

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2. 1 Tinjauan Studi**

Berbagai penelitian telah dilakukan untuk menunjukkan betapa pentingnya suatu edge detection dalam perkembangan pengolahan suatu citra, berikut adalah beberapa penelitian terkait dengan edge detection yang dilakukan oleh beberapa peneliti terdahulu. Penelitian yang dilakukan oleh Dr.S.Vijayarani, Mrs.M.Vinupriya [5] menyatakan bahwa sejak deteksi tepi adalah langkah awal pengenalan obyek, adalah penting untuk mengetahui perbedaan antara teknik deteksi tepi. Mewakili sebuah gambar pada bagian tepinya memiliki keuntungan mengurangi jumlah data yang diperlukan untuk disimpan sementara tetap mempertahankan sebagian besar informasi gambar. Transmisi piksel tepi dalam gambar atau multimedia akan menghasilkan banyak kompresi dan terdapat algoritma yang sangat handal untuk merekonstruksi seluruh gambar berdasarkan peta tepi. Dalam makalah penelitian ini kinerja relatif dari berbagai teknik deteksi tepi dilakukan dengan satu set gambar.

Ini telah diamati bahwa algoritma deteksi tepi Canny menghasilkan akurasi yang lebih tinggi di deteksi tepi dan waktu eksekusi dibandingkan dengan Sobel algoritma deteksi tepi. Sedangkan penelitian lain yang dilakukan oleh Arifin, Budiman [6] menyimpulkan bahwa:

1. Semakin tinggi resolusi citra, semakin tajam dan halus edge yang dihasilkan pada citra output. Pada citra resolusi rendah, edge yang dihasilkan kurang tajam dan kelihatan kasar.
2. Semakin tinggi kekontrasan citra, semakin baik pula kualitas dan jumlah edge yang dihasilkan pada citra output.
3. Metode Roberts' Cross menggunakan dua buah matriks konvolusi berukuran kecil yaitu  $2 \times 2$ . Hal ini menyebabkan jumlah proses perkalian menjadi lebih sedikit, tetapi rawan terhadap noise atau pixel rusak

Penelitian berikutnya yang dilakukan oleh Y.Ramadevi, T.Sridevi, B.Poornima, B.Kalyani[7] mengungkapkan tentang segmentasi sebuah image menggunakan algoritma deteksi tepi . Interaksi antara segmentasi image dan pengenalan object dalam sebuah framework pada algoritma Sobel, Prewitt, Roberts, Canny, LoG, Expectation-Maximization (EM), OSTU and Genetic dipelajari. Dalam penelitian ini digunakan tools MATLAB 7.9. Hasil yang diharapkan adalah Algoritma Ekspektasi Maksimum(EM) dan OTSU menghasilkan efek segmentasi yang stabil.

Penelitian yang dilakukan oleh Sri Enggal Indraani, dkk[4] dibahas suatu detection pada citra grayscale, dimana menggunakan dua operator yang digunakan adalah optimal. Kedua operator ialah operator Prewitt dan operator Sobel. Menggunakan tiga perhitungan yaitu perhitungan dengan Matlab, Matrix dan secara manual. Pada perhitungan dengan Matlab, source code telah ada pada aturan codingnya dan pengaturan gambar, posisi dan operator apa yang akan digunakan. Sedangkan pada perhitungan matrix telah menggunakan rumusan dari rumusnya dan perkalian dari citra asli. Bila dilihat pada hasil pada operator Sobel pada perhitungan manual terlihat lebih kontras.

