sources for facts
Data search can also be done by searching various sources, such as collecting books that discuss ulos batak to obtain more accurate information.

d. Analyze the findings from the review
After all the results and sources are found, more accurate information will be analyzed regarding the Batak ulos, so that later it can be accounted for.

4. Design
The system design that will be made includes the C # programming language on Micrososft Visual Studio 2010, and UML. In this system, a system design will be designed to detect the ulos motif image using the canny operator.

5. System testing
System testing is carried out to ensure that the resulting system is in accordance with the needs for completing the research. In addition, system testing is also carried out to provide an assessment of the system being created and see the ability to detect in accordance with the objectives of the research being carried out.

6. Implementation
The implementation stage is the final stage in this research. Implementation is the stage for implementing and maintaining the system.

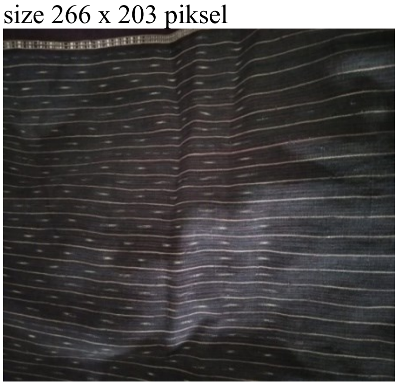
3. RESULTS AND DISCUSSION

3.1. Data analysis

The data used in this research is the Ulos Batak image in JPG format and the output image is the BMP format. The first step taken for image processing is to enter an RGB image then the image is converted to a grayscale image. Then proceed with the segmentation process and then proceed with the image edge detection process to identify ulos motifs using the C # language in Microsoft Visual Studio 2010. The following is the image data that will be processed:

Table 1. Ulos Batak Image Data Sample To Be Detected

Image of Ulos Batak	Name Motive	Information

size 266 x 203 piksel 	Gosta	The basic color of the ulos is red and the color of the motif is white. Usually used as a headband and sling cloth.
size 375 x 285 piksel 	Sibolang, runcing	The basic colors are black and white and blue to form the ulos motif. Usually given to the bride and groom.
size 266 x 203 piksel 	Rotan(hotang)	The basic color is black, and the function of the rattan on this ulos is as a binder.

size 375 x 285 	Garis-garis putus	The base color is red. Usually used for mourning.
size 375 x 285	Gugusan bintang	This star cluster describes the harmony in the family that is expected to remain compatible and obedient.

		
size 375 x 285 	Marsunsang(berlawanan arah)	Its base color is red, and is usually used as a sling and headband.

size 375 x 285 	Bentuk motif seperti burung tekukur	This ulos is a special ulos for the clan descendants of Silalahi fighting.
size 375 x 285 	Bentuk motif seperti garis-garis	This ulos is the longest ulos, and is usually used when dancing.
Size 375 x 285	Bentuk motif seperti panah	The basic color is red, but is given a little black as lyrical. Usually used as a

		head band for music players.
size 375 x 285 	Motif pada ulos ini rambu(rambut) ulos	Ulos is usually used for women who have miscarried.
size 375 x 285 	Bentuk motif seperti sisir	This Ulos is a symbol of neatness. The base color is black.
size 375 x 285 	Bentuk motif adalah panah	The basic color is red, and this ulos is believed to have a very strong mystique.

3.2. Analysis Method For Edge Detection

The process carried out is by entering the original image containing RGB values, then the image is converted into a grayscale image. The goal is to simplify the detection process. After the image is converted to a grayscale image, then the grayscale image is processed with the canny operator for edge detection. The following is the image edge detection process that will be detected using Visual Studio 2010.

a. RGB Image Input

The first step in image processing is inputting an RGB image. RGB image is an image that presents colors in the form of components R (red), G (green), B (blue). To get the RGB value in the image, you can use Matlab to make manual calculations easier by inputting a program like the following:

```
>> Img = imread ('filename.jpg');
>> figure, imtool (Img);
```

Below is a display of the image and the representation of the pixel value of the image with a 5 x 5 kernel.

