



# PENERAPAN METODE PREWITT, CANNY DAN SOBEL PADA PROSES DETEKSI TEPI CITRA

Polinus Laia

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia  
Jalan Sisingamangaraja No. 338, Medan, Indonesia

## Abstrak

Deteksi tepi merupakan langkah pertama untuk melengkapi informasi di dalam citra dimana tepi mencirikan batas – batas objek yang berguna untuk proses identifikasi. Banyak metode dalam deteksi tepi, namun dalam penelitian ini metode yang diambil yaitu metode prewitt, metode canny dan metode sobel untuk mendeteksi tepi objek pada citra digital. Sistem ini diterapkan ketiga metode tersebut untuk meningkatkan penampakan garis batas suatu objek. Penelitian ini membahas tentang cara penerapan ketiga metode tersebut dalam aplikasi matlab 7.1. Tujuan dan manfaat aplikasi ini adalah untuk mengenal penerapan hasil dari deteksi tepi serta memperoleh informasi awal mengenai fitur deteksi tepi untuk pengenalan objek bagi peneliti selanjutnya.

**Kata Kunci:** Deteksi Tepi, Prewitt, Canny, Sobel

## Abstract

Edge detection is the first step to completing information in an image where the edges characterize object boundaries that are useful for the identification process. There are many methods of edge detection, but in this study, the methods taken are the Prewitt method, the CANNY method and the Sobel method to detect the edges of objects in digital images. This system applies all three methods to improve the appearance of the boundary line of an object. This study discusses how to apply the three methods in the application Matlab 7.1. The purpose and benefits of this application are to recognize the application of the results of edge detection and obtain initial information about the edge detection features for object recognition for future researchers.

**Keywords:** Edge Detection, Prewitt, Canny, Sobel

## 1. PENDAHULUAN

Permasalahan yang sering muncul dalam citra adalah *vignetting*. *Vignetting* adalah gejala munculnya area gelap pada tepi citra, terutama di pojok-pojoknya. Kondisi ini terjadi saat diameter citra yang dihasilkan lensa lebih kecil dari sensor yang disebabkan oleh tiga hal yaitu dengan penggunaan lensa yang tidak tepat pada kamera dengan sensor yang lebih besar, halangan fisik pada sisi muka lensa (yang biasanya berupa penggunaan filter yang terlalu tebal) dan kualitas desain lensa yang kurang optimal. Salah satu solusi penyelesaian masalah tersebut adalah melalui deteksi tepi.

Deteksi tepi sangat penting dalam pengolahan citra karena pendeteksian tepi merupakan langkah pertama untuk melengkapi informasi di dalam citra. Tepi mencirikan batas-batas objek, karena itu tepi berguna untuk proses segmentasi dan identifikasi objek dalam citra. Tujuan pendeteksian tepi adalah untuk meningkatkan penampakan garis batas suatu daerah atau objek di dalam citra (Puji Triono, Murinto, 2015, 1116).

Metode *prewitt* menggunakan persamaan yang sama dengan operator *sobel*, hanya saja konstanta  $c$  yang digunakan bernilai 1. Operator *prewitt* tidak menekankan pembobotan pada piksel - piksel yang lebih dekat dengan titik pusat kernel (T. Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto, 2009). Pengembangan dari *gradient* operator dengan menggunakan 2 *mask* (*horizontal* dan *vertikal*) ukuran 3x3. Operator ini kekuatan *gradient* ditinjau dari sudut pandang *horizontal* dan *vertikal* memperhatikan titik disekitar pada posisi *horizontal* dan *vertikal*.

