BAB III

ANALISIS DAN PERANCANGAN

II.1 Analisis Sistem

Algoritma canny adalah salah satu operator yang digunakan untuk deteksi tepi pada citra, Operator ini mirip seperti operator sobel. Aloritma *canny* dilakukan dengan cara konvolusi dengan melibatkan kernel canny. Konvolusi dilakukan dengan nilai pixel dari citra yang akan diproses atau dideteksi proses citranya. Berikut ini nilai pixel citra yang akan diproses dengan mmelakukan konvolusi dari kernel *canny*, untuk mengambil nilai pixel bisa dilakukan dengan menggunakan tool matla, sebagai contoh berikut adalah nilai pixel pada citra.

$$\begin{pmatrix} 55 & 63 & 77 & 80 & 129 \\ 22 & 94 & 87 & 27 & 26 \\ 107 & 154 & 76 & 30 & 27 \\ 108 & 252 & 220 & 75 & 30 \\ 20 & 111 & 185 & 51 & 10 \end{pmatrix}$$

Setelah menentukan nilai pixel berikutnya adalah menentukan nilai kernel yang digunakan untuk mendeteksi citra.

$$X \quad \begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Berikutnya adalah menghitung nilai konvolusi dari citra dengan menggunakan kernel yang ada, berikut adalah hasilnya

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Hasil konvolusi = 82 Nilai ini dihitung dengan cara berikut :

(-1 x 55) + (0 x 63) + (1 x 77) + ($-\sqrt{2}$ x 22) + (0 x 94) + ($\sqrt{2}$ x 87) +(-1 x 107) +(0 x 154) +(1 x 76) = 82

	82			

1. Geser kernel satu pixel ke kanan, kemudian hitung nilai pixel pada posisi (0,0) dari kernel

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

20	111	185	51	10

Hasil konvolusi = -200.8 Nilai ini dihitung dengan cara berikut :

(-1 x 63) + (0 x 77) + (1 x 80) + ($-\sqrt{2}$ x 94) + (0 x 87) + ($\sqrt{2}$ x 27) +(-1 x 154) + (0 x 76) +(1 x 30) = -200.8

	82	-200.8		

2. Geser kernel satu pixel ke kanan, kemudian hitung nilai pixel pada posisi (0,0) dari kernel.

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Hasil konvolusi = -82.4 Nilai ini dihitung dengan cara berikut :

(-1 x 77) + (0 x 80) + (1 x 129) + ($-\sqrt{2}$ x 87) + (0 x 27) + ($\sqrt{2}$ x 26) +(-1 x 76) + (0 x 30) +(1 x 27) = -82.4

	82	-200.8	-82.4	

3. Selanjutnya geser kernel satu pixel ke bawah, lalu mulai lagi melakukan konvolusi dari sisi kiri citra. Setiap kali konvolusi, geser kernel satu pixel ke kanan.

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Hasil konvolusi = 133.6 Nilai itu dihitung dengan cara berikut :

(-1 x 22) + (0 x 94) + (1 x 87) + ($-\sqrt{2}$ x 107 + (0 x 154) + ($\sqrt{2}$ x 76) +(-1 x 108) + (0 x 252) +(1 x 220) = 133.6

	82	-200.8	-82.4	
	133.6			

4. Geser kernel satu pixel ke kanan, kemudian hitung nilai pixel pada posisi (0,0) dari kernel.

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Hasil konvolusi = -417.6 Nilai itu dihitung dengan cara berikut :

(-1 x 94) + (0 x 87) + (1 x 27) + (-$\sqrt{2}$ x 154) + (0 x 76) + ($\sqrt{2}$ x 30) +(-1 x 252) + (0 x 220) +(1 x 75) = -417.6

	82	-200.8	-82.4	
	133.6	-417.6		

5. Geser kernel satu pixel ke kanan, kemudian hitung nilai pixel pada posisi (0,0) dari kernel.

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Hasil konvolusi = -319.6 Nilai itu dihitung dengan cara berikut :

(-1 x 87) + (0 x 27) + (1 x 26) + ($-\sqrt{2}$ x 76) + (0 x 30) + ($\sqrt{2}$ x 27) +(-1 x 220) + (0 x 75) +(1 x 30) = -319.6