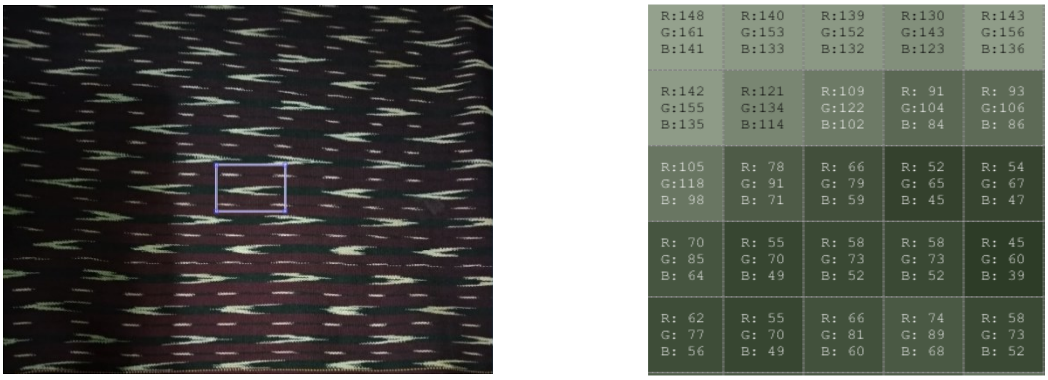


Figure 2. RGB image and image color representation

b. Grayscale image

The second step is the process of converting an RGB image to a grayscale image or grayscale image. A grayscale image is an image that only has black and white colors. The pixel value in the grayscale image is obtained from the pixel value in the RGB image. Each pixel value in the RGB image is added and divided by 3. The following is a manual calculation for a grayscale image:

Table 2. Pixel values for R (red)

R	0	1	2	3	4
0	148	140	139	130	143
1	142	121	109	91	93
2	102	78	66	52	54
3	70	55	58	58	45
4	62	55	66	74	58

Table 3. Pixel values for G (green)

G	0	1	2	3	4
0	161	153	152	143	156
1	155	134	122	104	106
2	118	91	79	65	67

3	85	70	73	73	60
4	77	70	81	89	73

Table 4. Pixel values for B (blue)

B	0	1	2	3	4
0	141	133	132	123	136
1	135	114	102	84	86
2	98	71	59	45	47
3	64	49	52	52	39
4	56	49	60	68	52

Each pixel value contained in the table is obtained from the RGB pixel value found in Figure 2. After the value of each pixel in the image is found, manual calculations are carried out to convert the RGB image to a grayscale image. The following is a manual calculation of grayscale images with image samples at coordinate points [0,0], [1,1], [2,2]

S = (148 + 161 + 141) / 3 = 150

S = (121 + 134 + 114) / 3 = 123

S = (66 + 79 + 59) / 3 = 68

This calculation is done for each pixel in the image, after the calculation is done it will produce an image with a grayscale format as shown in Table 5 below:

Table 5. Grayscale image pixel values

S	0	1	2	3	4
0	150	142	141	132	145
1	144	123	111	93	95
2	106	80	68	54	56
3	73	58	61	61	48
4	65	58	69	77	61

The following is an example of a grayscale image:

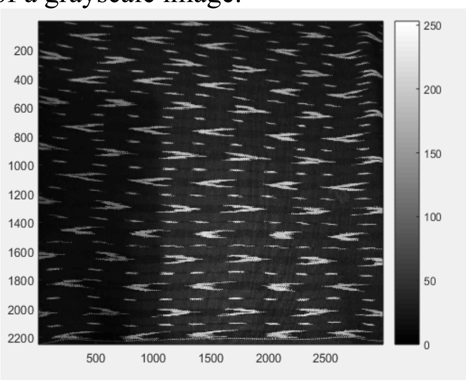


Figure 3. Grayscale image

c. Canny edge detection

The canny operator is one of the optimal edge detectors. The canny operator is able to detect the edge very clearly. The way the canny edge detection works is to first do a smooting image using the Gaussian filter then find the edge by looking for the local maxima of the f (x.y) gradient. Then the gradient is calculated using Gaussian. The canny operator uses two thresholds to detect a strong edge (high) and a weak edge (low), and inserts a weak edge in the output only when they connect to a strong edge. Hence, this method is more likely to detect weak edges correctly. Following are the steps in the edge detection process using the canny operator:

