

IMAGE DETECTION EDGE IMAGE USING CANNY EDGE ALGORITHM

¹Zekson Arizona Matondang, ²Romanus Damanik, ³Parasian D.P Silitonga,

¹Sistem Informasi, Universitas Katolik Santo Thomas Medan, Jl. Setia Budi No. 479-F Medan

²Sistem Informasi, Universitas Katolik Santo Thomas Medan, Jl. Setia Budi No. 479-F Medan

³Teknik Informatika, Universitas Katolik Santo Thomas Medan, Jl. Setia Budi No. 479-F Medan
rdfikom@gmail.com

Abstract

In general, edge detection is widely used in image processing to find the boundaries of objects in an image. In this study, a technique for detecting edges that is commonly used will be studied, namely the Canny method under different conditions using Visual Studio software. In the testing process that the author carried out, it was concluded that sooner or later a process of converting an object would be, that the following factors affect the conversion process, including; type of image, image resolution, image format, specifications of the camera used to take pictures as well as the specifications of a computer that has a processing speed that is expected to be able to process images (images). The use of the Canny method to detect the edges of an image is considered more accurate and effective because this method is able to provide detailed output by including a calculated execution time.

Keywords: Image Edge Detection, Canny, Canny Edge Algorithm.

1. Introduction

Image Processing is an image analysis and processing process that uses a visual perception process, in the image processing process it has input, processes and produces output in the form of digital images and is commonly called as a two-dimensional image processing using a computer. Even though an image is rich in information, our image often experiences a decrease in quality intensity, for example, contains defects or noise, the color is too contrasting or blurry. Of course, an image like this will be difficult to represent so that the available information is reduced. In order for the disturbed image to be easily represented, the image needs to be manipulated into another image with better quality. (Winarno, 2011). Basically, the edge is something that limits between two areas that are similar or homogeneous. The image edge detector is generally used on computers to detect changes in the intensity of the image (image) and is also used to find the boundary of the object in the image (image). Edge detection significantly reduces the amount of data and filters out useless information, while preserving important structural properties in an image. In noise-free images, edge detection is helpful. (Mulawarman et al., 2016)

2. Research Methodology

Digital image is a representation of the light intensity function in discrete form on a two-dimensional plane. The image is composed of a set of pixels (picture elements) which have coordinates (x, y) and an amplitude of f(x, y). The coordinates (x, y) indicate the location/position of pixels in an image, while the amplitude f(x, y) indicates the color intensity value of the image. Edge detection is used to obtain objects. Edge detection takes advantage of drastic changes in intensity values at the boundary of two areas. The definition of the edge here is "a set of connected pixels that lie on two boundaries of two areas". Edge detection can be divided into two groups. Edge is a sudden (large) change in the intensity of the gray level in a short distance. There are three kinds of edges in a digital image, namely: Steep Edge

2.1 Steep Edge

A steep edge is an edge with a sharp change in intensity. The edge direction is around 90°. Figure 1 shows the edge detection of the steep edge type.

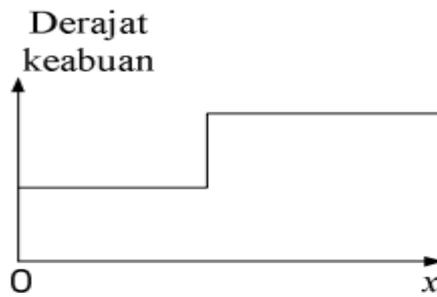


Fig. 1. Steep Edge

2.2 Sloping Edge

A sloping edge is an edge with a small directional angle. A sloping edge can be thought of as consisting of a number of adjacent local edges. Figure II.2 shows the edge detection of the sloping edge type.