	82	-200.8	-82.4	
	133.6	-417.6	-319.6	

6. Dengan cara yang sama seperti tadi, maka pixel-pixel pada baris ke tiga dikonvolusi sehingga menghasilkan :

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Hasil konvolusi = 290.8 Nilai itu dihitung dengan cara berikut :

(-1 x 107) + (0 x 154) + (1 x 76) + ($-\sqrt{2}$ x 108) + (0 x 252) + ($\sqrt{2}$ x 220) +(-1 x 20) + (0 x 111) +(1 x 185) = 290.8

	82	-200.8	-82.4	
	133.6	-417.6	-319.6	
	290.8			

7. Geser kernel satu pixel ke kanan, kemudian hitung nilai pixel pada posisi (0,0) dari kernel.

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

Hasil konvolusi = -431.8 Nilai itu dihitung dengan cara berikut :

(-1 x 154) + (0 x 76) + (1 x 30) + ($-\sqrt{2}$ x 252) + (0 x 220) + ($\sqrt{2}$ x 75) +(-1 x 111) + (0 x 185) +(1 x 51) = -431.8

	82	-200.8	-82.4	
	133.6	-417.6	-319.6	
	290.8	-431.8		

8. Geser kernel satu pixel ke kanan, kemudian hitung nilai pixel pada posisi (0,0) dari kernel.

55	63	77	80	129
22	94	87	27	26
107	154	76	30	27
108	252	220	75	30
20	111	185	51	10

X

$$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$$

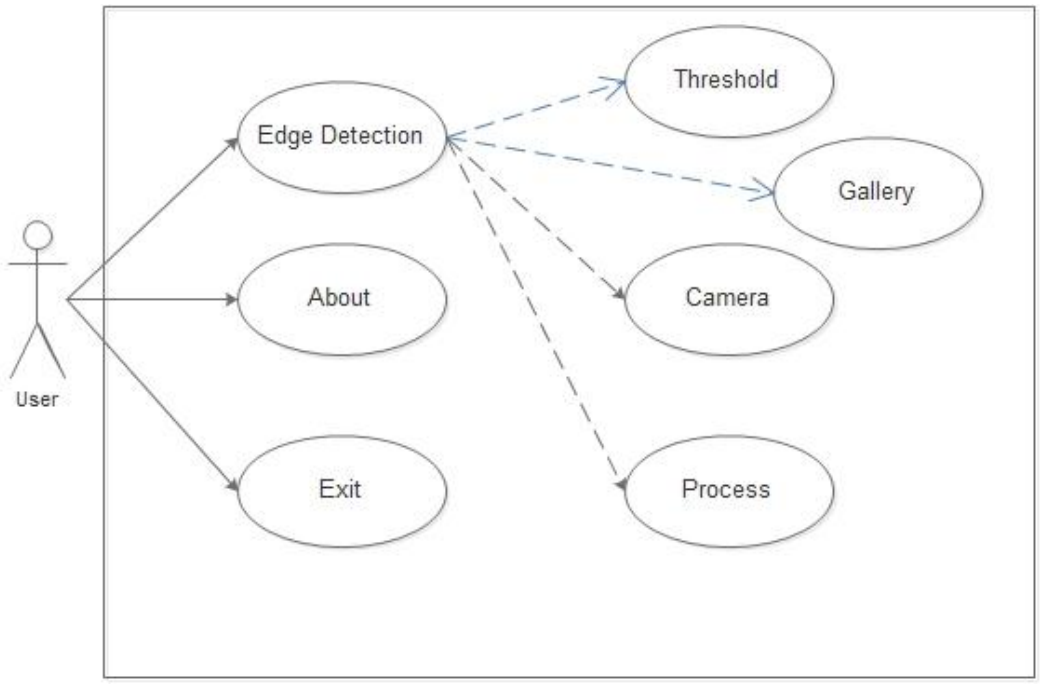
Hasil konvolusi = -490 Nilai itu dihitung dengan cara berikut :

(-1 x 76) + (0 x 30) + (1 x 27) + ($-\sqrt{2}$ x 220) + (0 x 75) + ($\sqrt{2}$ x 30) +(-1 x 185) + (0 x 51) +(1 x 10) = -490

55	63	77	80	129
22	82	-200.8	-82.4	26
107	133.6	-417.6	-319.6	27
108	290.8	-413.8	-490	30
20	111	185	51	10