Metode *canny* yang dikemukakan oleh John Canny pada tahun 1986, terkenal sebagai operator deteksi tepi yang optimal. Algoritma ini memberikan tingkat kesalahan rendah, melokalisasi titik-titik tepi (jarak piksel-piksel yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek) dan hanya memberikan satu tanggapan untuk satu tepi (Abdul Kadir, Adhi Susanto, 2013). Metode *sobel* merupakan salah satu operator yang menghindari adanya perhitungan *gradient* di titik interpolasi. Operator ini menggunakan kernel ukuran 3x3 piksel untuk perhitungan *gradient* sehingga perkiraan *gradient* berada tepat di tengah jendela (T. Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto, 2009). Biasanya operator *sobel* menempatkan penekanan atau pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat jendela, sehingga pengaruh piksel-piksel tetangga akan berbeda sesuai dengan letaknya terhadap titik di mana *gradient* dihitung. Susunan nilai-nilai pembobotan pada jendela terlihat bahwa perhitungan terhadap *gradient* juga merupakan gabungan dari posisi *horizontal* dan posisi *vertikal*.

Mendeteksi tepi citra merupakan langkah untuk menyelesaikan permasalahan *vignetting* pada gambar (citra) dengan menerapkan metode *prewitt*, *canny* dan *sobel*. Penerapan tiga metode ini dilakukan secara individu (satu persatu) dengan tujuan yang sama adalah mendapatkan tepi citra. Penerapan metode *prewitt*, *canny* dan *sobel* dalam sebuah aplikasi deteksi tepi citra berbasis GUI (*Grafic User Interface*) dari Matlab 7.1 dapat membantu



mendeteksi tepi citra untuk mengambil informasi permasalahan *vignetting* dalam sebuah citra sehingga ada pertimbangan kepada pengelolah memperbaiki kualitas gambar (citra).

## 2. TEORITIS

### 2.1 Deteksi Tepi

Deteksi tepi berfungsi untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area. Defenisi tepi di sini adalah himpunan piksel yang terhubung yang terletak pada batas dua area. Perlu diketahui, tepi sesungguhnya mengandung informasi sangat penting. Informasi yang diperoleh dapat berupa bentuk maupun ukuran objek (Abdul Kadir, Adhi Susanto, 2013).

### 2.2 Metode Prewitt

Metode *prewitt* menggunakan persamaan yang sama dengan operator *sobel*, hanya saja konstanta  $c$  yang digunakan bernilai 1. Operator *prewitt* tidak menekankan pembobotan pada piksel - piksel yang lebih dekat dengan titik pusat kernel. Metode ini mengambil prinsip dari fungsi *laplacian* yang dikenal sebagai fungsi untuk membangkitkan *High Pass Filter* (T. Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto, 2009).

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \text{ dan } S_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

### 2.3 Metode Canny

Metode *canny* yang dikemukakan oleh *John Canny* pada tahun 1986, terkenal sebagai operator deteksi tepi yang optimal. Algoritma ini memberikan tingkat kesalahan rendah, melokalisasi titik-titik tepi (jarak piksel-piksel yang ditemukan deteksi dan tepi yang sesungguhnya sangat pendek) dan hanya memberikan satu tanggapan untuk satu tepi (Abdul Kadir, Adhi Susanto, 2013).

### 2.4 Metode Sobel

Metode *sobel* merupakan salah satu operator yang menghindari adanya perhitungan *gradient* di titik interpolasi. Operator ini menggunakan kernel ukuran 3x3 piksel untuk perhitungan *gradient* sehingga perkiraan *gradient* berada tepat di tengah jendela. Misalkan susunan piksel- piksel di sekitar piksel  $x, y$  (T. Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto, 2009) adalah:

$a_0$	$a_1$	$a_2$
$a_7$	$(x, y)$	$a_3$
$a_6$	$a_5$	$a_4$

Berdasarkan susunan piksel tetangga tersebut besaran *gradient* yang dapat dihitung menggunakan operator *sobel* (T. Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto, 2009).

## 3. ANALISA DAN PEMBAHASAN

Tahapan yang digunakan untuk mendeteksi tepi citra menggunakan ketiga operator *prewitt*, *canny* dan *sobel* yaitu tahap ekstraksi warna, tahap *cropping*, tahap deteksi tepi menggunakan metode *prewitt*, *canny* dan *sobel*, tahap operasi dilatasi citra, tahap pengisian objek, pembuatan garis tepi untuk menyelesaikan permasalahan *vignetting*. Prosedur dalam mendeteksi tepi citra sebagai berikut:

1. *Image Acquisition*  
 Akuisisi citra merupakan tahap awal, pada tahap ini citra akan diambil/ditangkap. Kamera akan dipergunakan untuk tahapan ini. Kamera yang umum dipergunakan pada tahap ini, dapat berupa kamera digital.
2. *Pra Processing*  
 Pada Tahap ini dilakukan proses *grayscale* terhadap citra yang akan dideteksi tepi citranya.
  - 1) Proses *grayscale* : proses untuk mengubah gambar yang memiliki warna menjadi gambar yang memiliki tingkat warna ke abu-abuan (*grayscale*).
  - 2) Gambar yang akan di-*gray-scilling* nilai tiap titik akan disamakan dengan nilai red-green-blunya sehingga untuk tiap titik hanya memiliki 1 nilai saja, yang disebut nilai *gray-level*-nya.
  - 3) Pada dasarnya proses ini dilakukan dengan meratakan nilai pixel dari 3 nilai RGB menjadi 1 nilai.
3. *Segmentasi (Segmentation)*  
 Pada tahap ini citra yang telah diubah menjadi *grayscale* akan dideteksi tepi citranya menggunakan operator *prewitt*, *canny* dan *sobel*.
4. *Post Processing*



Pada tahap ini akan dihasilkan citra yang berupa citra yang hanya terdiri dari gambar hitam putih yang memiliki garis tepi citra.

Berikut merupakan tabel grayscale citra.

Tabel 1. Hasil Proses Perubahan Citra RGB ke *Grayscale*

51	37	40	41	45	46	46	45
39	41	42	43	43	40	38	32
39	43	42	41	37	31	32	17
44	46	41	34	33	37	45	30
45	57	37	32	35	42	55	52
53	48	39	41	47	45	51	56
52	47	47	50	55	58	55	57
50	49	51	52	52	54	55	55

### 3.1 Penerapan Operator *Prewitt*

Berikut adalah hasil perhitungan perkalian matrix penggunaan operator *prewitt* untuk mengetahui tepi sebuah gambar citra dengan matrix 8 x 8. Konvolusi pertama dilakukan terhadap piksel (titik pusat mask).

Tabel 2. Nilai Citra Grayscale Matrix 8 x 8

51	37	40	41	45	46	46	45
39	41	42	43	43	40	38	32
39	43	42	41	37	31	32	17
44	46	41	34	33	37	45	30
45	57	37	32	35	42	55	52
53	48	39	41	47	45	51	56
52	47	47	50	55	58	55	57
50	49	51	52	52	54	55	55

- Konvolusi pertama dilakukan terhadap piksel yang bernilai 41 (titik pusat mask):  
 $S_x = (40)(1)+(1)(42)(1)+(42)(1) - (51)(-1)+(1)(39)(-1)+(39)(-1) = 253$   
 $S_y = (51)(1)+(1)(37)(1)+(40)(1) - (39)(-1)+(1)(43)(-1)+(42)(-1) = 252$   
 Maka:  $M = \sqrt{S_x^2 + S_y^2} = 505 = 22$
- Konvolusi kedua dilakukan terhadap piksel yang bernilai 42 (titik pusat mask):  
 $S_x = (41)(1)+(1)(43)(1)+(41)(1) - (37)(-1)+(1)(41)(-1)+(43)(-1) = 246$   
 $S_y = (37)(1)+(1)(40)(1)+(41)(1) - (43)(-1)+(1)(42)(-1)+(41)(-1) = 244$   
 Maka:  $M = \sqrt{S_x^2 + S_y^2} = 490 = 22$
- Konvolusi kedua dilakukan terhadap piksel yang bernilai 42 (titik pusat mask):  
 $S_x = (45)(1)+(1)(43)(1)+(37)(1) - (40)(-1)+(1)(42)(-1)+(42)(-1) = 249$   
 $S_y = (40)(1)+(1)(41)(1)+(45)(1) - (42)(-1)+(1)(41)(-1)+(37)(-1) = 247$   
 Maka:  $M = \sqrt{S_x^2 + S_y^2} = 496 = 22$
- Konvolusi kedua dilakukan terhadap piksel yang bernilai 42 (titik pusat mask):  
 $S_x = (46)(1)+(1)(40)(1)+(31)(1) - (41)(-1)+(1)(43)(-1)+(41)(-1) = 242$   
 $S_y = (41)(1)+(1)(45)(1)+(46)(1) - (41)(-1)+(1)(37)(-1)+(31)(-1) = 241$   
 Maka:  $M = \sqrt{S_x^2 + S_y^2} = 487 = 22$
- Konvolusi selanjutnya sama dengan konvolusi satu, dua, tiga dan empat. Setelah di konvolusikan, maka hasil nilainya terlihat pada tabel berikut:

Tabel 3. Hasil Konvolusi Tepi Citra Matrix 8 x 8

51	37	40	41	45	46	46	45
39	22	22	22	22	22	22	32
39	22	22	22	22	22	22	17
44	22	22	22	22	22	22	30
45	22	22	22	22	22	22	52
53	22	22	22	22	22	22	56
52	22	22	22	22	22	22	57
50	49	51	52	52	54	55	55



Maka hasil deteksi tepi yang didapat dari perhitungan matrix untuk mengetahui segmentasi pada gambar menggunakan metode *prewitt*, hasil yang didapat pada perkalian matrix 3x3 dengan citra matrix 8 x 8 pada tabel diatas.

### 3.2 Penerapan Operator Canny

Berikut adalah perhitungan perkalian matrix penggunaan operator *canny* untuk mengetahui masalah *vignetting* pada citra dengan matrix 8 x 8. Konvolusi pertama dilakukan terhadap piksel (titik pusat mask).

Tabel 4. Contoh Nilai Citra Matrix 8 x 8

51	37	40	41	45	46	46	45
39	41	42	43	43	40	38	32
39	43	42	41	37	31	32	17
44	46	41	34	33	37	45	30
45	57	37	32	35	42	55	52
53	48	39	41	47	45	51	56
52	47	47	50	55	58	55	57
50	49	51	52	52	54	55	55

Menghitung nilai konvolusi pada citra

Persyaratan Konvolusi Citra:

- Jika hasil konvolusi nilai piksel negatif, maka nilai dijadikan nol
  - Jika hasil konvolusi nilai piksel > nilai keabuan maksimum 255, maka nilai dijadikan kenilai keabuan maksimum 255.
  - Meng-konvolusi piksel pinggir border diabaikan, sehingga nilai piksel pinggir = nilai pada citra semula.
- Konvolusi pertama dilakukan terhadap piksel yang bernilai 41 (titik pusat mask):  
 $G_x = (40)(-1)+(2)(42)(-1)+(42)(-1) - (51)(1)+(2)(39)(1)+(39)(1) = -332$   
 $G_y = (51)(-1)+(2)(37)(-1)+(40)(-1) - (39)(1)+(2)(43)(1)+(42)(1) = -332$   
 $|G| = |G_x|+|G_y| = 0$
  - Konvolusi kedua dilakukan terhadap piksel yang bernilai 42 (titik pusat mask):  
 $G_x = (41)(-1)+(2)(43)(-1)+(41)(-1) - (37)(1)+(2)(41)(1)+(43)(1) = -248$   
 $G_y = (37)(-1)+(2)(40)(-1)+(41)(-1) - (43)(1)+(2)(42)(1)+(41)(1) = -326$   
 $|G| = |G_x|+|G_y| = -80$ . Maka nilai menjadi = 0
  - Konvolusi kedua dilakukan terhadap piksel yang bernilai 43 (titik pusat mask):  
 $G_x = (45)(-1)+(2)(43)(-1)+(37)(-1) - (40)(1)+(2)(42)(1)+(42)(1) = -334$   
 $G_y = (40)(-1)+(2)(41)(-1)+(45)(-1) - (42)(1)+(2)(41)(1)+(37)(1) = -328$   
 $|G| = |G_x|+|G_y| = 06$
  - Konvolusi selanjutnya sama dengan konvolusi satu, dua, dan tiga dan. Setelah di konvolusikan, maka hasil nilainya terlihat pada tabel berikut:

Tabel 5. Hasil Konvolusi Citra Menggunakan Operator Canny

51	37	40	41	45	46	46	45
39	0	0	6	6	6	0	32
39	0	0	6	6	6	0	17
44	0	6	0	0	0	0	30
45	0	6	0	0	6	6	52
53	6	0	6	0	6	6	56
52	6	6	0	0	6	0	57
50	49	51	52	52	54	55	55

Maka hasil deteksi tepi yang didapat dari perhitungan matrix untuk mengetahui nilai pada gambar 3.4 menggunakan metode *canny*, hasil yang didapat pada perkalian *matrix* 3 x 3 dengan citra *matrix* 8 x 8.

### 3.3 Penerapan Operator Sobel

Berikut adalah hasil perhitungan perkalian matrix penggunaan operator *sobel* untuk mengetahui segmentasi sebuah gambar citra dengan matrix 8 x 8. Konvolusi pertama dilakukan terhadap piksel (titik pusat mask).

Tabel 6. Contoh Nilai Citra Matrix 8 x 8

51	37	40	41	45	46	46	45
----	----	----	----	----	----	----	----



39	41	42	43	43	40	38	32
39	43	42	41	37	31	32	17
44	46	41	34	33	37	45	30
45	57	37	32	35	42	55	52
53	48	39	41	47	45	51	56
52	47	47	50	55	58	55	57
50	49	51	52	52	54	55	55

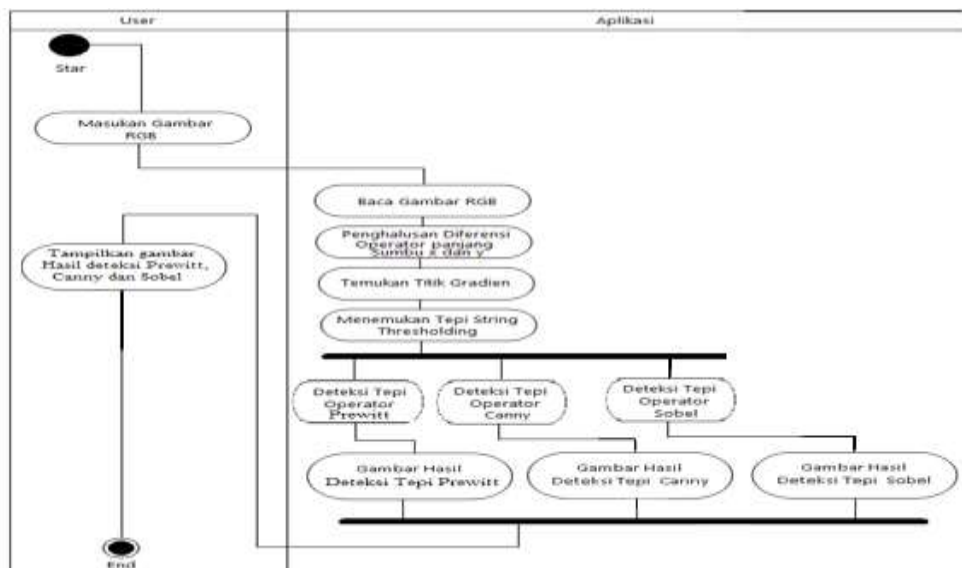
- Konvolusi pertama dilakukan terhadap piksel yang bernilai 41 (titik pusat mask):  
 $S_x = (40)(1)+(2)(42)(2)+(42)(1) - (51)(-1)+(2)(39)(-2)+(39)(-1) = 496$   
 $S_y = (51)(1)+(2)(37)(2)+(40)(1) - (39)(-1)+(2)(43)(-2)+(42)(-1) = 512$   
 Maka:  $M = \sqrt{S_x^2 + S_y^2} = 1008 = 32$
- Konvolusi kedua dilakukan terhadap piksel yang bernilai 42 (titik pusat mask):  
 $S_x = (41)(1)+(2)(43)(2)+(41)(1) - (37)(-1)+(2)(41)(-2)+(43)(-1) = 498$   
 $S_y = (37)(1)+(2)(40)(2)+(41)(1) - (43)(-1)+(2)(42)(-2)+(41)(-1) = 504$   
 Maka:  $M = \sqrt{S_x^2 + S_y^2} = 1002 = 32$
- Konvolusi ketiga dilakukan terhadap piksel yang bernilai 43 (titik pusat mask):  
 $S_x = (45)(1)+(2)(43)(2)+(37)(1) - (40)(-1)+(2)(42)(-2)+(42)(-1) = 504$   
 $S_y = (40)(1)+(2)(41)(2)+(45)(1) - (42)(-1)+(2)(41)(-2)+(37)(-1) = 492$   
 Maka:  $M = \sqrt{S_x^2 + S_y^2} = 996 = 32$
- Konvolusi selanjutnya sama dengan konvolusi satu, dua dan tiga. Setelah di konvolusikan, maka hasil nilainya terlihat pada tabel berikut:

Tabel 7. Hasil Konvolusi Tepi Citra Matrix 8 x 8

51	37	40	41	45	46	46	45
39	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	32
39	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	17
44	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	30
45	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	52
53	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	56
52	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	<b>32</b>	57
50	49	51	52	52	54	55	55

Maka hasil deteksi tepi yang didapat dari perhitungan matrix untuk mengetahui segmentasi pada gambar menggunakan metode *sobel*, hasil yang didapat pada perkalian matrix 3x3 dengan citra matrix 8x8 pada tabel diatas.

Berikut merupakan diagram activity dari perancangan deteksi tepi:



Gambar 1. Activity Diagram Proses Deteksi Tepi Citra



#### 4. KESIMPULAN

Hasil penelitian dapat ditarik kesimpulan dari deteksi tepi menggunakan 3 metode, adalah:

1. Prosedur dalam mendeteksi tepi itu sangat penting mulai dari pengambil gambar *praprocessing*, deteksi tepi hingga tahap terakhir mendapatkan tepi yaitu *post processing*. Perlu diperhatikan bahwa tahap *pra processing* sangat penting dimana citra harus terlebih dahulu diubah menjadi *grayscale*.
2. Penerapan metode *prewit*, *canny* dan *sobel* dilakukan satu persatu melalui prosedur - prosedur deteksi tepi. Langkah awal dalam penerapan ini adalah tahap analisa dimana setiap formula akan dilakukan perhitungannya dengan matriks 8x8 sebagai uji coba. Hasil dari perhitungan pada tahap analisa merupakan sebuah tepi citra. Tepi yang dihasilkan dari ketiga metode *prewitt*, *canny* dan *sobel*, lebih optimal tepi yang dihasilkan metode *canny* dimana garis tepinya lebih tipis dari tepi lainnya.

#### REFERENCES

- [1] Abdul Kadir, Adhi Susanto "Teori dan Aplikasi Pengelolaan Citra", Penerbit Andi, Yogyakarta, Edisi I, 2013.
- [2] Adi Nugroho, "Rekayasa Perangkat Lunak Berorientasi Objek" Penerbit Andi, Yogyakarta, Edisi I, 2010
- [3] Evi Triandini, I Gede Suardika, "Step By Step Desain Proyek Menggunakan UML", Penerbit Andi, Yogyakarta, Edisi I, 2012
- [4] Mohammad Yazdi Pusadan, "Pemrograman Matlab Pada Sistem Pakar Fuzzy", Penerbit Deepublish, Yogyakarta, Edisi I, 2014
- [5] Muhammad Arhami, S.Si., M.Kom, Anita Desiani, S.Si., M.Kom "Pemrograman Matlab", Penerbit Andi, Yogyakarta, Edisi I, 2005.
- [6] Prabowo Pudjo Widodo, Herlawati "Menggunakan UML", Penerbit Informatika, Bandung, 2011.
- [7] Susan B. Bastable, "Perawat Sebagai Pendidik" Penerbit EGC, Jakarta, 2012.
- [8] T. Sutoyo, Edy Mulyanto, Vincent Suhartono, Oky Dwi Nurhayati, Wijanarto "Teori Pengelolaan Citra Digital," Penerbit Andi, Yogyakarta, Edisi I, 2009.