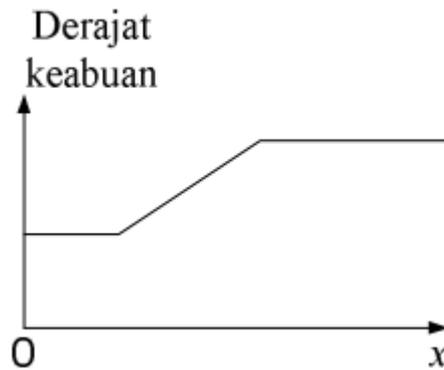


Fig. 2. Sloping Edge

3. Analysis And Design

3.1. Analysis

Software development consists of several stages, one of which is the system analysis stage. Analysis is a thinking activity to describe a thing into certain parts or components so that the characteristics or signs in each explanation can be known, the relationship between one component to another component, and also the function of each component. The Canny algorithm is an ingenious edge detection with a technique that extracts useful structural information from different visual objects and dramatically reduces the amount of data to be processed. It has been widely applied in various computer vision systems. Canny has found that the requirements for implementing edge detection across diverse vision systems are relatively the same. Thus, edge detection solutions to address these requirements can be implemented in various situations.

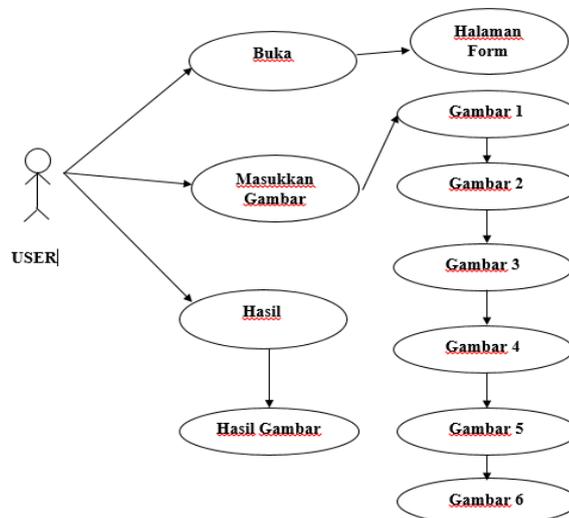


Fig. 3 Use Case Diagram User

3.2. Design

Interface design describes the display plan of each form that will be used in the actual application display.