Penelitian lainnya adalah yang dilakukan oleh C.NagaRaju, dkk [3] yaitu Morfologi Algoritma Deteksi Tepi Berdasarkan Multi-Structure Elemen Arah Berbeda Penelitian ini pada pengolahan gambar adalah kumpulan teknik yang dapat diterapkan pada diberikan gambar. Dalam tulisan ini, sebuah novel elemen multi-struktur algoritma deteksi tepi morfologi diusulkan untuk mendeteksi tepi gambar. Teknik yang dikembangkan sangat berguna untuk segmentasi Gambar dan klasifikasi. Pemilihan elemen struktur merupakan faktor kunci dalam morfologi pengolahan citra. Ukuran dan bentuk dari SE memutuskan Hasil akhir dari tepi terdeteksi.

Teori dasar multi elemen struktur morfologi adalah untuk membangun berbeda elemen struktur di jendela persegi yang sama. Dan ini struktur elemen terdiri hampir semua garis memanjang arah di jendela persegi. Yang diberikan eksperimental Hasil penelitian menunjukkan bahwa algoritma ini lebih efisien daripada biasanya digunakan tunggal dan simetris SE morfologi Operator

deteksi tepi dan deteksi tepi diferensial operator seperti metode DAS, operator Sobel dan Operator cerdas. Tepi terdeteksi lebih menonjol, Informasi integral dan terus-menerus, dan tepi lebih berlimpah. Selain itu, algoritma baru yang diusulkan dapat Filter kebisingan lebih berhasil daripada operator lain oleh pencahayaan tinggi tepi cerah. Meskipun metode ini menghasilkan hasil yang lebih baik, gagal untuk bayangan penghapusan

No	Nama Peneliti	Tahun	Judul	Metode	Hasil
1	Dr.S.Vi jayanan, Mrs.M. Vinupri ya	October 2013	Performance Analysis of Canny and Sobel Edge Detection Algorithms in Image Mining	Canny dan Sobel	algoritma deteksi tepi Canny menghasilkan akurasi yang lebih tinggi di deteksi tepi dan waktu eksekusi dibandingkan dengan Sobel
2	Arifin, Budiman	2012	Edge Detection Menggunakan Metode Roberts' Cross	Robert Cross	Metode Roberts' Cross rawan terhadap noise atau pixel rusak, karena perhitungannya berdasarkan matriks 2x2
3	Y.Ramadevi, dkk	6 Desember 2010	Segmentation And Object Recognition Using Edge Detection Techniques	Sobel, Prewitt, Robert, Canny, OSTU and Genetic	Algoritma Ekspektasi Maksimum(EM) dan OSTU menghasilkan efek segmentasi yang stabil.

4	Sri Enggal Indraan i, dkk	2014	Implementasi Edge Detection Pada Citra Grayscale dengan Metode Operator Prewitt dan Operator Sobel	Prewit, dan Sobel	suatu detection pada citra grayscale, dimana menggunakan dua operator yang digunakan adalah optimal
---	------------------------------------	------	--	-------------------------	--

*Tabel 2.1. Tabel Penelitian Terkait*

## 2.2 Tinjauan Pustaka

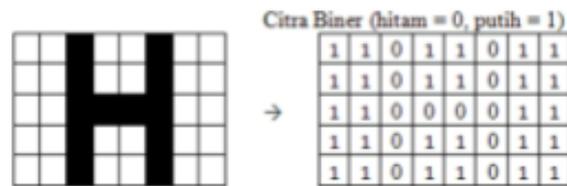
### 2.2.1 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra digital merupakan citra yang dapat diolah komputer. Yang disimpan dalam komputer hanyalah angka-angka yang menunjukkan besar intensitas pada masing-masing piksel. Karena berbentuk data numerik, maka citra digital dapat diolah dengan komputer. Piksel merupakan elemen citra yang memiliki nilai yang menunjukkan intensitas warna. Ada banyak cara untuk menyimpan citra digital di dalam memori.

Dari cara penyimpanan inilah, maka citra digital terbagi menjadi 3 jenis, yaitu:

a. Citra biner

Citra biner memiliki 2 jenis warna, yaitu hitam dan putih. Jadi dibutuhkan 1 bit di memori untuk menyimpan kedua jenis warna ini.