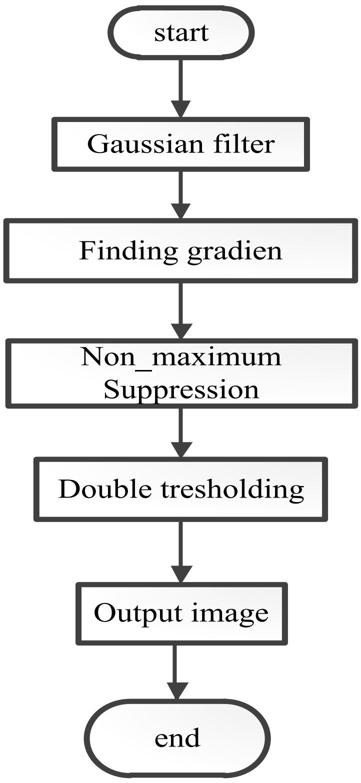


Figure 4. Flowcart Operator Canny

a. *Gaussian* filter

The first step is to define a Gaussian kernel, for example a kernel of 5.

-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2

-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2
-2	-1	0	1	2

Figure 5. Example of horizontal and vertical kernels

For example, the weights can be used by making σ^2 equal 1.Thus, the following results are obtained using the formula:

$$G(y.x) = e^{\frac{x^2+y^2}{2\sigma^2}}$$

$$G[0,0] = \frac{1}{2 * \pi * 1} * \exp\left(-\frac{(-2)^2 + (-2)^2}{2 * 1^2}\right)$$

$$G[0,0] = \frac{1}{2 * 3{,}14 * 1} * \exp(-4)$$

$$G[0,0] = 0{,}1591549 * 0{,}0183$$

$G[0,0] = 0{,}002915$

The above calculation can be done for each pixel value in the horizontal and vertical kernels, resulting in the following kernel values:

Table 6. Intermediate kernel values

0,002914013	0,013073248	0,021544586	0,013073248	0,002914013
0,013073248	0,058582803	0,096576433	0,058582803	0,013073248
0,021544586	0,096576433	0,159235669	0,096576433	0,021544586
0,013073248	0,058582803	0,096576433	0,058582803	0,013073248
0,002914013	0,013073248	0,002914013	0,013073248	0,002914013

From the results of table 6, the smallest value of the kernel formed is then taken to calculate the multiplier value.

$$Multiplier = \frac{1}{minvalue}$$

Minvalue = 0,002914013

$$Multiplier = \frac{1}{0{,}002914014}$$

$$Multiplier = 343{,}16$$

After the multiplier value is found, the next step is to multiply the intermediate value by the multiplier value to produce a Gaussian kernel. So that it produces a Gaussian kernel like table 7:

Table 7. Gaussian kernels

1	4	7	4	1
4	20	33	20	4
7	33	55	33	7
4	20	33	20	4
1	4	7	4	1

The second step is to determine the weight of the Gaussian kernel

To calculate the weight of the Gaussian kernel, it can be calculated by adding up each value listed in table 7, the results are as follows:

$$W = 1 + 4 + 7 + 4 + 1 + 4 + 20 + 33 + 20 + 4 + 7 + 33 + 55 + 33 + 7 + 4 + 20 + 33 + 20 + 4 + 1 + 4 + 7 + 4 + 1$$

W = 331

The third step is to determine the pixel value of the image after convoluting it with the Gaussian kernel. This process involves convolution of the Gaussian kernel in table 7 with the pixels to be processed. The following is an example of a 5 x 5 pixel grayscale image.

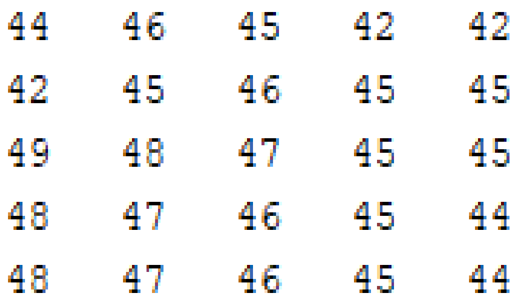


Figure 6. Image sample 10x10

From this image, the pixel value with a size of 5x5 is taken to be combined with the pixel value contained in the Gaussian kernel. Convolution is done by multiplying the values in each appropriate matrix, and then adding up all the resulting values, then dividing by the weight of the previous step.