1. Design Form Main Menu

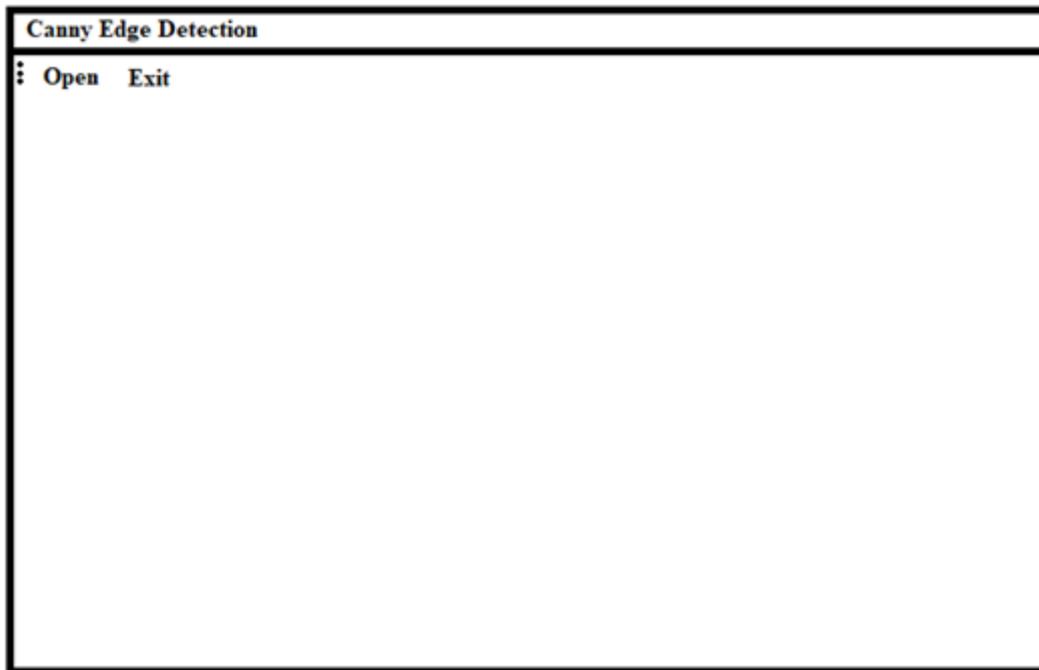


Fig. 4 Design Main Menu

2. Form Input Gambar Canny

This form is the main page displaying the image to be detected. The screen below shows canny image input.

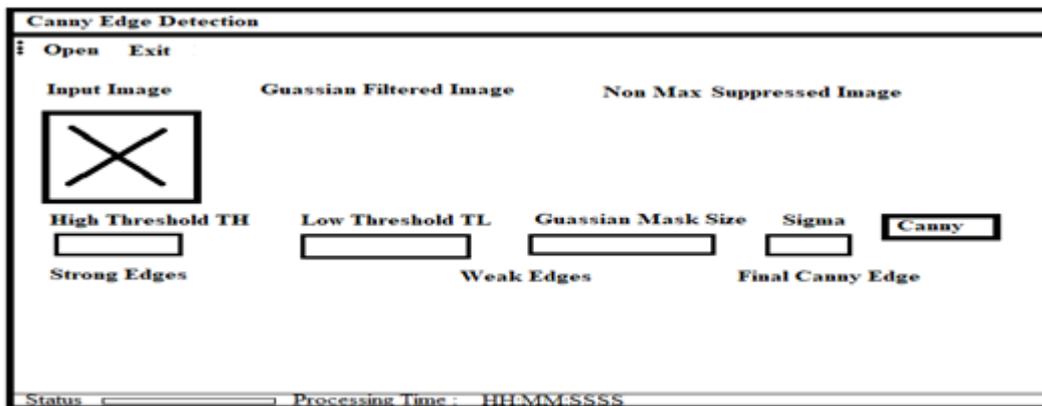


Fig. 5 Form Input Image

3. Form Menu Result For Canny Edge Detection

In the interface, these results show the results of Canny Edge Detection, when the Canny button is clicked, the results are as follows:

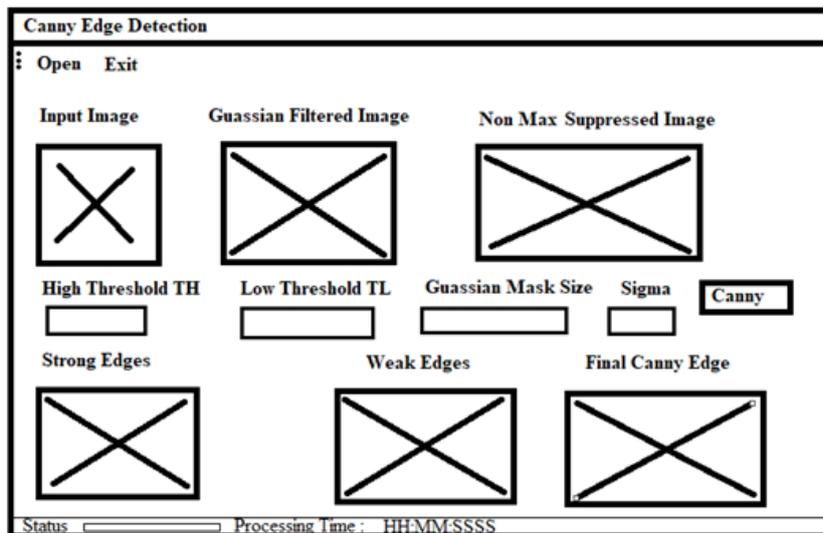


Fig. 6 Form Menu Result Of Canny Edge Detection

4. Form Menu-menu

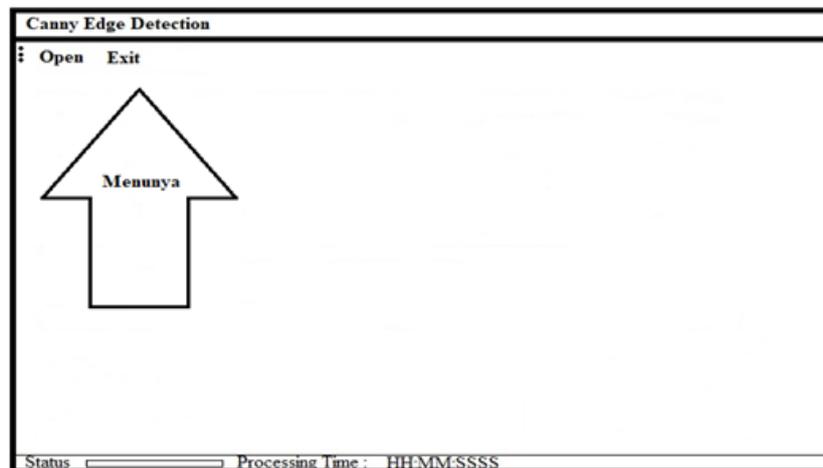


Fig.7 Form Menu

4. Results and Discussion

4.1 Results

The following are the results of the system when it is run on a computer, while the pictures below are screenshots, including;

a. Main Menu

The main window is the first display that is displayed, namely the open, exit, button canny, High Threshold, Low Threshold, Gaussian Mask Size and other filters such as: Input Image, Guassian Filtered Image, Non Max Suppressed Image, Strong Edges, Weak Edges , Final Canny Edges, Status, and Processing Time.

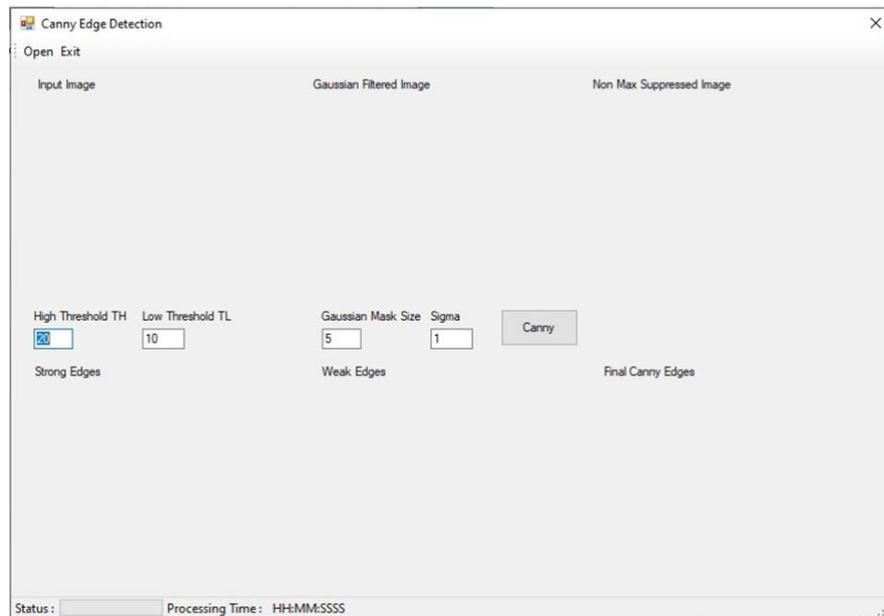


Fig. 8 Main Menu Form

b. Form Open

This open view is a view where when clicked the open file section appears on our device or the computer we use, and after we select the image or photo we have it will appear in the Input Image section. In the image section below are the results when the selected image or photo in the open file then appears in the Input Image section.