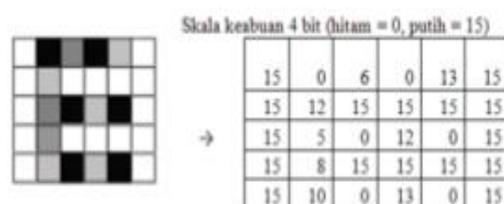


Gambar 2.1 Citra Biner

Gambar 2.1 menggambarkan huruf H, di mana bit 1 menggambarkan warna hitam dan bit 0 menggambarkan warna putih [6].

b. Citra *grayscale*

Citra *grayscale*, merupakan citra yang nilai piksel-nya merepresentasikan derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah adalah merepresentasikan warna hitam dan nilai intensitas paling tinggi merepresentasikan warna putih. Banyaknya warna pada jenis citra *grayscale* bergantung pada jumlah bit yang disediakan oleh memori untuk menampung. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya sampai 8 bit.

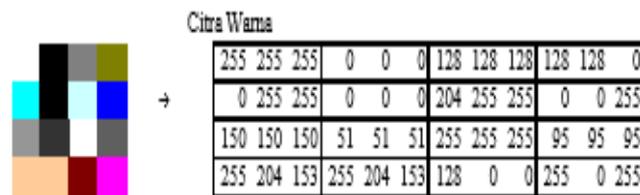


Gambar 2.2 Perbandingan gradasi warna citra *grayscale*

Gambar 2.2 menggambarkan perbandingan gradasi warna dari citra *grayscale*. Semakin besar bit-nya maka semakin halus warna *grayscale* yang ada ditampilkan [6].

c. Citra warna

Citra berwarna, merupakan citra yang nilai piksel-nya merepresentasikan warna tertentu. Banyaknya warna yang mungkin digunakan bergantung kepada kedalaman piksel citra yang bersangkutan. Citra berwarna direpresentasikan dalam kanal (*channel*) yang menyatakan komponen-komponen warna penyusunnya. Setiap piksel pada citra warna mewakili suatu warna yang merupakan kombinasi dari 3 warna dasar, yaitu RGB (*Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit atau 1 byte, yang berarti setiap warna memiliki 255 gradasi warna. Jadi setiap piksel mempunyai kombinasi warna sebanyak  $2^8 \cdot 2^8 \cdot 2^8 = 2^{24} = 16$  juta warna lebih.



Gambar 2.3 (a) Citra warna (b) Penyimpanan di memori

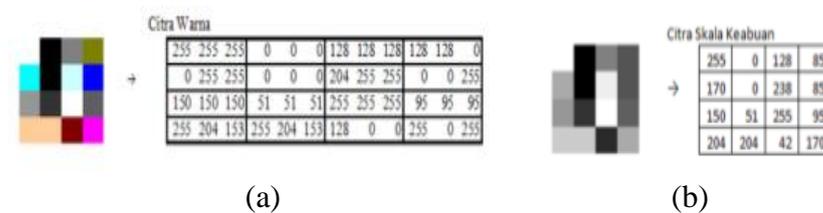
Gambar 2.3 menjelaskan bahwa gambar (a) menggambarkan citra warna yang penyimpanannya di dalam memori digambarkan oleh gambar (b) yang masing-masing pikselnya memiliki 3 warna dasar utama dan setiap warnanya memiliki kombinasi warna sebesar 8 bit [6].