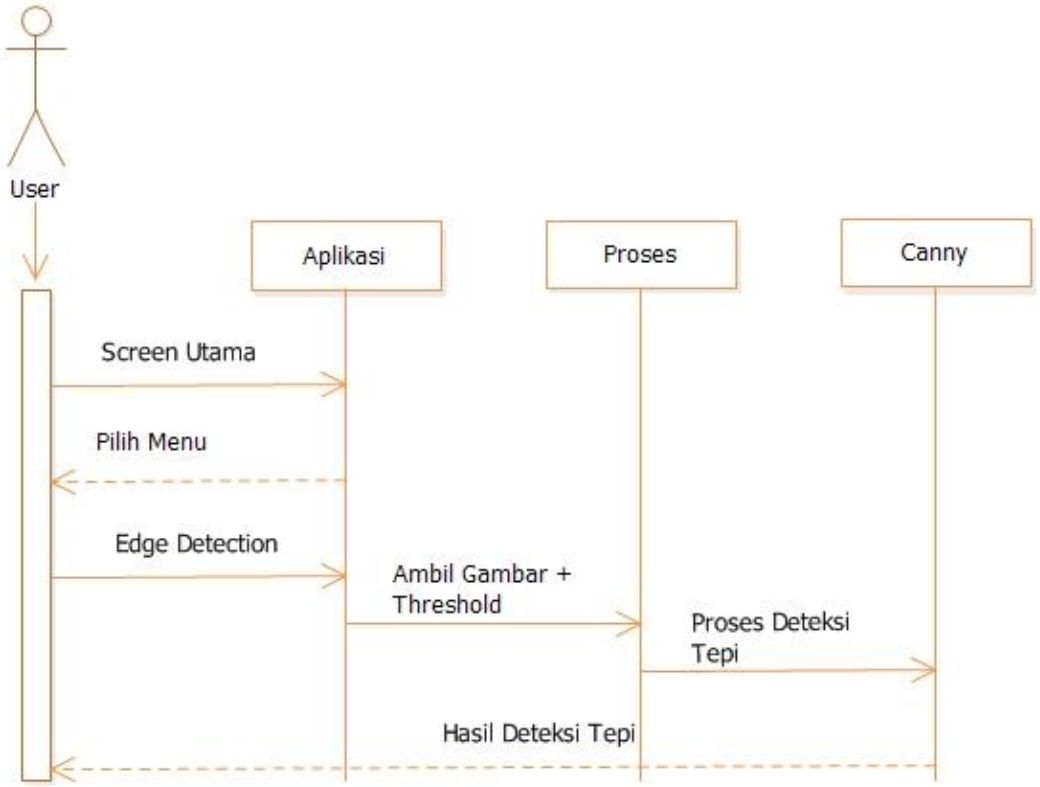
III.2 Use Case Diagram

Use case diagram untuk sistem deteksi tepi citra ini memilki entitas yaitu pengguna sistem dan beberapa objek diagram, use case diagram dapat dilihat pada gambar III.1 berikut ini:



Gambar III.1 Use Case Diagram Sistem Deteksi Tepi

Use case diagram hanyalah diagram yang menggambarkan secara umum dari sistem deteksi tepiyang dirancang, untuk lebih jelas bagaimana penggunaan sistem ini maka dibuatlah sequence diagram yang menggambarkan secara terperinci penggunaan sistem ini.