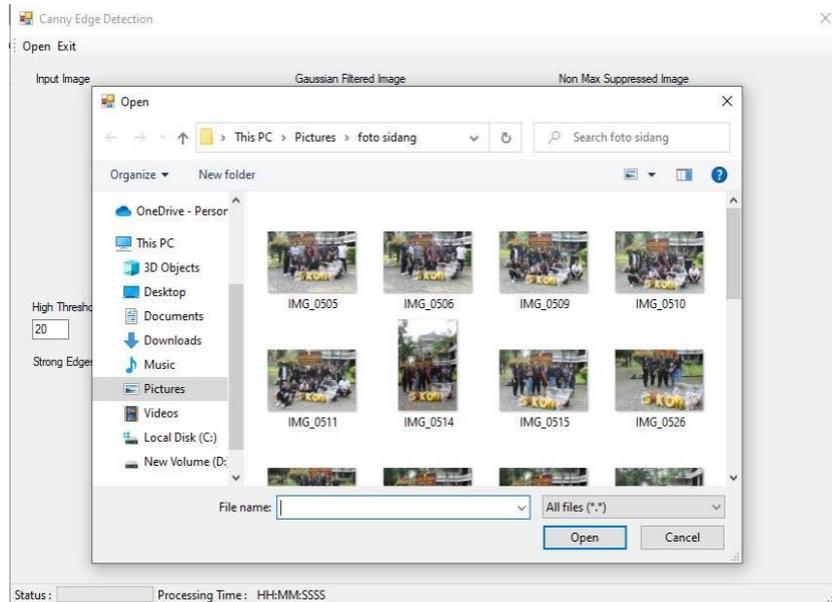


Fig. 9 Form Open File

c. Form Input File

This Input File is a view where when clicked the open file section appears on our device or the computer we use, and after we select the image or photo we have it will appear in the Input Image section. In the image section below are the results when the selected image or photo in the input file then appears in the Input Image section

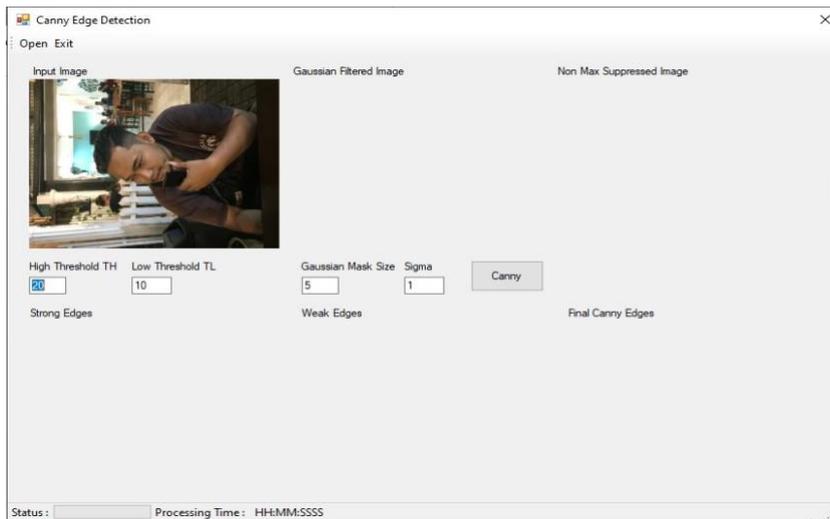


Fig. 10 Form Input File

d. Form High Threshold pada Canny Edge Detection

Form High Threshold in this program as a provider of high threshold sizes on images or photos that will be processed on the canny.



Fig. 11 Form High Threshold pada Canny Edge Detection

e. Form Low Threshold On Canny Edge Detection

Form Low Threshold in this program as a provider of low threshold sizes on images or photos that will be processed on the canny.



Fig. 12 Form Low Threshold On Canny Edge Detection

f. Form Gaussian Mask Size pada Canny Edge Detection

Form Gaussian Mask Size in this program as a provider of Gaussian mask sizes whose Gaussian sizes are limited to only 5 on the image or photo that will be processed on the canny.



Fig. 13 Form Gaussian Mask Size On Canny Edge Detection

g. Form Sigma On Canny Edge Detection

The Sigma form on this program is a provider of sigma sizes for images or photos that will be processed on the canny.