### 2.2.2. Konversi Citra Warna (RGB) ke Citra Skala Keabuan (Grayscale)

Proses pertama dari metode pendeteksi citra secara umum termasuk metode Sobel adalah melakukan konversi citra, dari citra berwarna ke citra skala keabuan (grayscale). Rumus yang digunakan untuk konversi adalah sebagai berikut:

$$X = (R+G+B)/3$$

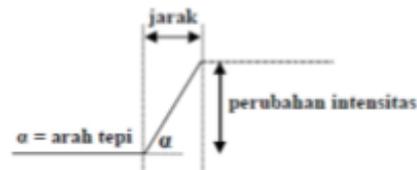
, dimana R adalah nilai warna merah (Red), G adalah nilai warna hijau (Green), dan B adalah nilai warna biru (Blue) dari pixel yang diproses. Gambar 2.4 (a) menunjukkan hasil proses konversi citra awal pada gambar 2.4 (b) yang merupakan sebuah citra keabuan [6]



Gambar 2.4 (a) Konversi Citra warna (b) Konversi ke citra keabuan

### 2.2.3. Edge Detection

Edge dapat didefinisikan sebagai batas antara dua region (dua piksel yang saling berdekatan) yang memiliki perbedaan intensitas yang tajam atau tinggi [8]. Edge dapat diorientasikan dengan suatu arah, dan arah ini berbeda-beda, tergantung pada perubahan intensitas. Gambar 2.5 memperlihatkan model edge dalam ruang satu dimensi [6].

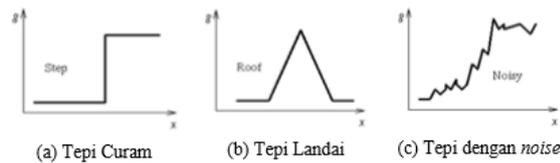


Gambar 2.5 Model Edge dalam ruang satu dimensi

Menurut Munir [9] ada tiga macam tepi (edge) yang terdapat di dalam citra digital seperti yang diperlihatkan pada gambar 2.6. Ketiga macam edge tersebut adalah:

1. Tepi curam, di mana tepi terbentuk karena perubahan intensitas yang tajam, berkisar 90°.
2. Tepi landau, di mana tepi lebar, sudut arah kecil. Terdiri dari sejumlah tepi-tepi lokal yang lokasinya berdekatan.
3. Tepi yang mengandung noise, di mana biasanya dilakukan

operator image enhancement terlebih dahulu, misalnya Operator Gaussian yang berfungsi untuk menghaluskan citra.

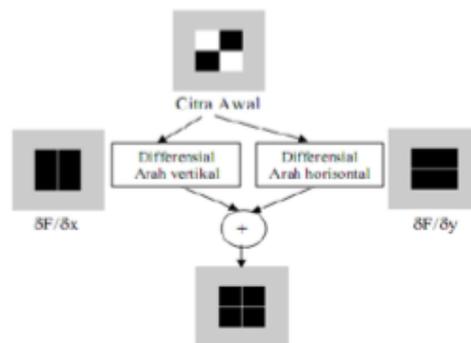


Gambar 2.6 Jenis-jenis Tepi

Edge detection pada suatu citra memiliki tujuan sebagai berikut [8]:

- Menandai bagian yang menjadi detil citra.
- Memperbaiki detil citra yang kabur karena error atau efek proses akuisisi.

Gambar 2.7 memperlihatkan bagaimana edge dari suatu citra dapat diperoleh dengan operasi edge detection [6]. Berdasarkan prinsip-prinsip filter pada citra, edge suatu gambar dapat diperoleh menggunakan High Pass Filter (HPF), dengan karakteristik:  $\sum \sum H(x,y) = 0$



Gambar 2.7 Proses Edge Detection

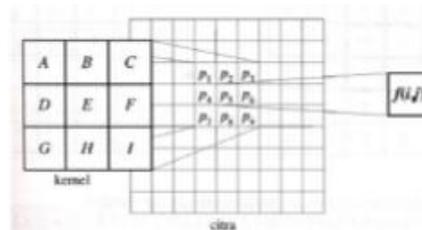
#### 2.2.4. Konvolusi

Edge detection merupakan salah satu proses pengolahan citra yang menggunakan filter, yang dilakukan melalui metode konvolusi. Konvolusi dinyatakan dalam matriks, dimana setiap elemen matriks dinamakan koefisien konvolusi. Operasi konvolusi bekerja dengan menggeser kernel piksel per piksel, yang hasilnya disimpan dalam matriks baru. Untuk lebih jelasnya, berikut contoh konvolusi yang terjadi

antara citra  $f(x,y)$  berukuran  $5 \times 5$  dengan sebuah kernel berukuran  $3 \times 3$  yang diperlihatkan oleh Gambar 2.9 Tanda ■ menunjukkan posisi  $(0,0)$  dari kernel [6].