Table 8. Pixel convolution with Gaussian kernel

	44	46	45	42	42

	42	45	46	45	45		1	4	7	4	1
fg :	49	48	47	45	45		4	20	33	20	4
	48	47	46	44	45	*	7	33	55	33	7
	48	47	46	44	45		4	20	33	20	4
							1	4	7	4	1

Table 9. The result of pixel convolution with the Gaussian kernel

	44	184	315	168	42
	168	900	1518	900	180
fg:	343	1584	2585	1485	315
	192	940	1518	880	180
	48	188	322	176	45

Furthermore, all the values resulting from the convolution are then added and divided by the total weight of the Gaussian kernel.

$$f[2,2] = \frac{sum(fG)}{331}$$
$$f[2,2] = \frac{15220}{331}$$
$$f[2,2] = 46$$

This process is continued for each related pixel, so that an image sample with a size of 10 x 10 will produce an image that has gone through the Gaussian filter process:

Table 10. Gaussian filter results

44	46	45	42	42
42	45	46	45	45
49	48	46	45	45
48	47	46	45	44
48	47	46	45	44

b. Gradient piksel

The process for calculating the pixel gradient intensity value is carried out like the Gaussian filter process. To find a gradient done using the sobel operator. The calculation is done by taking the image value in table 10 with a 3 x 3 kernel.

The first step is to determine the Y derivative on the vertical axis

	1	0	-1		44	46	45
Gx:	1	0	-1	*	42	45	46
	1	0	-1		49	48	46

	44	0	-45
Gx:	42	0	-46
	49	0	-46

Gx[1,1] = 44 + 0 +(-45) + 42 + 0 + (-46) + 49 + 0 + (-46)
= - 2

This process is carried out for each pixel value in table 4.11, the following results will be obtained:

Table 11. Samples of the X derivative

0	0	0	0	0

0	-2	7	-6	-7
0	1	5	4	-2
0	7	7	6	0
0	6	6	5	0

Determines the Y derivative on the horizontal axis

	1	1	1		44	46	45
Gy:	0	0	0	*	42	45	46
	-1	-1	-1		49	48	46

	44	46	45
Gy:	0	0	0
	-49	-48	-46

Gx [1,1] = 44 + 46 + 45 +0 + 0 + 0 +(-49) + (-48) + (-46)
= - 8

This process is also performed for each pixel. So as to produce a pixel value like the following

Table 12. Samples of Y derivatives

0	0	0	0	0
0	-8	-6	-5	-2
0	-8	-2	1	3
0	2	1	2	1
0	0	0	-1	-1

Specifies the gradient value

To determine the gradient value can be calculated using a formula : $G = \sqrt{Gx^2 + Gy^2}$

For example, by taking a pixel gradient intensity value [2,2]:

$$G = \sqrt{5^2 + -2^2}$$

$$G = \sqrt{29} \quad G = 5$$

So that the resulting gradient intensity values are as follows:

Table 13. Samples of gradient intensity values (G)

0	0	0	0	0
0	8	9	8	7
0	8	5	4	3
0	7	7	6	1
0	6	6	5	1

c. Non-maximum emphasis

Non-maximum pressing can be performed by knowing in advance the pixel gradient value, comparing the intensity of the pixel value with the corresponding neighboring pixel. For example, by performing calculations using pixels [2,2].

$$\theta = atan(\frac{Gy}{Gx}) * \frac{180}{\pi}$$

$$\theta = atan(\frac{-2}{5}) * \frac{180}{\pi}$$

$$\theta = 1{,}19 * \frac{180}{\pi}$$

$$\theta = 68$$

After the gradient value is obtained, then perform a comparison process with the intensity value. Since the value of 68 lies between 67.5 and 157.5, pixel [2,2] is a pixel with a vertical gradient and the intensity value will be compared with the pixel value directly above or below it.

Table 14. Sample to determine the pixel gradient direction [2,2]

0	0	0	0	0
0	8	9	8	7

0	8	**5**	4	3
0	7	7	6	1
0	6	6	5	1

The pixel value [2,2] = 5 is smaller with the pixel value above and below it, so the pixel value [2,2] is changed to 0.

Table 15. Results of non-maximum emphasis on pixels [2,2]

0	0	0	0	0
0	8	9	8	7
0	8	**0**	0	0
0	7	7	6	0
0	6	6	0	0

d. Dual tresholding

After the non-maximum pressing process, then the tresholding process is carried out. Before the edge detection process, the segmentation process is first carried out. In this research, it is used is tresholding. The output of the tresholding process is a binary image or an image that has intensity values of 0 and 1.