Fig. 14 Form Sigma On Canny Edge Detection

h. Form Button Canny On Canny Edge Detection

The Canny button in this program functions to carry out the processing process for all filters, if the canny button is clicked, the results of all filters will come out.

The screenshot shows a form with four input fields and a button. The fields are labeled 'High Threshold TH', 'Low Threshold TL', 'Gaussian Mask Size', and 'Sigma'. The values entered are 1, 10, 5, and 1 respectively. To the right of these fields is a button labeled 'Canny'.

Fig. 15 Form Button Canny On Canny Edge Detection

i. Form Status

This Status Display serves to display the running process of the filter on the canny edge detection program that is being run.

The screenshot shows a status bar with a green progress bar on the left and the text 'Status : Processing Time : 00:00:17.3442518' on the right.

Fig. 16 Form Status

j. Form Processing Time

Tampilan Processing Time ini berfungsi untuk menampilkan proses lamanya waktu yang digunakan selama jalannya proses filterisasi.

The screenshot shows a status bar with a green progress bar on the left and the text 'Status : Processing Time : 00:00:17.3442518' on the right.

Fig. 17 Form Processing Time

k. Form Output 1 Canny Edge Detection

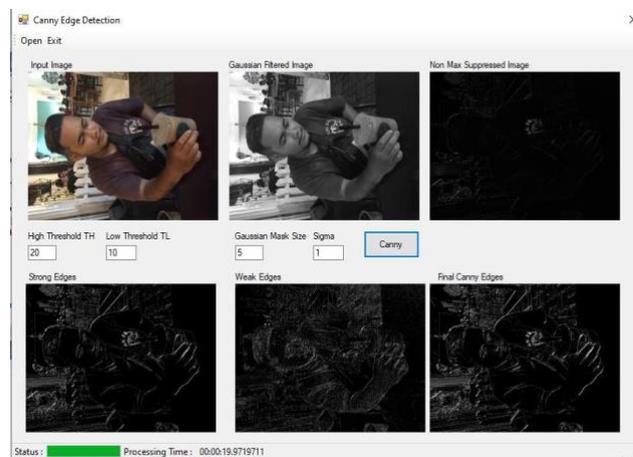


Fig. 18 Form Output 1 Canny Edge Detection

l. Form Output 2 Canny Edge Detection

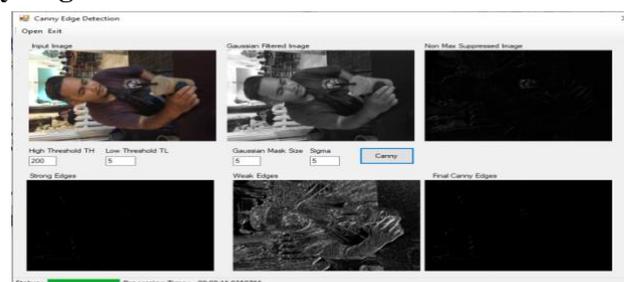


Fig. 19 Form Output 2 Canny Edge Detection

m. Form Output 3 Canny Edge Detection

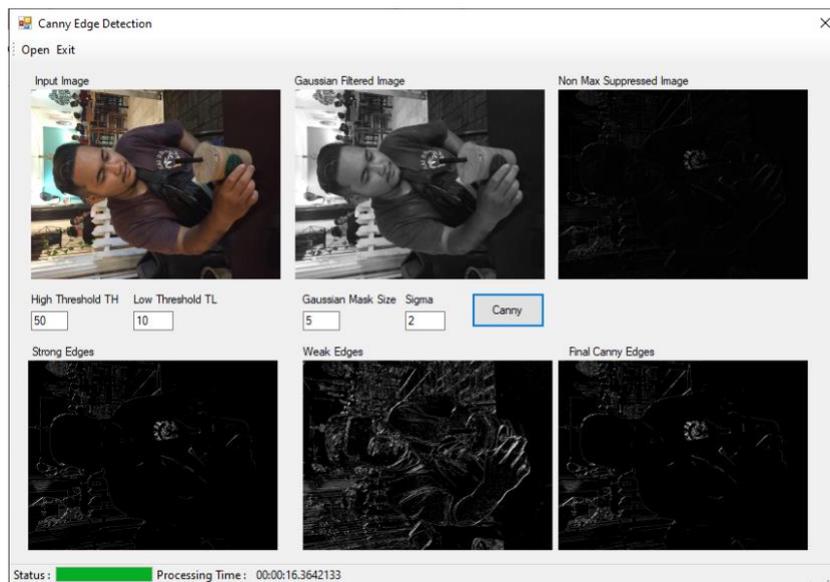


Fig. 20 Form Output 3 Canny Edge Detection

Figure in Figure 13. inputting with High Threshold with a value of 50, Low Threshold with a value of 10, Gaussian Mask Size with a value of 5, and Sigma which is given a value of 2 then the results are as shown above with this, the results of all outputs must be different.

4. CONCLUSION

1. From the results of the discussion that the author has done, the following conclusions can be drawn:
2. 1. Successfully made an image edge detection application with JPEG format.