Gambar III.2 Sequence Diagram Deteksi Tepi

III.3 Perancangan Tampilan

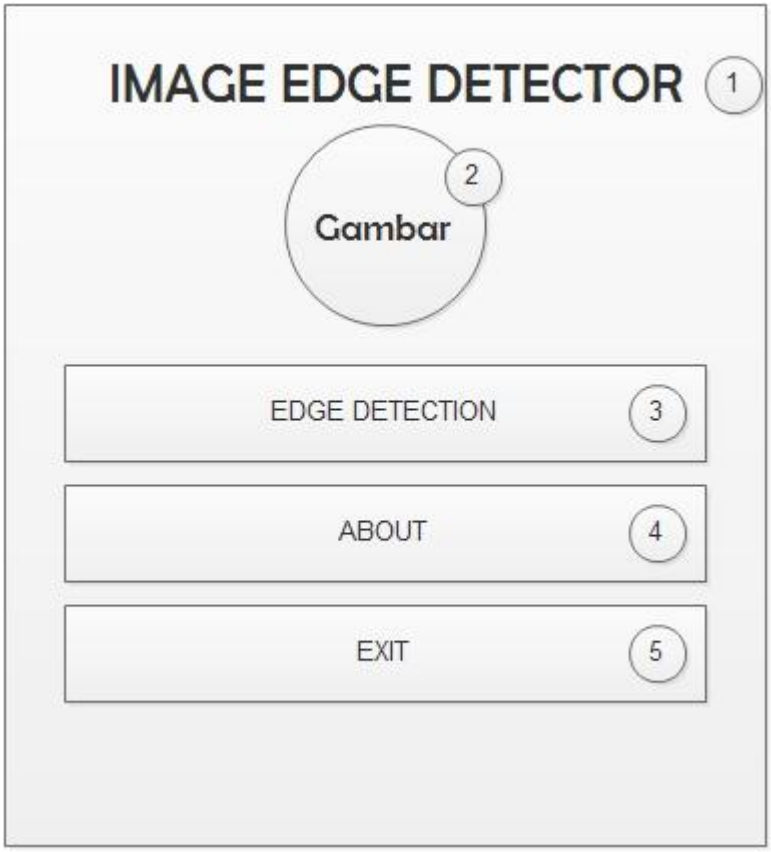
Perangkat lunak pembelajaran ini dirancang dengan menggunakan bahasa pemrograman *Android* dengan menggunakan beberapa komponen (*tools*) yang dimiliki. Dalam perancangan perangkat lunak ini, penulis juga menggunakan beberapa gambar sebagai tambahan untuk mempercantik aplikasi

Form – form yang terdapat dalam perangkat lunak pembelajaran ini yaitu,

1. *Form* Utama
2. *Form About*
3. *Form Edge Detection*

III.3.1 Form Utama

Form utama merupakan perancangan desain ketika aplikasi dijalankan, berikut adalah desainnya



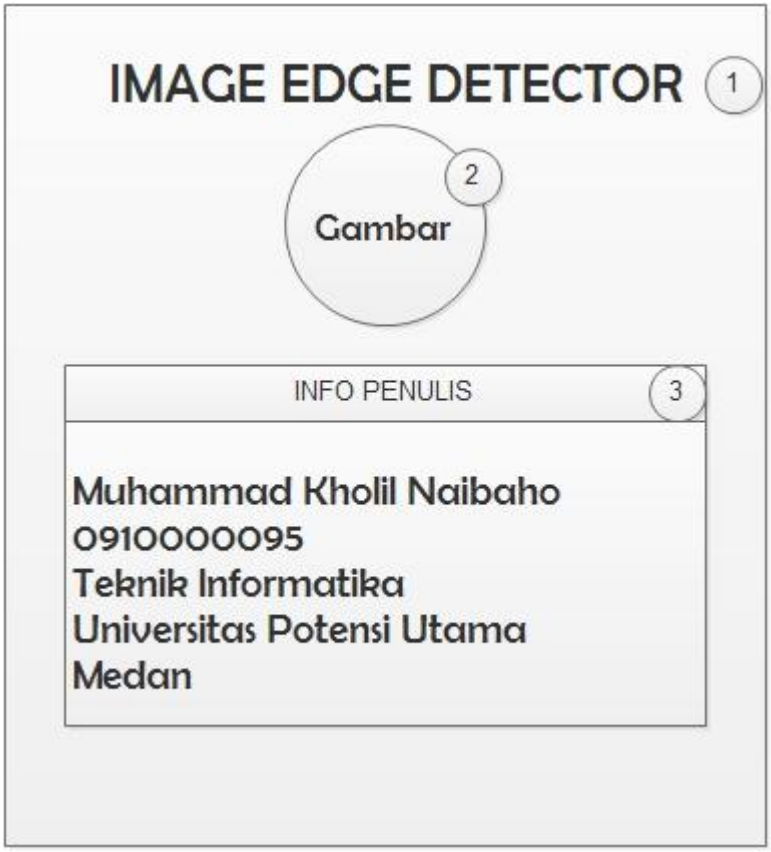
Gambar III.3 Desain Form Utama

Berikut adalah keterangannya

1. Judul dari aplikasi
2. Gambar dari logo aplikasi
3. Tombol untuk menampilkan form proses deteksi tepi
4. Tombol untuk menampilkan form tentang penulis
5. Tombol untuk keluar dari aplikasi