The equation used to convert the pixel value in a grayscale image to binary in the tresholding method is:

$$g(x,y) = \begin{Bmatrix} 1, jika\ f\ (x,y) \geq T \\ 0, jika\ f\ (x,y) < T \end{Bmatrix}$$

Where :

F(x,y) = citra *grayscale*

G(x,y) = citra biner

T = nilai *Tresholding*

After all steps are applied the output of the image is the image that has been detected using the CANNY operator. From the detection results using the Canny operator, the motive of the ulcer will be known.

3.3. Implementation

The following is the result of the implementation of the image edge detection application for the introduction of the batak ulos motif using the canny operator:

When the application is run, the page that will be displayed in the application is as shown below:

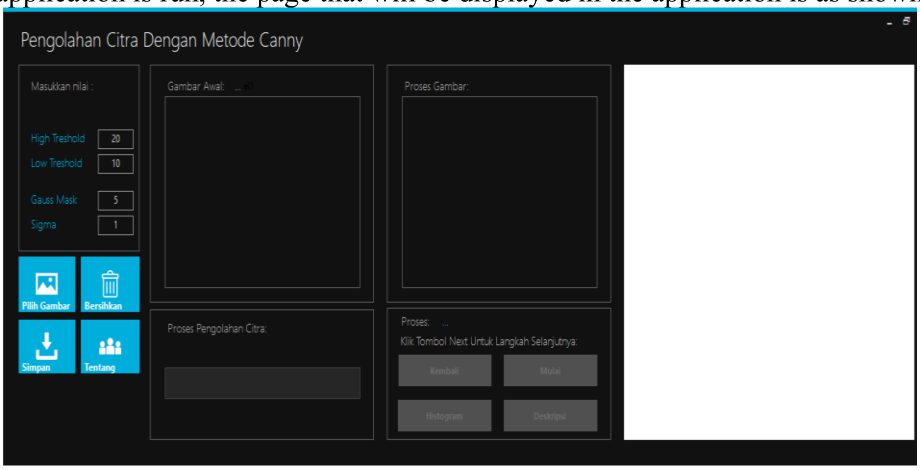


Figure 7. Main page

1. Click the "select image" button to enter the image to be detected, and an open file will appear as shown below:

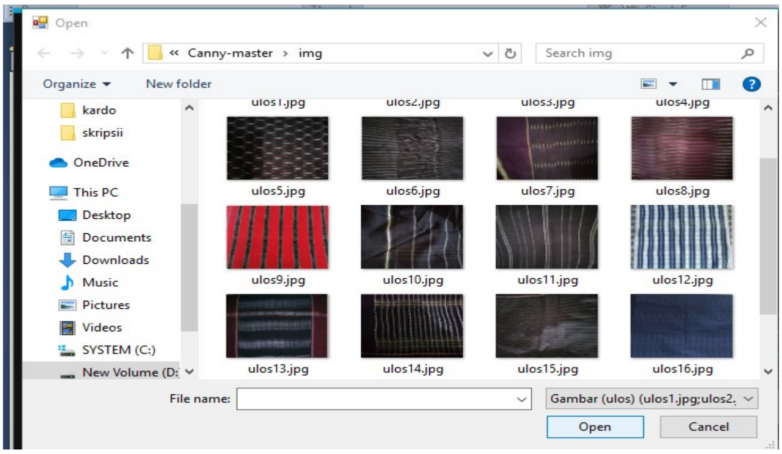


Figure 8. Display open file

2. Then select one of the images to be detected. For example, select the ulos image "ulos5.jpg". Then the appearance is as follows:

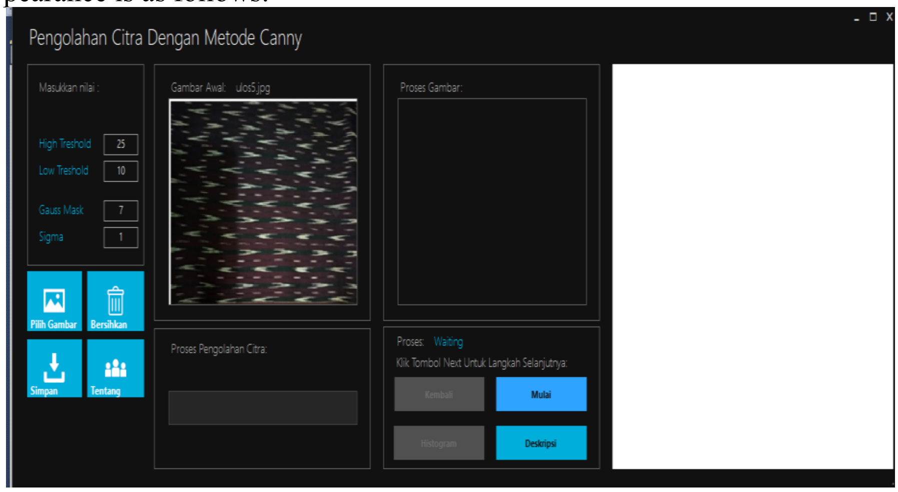


Figure 9. Current display of input image

3. The next step is, by pressing the "start" button for the image processing process starting from the Gaussian filter. But before pressing the "start" process button, you can first set the value to be input in the High and low threshold columns, then the Gaussian and sigma columns:

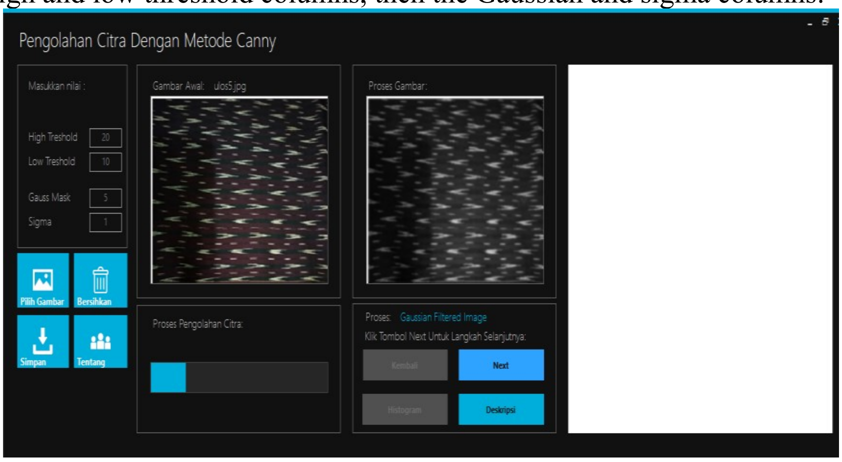


Figure 10. Gaussian filter process

4. After the Gaussian filter process is complete, then press the "next" button again for the found image gradient process, and the results are as shown below:

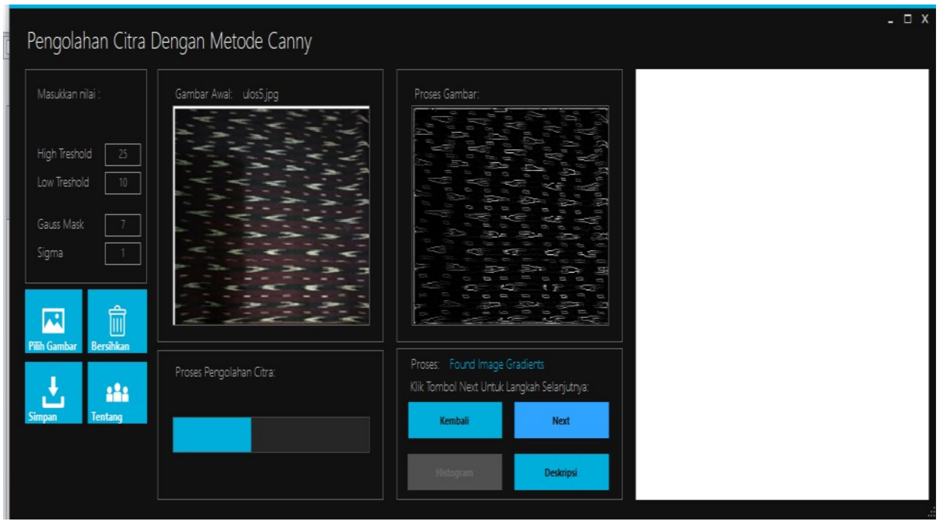


Figure 11. The process of found image gradients

5. The next step is to press the next button to do the found local maxima process, and it looks like this:

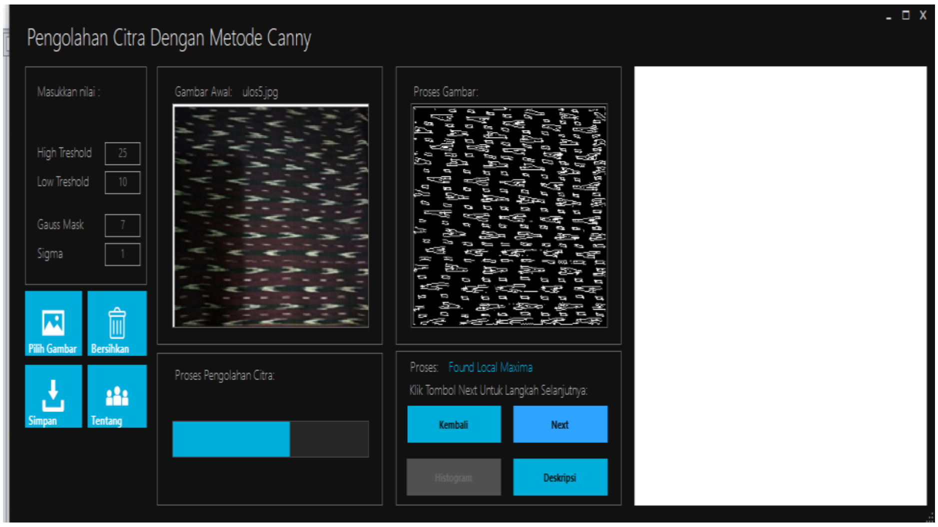


Figure 12. Local maxima foud process

6. The next step is to press the next button again, for the thresholding process, and the results are as follows:

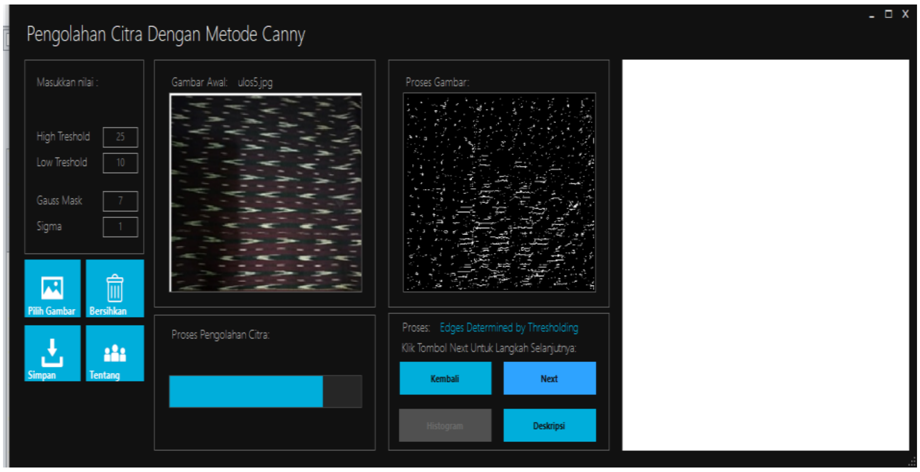


Figure 13. Thresholding process

7. The final step is to press the "next" button for the last process, namely the Canny process, and the results are as shown below:

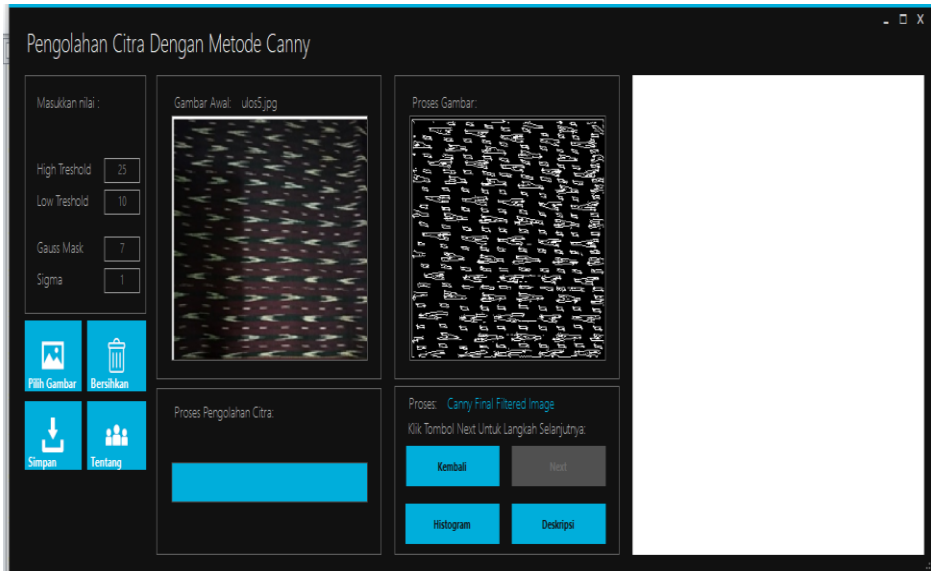


Figure 14. Canny Process

4. CONCLUSION

After completing the process of making a system to detect edges using the CANNY operator, it can be concluded that several things are The application can detect an image just by following the steps in the main display. The canny operator can clearly detect the input image. The application for the introduction of Batak ulos motifs can be built using Microsoft Visual Studio 2010.

REFERENCES

[1] M. Al Amin and D. Juniati, "Klasifikasi Kelompok Umur Manusia Berdasarkan Analisis Dimensi Fraktal Box Counting Dari Citra Wajah Dengan Deteksi Tepi Canny," *J. Ilm. Mat.*, 2017.

[2] E. Budianita, J. Jasril, and L. Handayani, "Implementasi Pengolahan Citra dan Klasifikasi K-Nearest Neighbour Untuk Membangun Aplikasi Pembeda Daging Sapi dan Babi Berbasis Web," *J. Sains dan Teknol. Ind.*, 2015.

[3] C. N. Santi, "Mengubah Citra Berwarna Menjadi Gray-Scale dan Citra biner," *Teknol. Inf. Din.*, 2011.

[4] R. A. Surya, A. Fadlil, and A. Yudhana, "Ekstraksi Ciri Metode Gray Level Co-Occurrence Matrix (GLCM) dan Filter Gabor untuk Klasifikasi Citra Batik Pekalongan," *J. Inform. Pengemb. IT (JPIT , Vol. 02, No. 02, Juli 2017*, 2017.

[5] "ANALISIS KOMBINASI CITRA SENTINEL-1A DAN CITRA SENTINEL-2A UNTUK KLASIFIKASI TUTUPAN LAHAN (STUDI KASUS: KABUPATEN DEMAK, JAWA TENGAH)," *J. Geod. Undip*, 2018.

[6] R. Sampurno and A. Thoriq, "KLASIFIKASI TUTUPAN LAHAN MENGGUNAKAN CITRA LANDSAT 8 OPERATIONAL LAND IMAGER (OLI) DI KABUPATEN SUMEDANG," *J. Teknotan*, 2016, doi: 10.24198/jt.vol10n2.9.

[7] K. Firdausy, I. Hawariyanta, and M. Murinto, "IMPLEMENTASI WATERMARKING UNTUK PENYEMBUNYIAN DATA PADA CITRA DALAM DOMAIN FREKUENSI MENGGUNAKAN DISCRETE COSINE TRANSFORM," *TELKOMNIKA (Telecommunication Comput. Electron. Control.*, 2006, doi: 10.12928/telkomnika.v4i1.1240.

[8] V. Lusiana, "Deteksi Tepi pada Citra Digital menggunakan Metode Kirsch dan Robinson," *J. Teknol. Inf. Din.*, 2019.

[9] L. Sitinjak, "DESIGN OF DIGITAL IMAGE EDGE DETECTION APPLICATIONS USING FREI-CHEN ALGORITHM | INFOKUM," *INFOKUM*, 2020. http://infor.seaninstitute.org/index.php/infokum/article/view/13 (accessed Jan. 07, 2021).

[10] S. Sukatmi, "Perbandingan Deteksi Tepi Citra Digital dengan Menggunakan Metode Prewitt, Sobel dan Canny," *KOPERTIP J. Ilm. Manaj. Inform. dan Komput.*, 2017, doi:

10.32485/kopertip.v1i1.3.
[11] I. Faturrahman, "PENGENALAN POLA HURUF HIJAIYAH KHAT KUFI DENGAN METODE DETEKSI TEPI SOBEL BERBASIS JARINGAN SYARAF TIRUAN BACKPROPAGATION," *J. Tek. Inform.*, 2018, doi: 10.15408/jti.v11i1.6262.
[12] A. Haryoko and S. H. Pramono, "Pengenalan Karakter Plat Kendaraan Bermotor berbasis Citra dengan menggunakan Metode Canny dan Algoritma Backpropagation," *J. Mech. Eng. Mechatronics Submitt.*, 2016.