3. 2. Successfully applied the canny algorithm for image edge detection applications with JPEG format.
4. 3. This application can display the outline information on the image.
5. 4. The application allows the user to know the outline of the image.

References

- [1] Amelia, L., & Marwanti, R. (2013). *Perbandingan Metode Roberts Dan Sobel Dalam Mendeteksi Tepi Suatu Citra Digital. 1*, 1–16.
- [2] Christopher Danil (2011) 'Edge Detection dengan Algoritma Canny', *SpringerReference*. doi: 10.1007/springerreference_64846.
- [3] Dianta, E. G. (2012). Deteksi Tepi Menggunakan Metode Canny Dengan Matlab Untuk Membedakan Uang Asli dan Uang Palsu. *Jurnal Jurusan Teknik Informatika, Fakultas Teknologi, Industri Universitas Gunadarma, 1*(1), 1–13.
- [4] Herawati, D. and Kardian, A. R. (2018) 'Analisis Deteksi Tepi Pada Citra Digital Berbasis JPG Dengan Operator Canny Menggunakan Matrix Laboratory', *Jurnal Ilmiah Komputasi*, 17(3), pp. 191–208.
- [5] Hermana, A. N., & Juerman, M. S. (2014). Implementasi Algoritma Canny dan Backpropagation dalam Pengenalan Pola Rumah Adat. *Jurnal Itenas*, 1–10.
- [6] Kezia Satyawati, Junius Karel Tampubolon, R. G. S. (2016). Analisis Gaussian Dan Edge Connection Dalam. *Informatika*, 12(1), 33–42.
- [7] Laxmi, G. F., Eosina, P., & Fatimah, F. (2018). Implementasi Penggabungan Prewitt dan Canny Edge Detection untuk Identifikasi Ikan Air Tawar. *Krea-Tif*, 6(2), 120. <https://doi.org/10.32832/kreatif.v6i2.2185>
- [8] Mulawarman, J. I., Komputer, S., Informasi, F. T., Andalas, U., Manis, K. L., & Tepi, D. (2016). Perbandingan Kinerja Metode Deteksi Tepi Pada Pengenalan Objek Menggunakan *OpenCV. 11*(2), 17–21.
- [9] Scott W.Ambler (2001) 'UML (Unified Modelling Language) Diagram', *Pengenalan UML*, (January), p. 43.
- [10] Winarno, E. (2011) 'Aplikasi Deteksi Tepi pada Realtime Video Menggunakan Algoritma Canny Detection', *Jurnal Teknologi Informasi DINAMIK*, 16(1), pp. 5–10.
- [11] Yesputra, R. (2013) 'Rolly Yesputra, M.Kom | Organisasi Komputer | 2013 6', pp. 6–22.