III.3.2 Form About

Form about merupakan form desain menampilkan informasi penulis, berikut adalah desainnya



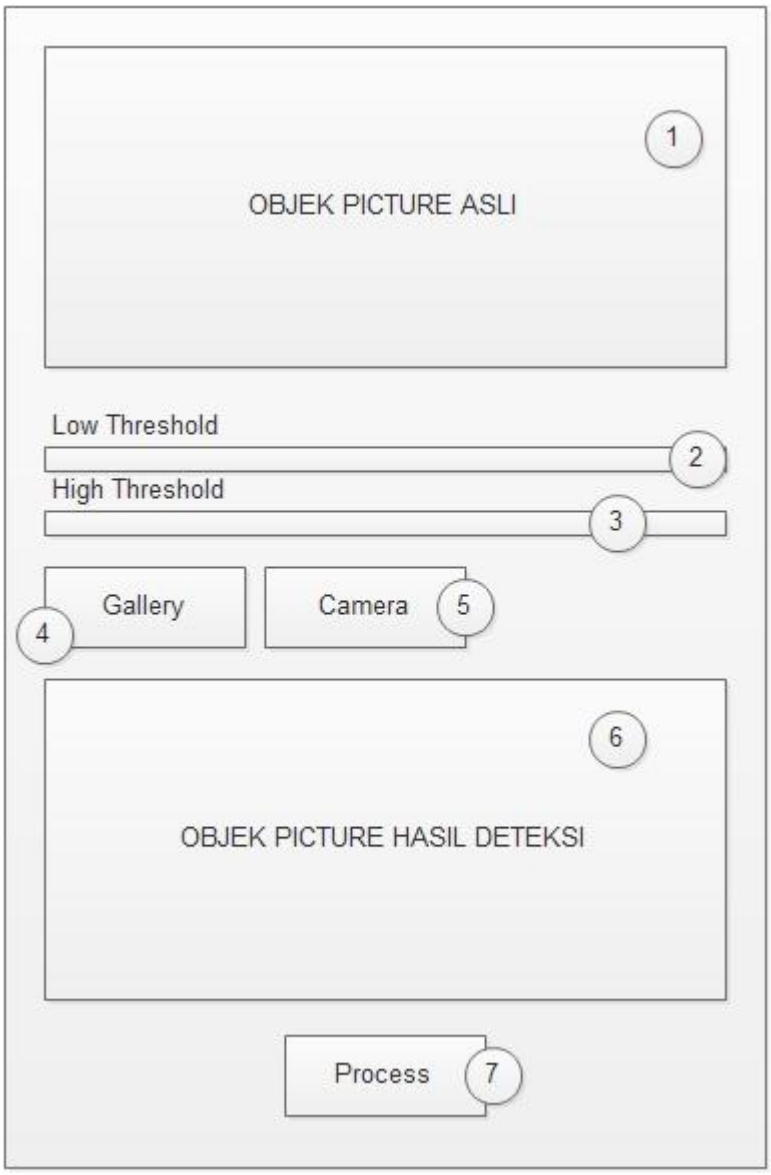
Gambar III.4 Desain Form About

Berikut adalah keterangannya

1. Judul dari aplikasi
2. Gambar dari logo aplikasi
3. Frame yang digunaka untuk menampilkan informasi penulis

III.3.3 Form Deteksi Tepi

Form deteksi tepi digunakan untuk melakukan proses pendeteksian tepi citra, berikut adalah desainnya



Gambar III.5 Desain Form Deteksi Tepi

Berikut adalah keterangannya

1. Objek picture yang digunakan untuk menampilkan gambar asli yang akan di deteksi tepi citranya
2. *Low threshold* merupakan pemberian nilai paling rendah untuk threshold gambar

3. *High threshold* merupakan pemberian nilai paling tinggi untuk threshold gambar
4. Tombol yang digunakan untuk mengambil file gambar dari smarphone android
5. Tombol yang digunakan untuk mengambil file gambar dari camera smarphone android
6. Objek picture yang digunakan untuk menampilkan gambar hasil pendeteksian tepi citra
7. Tombol untuk memproses deteksi tepi citra