$$f(x) = \begin{pmatrix} 4 & 4 & 3 & 5 & 4 \\ 6 & 6 & 5 & 5 & 2 \\ 5 & 6 & 6 & 6 & 2 \\ 6 & 7 & 5 & 5 & 3 \\ 3 & 5 & 2 & 4 & 4 \end{pmatrix} \quad g(x,y) = \begin{pmatrix} 0 & -1 & 0 \\ -1 & \blacksquare & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

Gambar 2.8 Proses Konvolusi



Gambar 2.9 Matriks & Kernel Sebelum Konvolusi

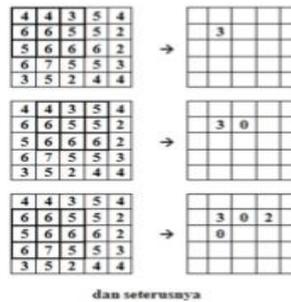
Proses perkalian matriks pada setiap titik pixel yang bersesuaian pada matriks citra dengan matriks kernel. Tahapan untuk mendapatkan hasil konvolusi yang terjadi antara citra dan kernel di atas dapat dilihat pada Gambar 2.10, sehingga diperoleh hasil akhir dari proses konvolusi tersebut, yang ditunjukkan oleh Gambar 2.11[6].

Seperti yang bisa diperhatikan pada Gambar 2.11, hasil akhir dari proses konvolusi mengalami pengurangan dua pixel secara horizontal maupun vertikal. Hal ini terjadi karena matriks kernel tidak bisa mengakomodasikan perhitungan pada pixel-pixel paling tepi dari citra awal. Walaupun demikian, pixel-pixel paling tepi dianggap tidak mengandung informasi yang cukup penting sehingga penghilangan pixel-pixel tersebut dari citra hasil konvolusi tidak akan memberikan pengaruh yang signifikan.

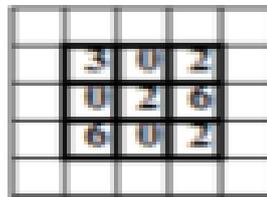
Dalam konvolusi terdapat dua kemungkinan yang jika ditemukan, diselesaikan dengan cara berikut, yaitu:

1. Untuk hasil konvolusi dengan nilai negatif, nilainya dijadikan nol (0).
2. Jika hasil konvolusi lebih besar dari derajat keabuan maksimum, maka

nilai diubah menjadi nilai maksimum derajat keabuan.



Gambar 2.10 Proses Pembentukan Konvolusi



Gambar 2.11 Matriks & Kernel Setelah Konvolusi

#### 2.2.5. Metode Sobel

Metode Sobel merupakan pengembangan metode robert dengan menggunakan filter HPF yang diberi satu angka nol penyangga. Metode ini mengambil prinsip dari fungsi laplacian dan gaussian yang dikenal sebagai fungsi untuk membangkitkan HPF. Kelebihan dari metode sobel ini adalah kemampuan untuk mengurangi noise sebelum melakukan perhitungan deteksi tepi. Peninjauan pengaturan pixel di sekitar pixelnya (x,y) adalah :

$$\begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & (x,y) & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

Operator sobel adalah magnitudo dari gradien yang dihitung dengan :

$$M = \sqrt{s_x^2 + s_y^2}$$

dalam hal ini turunan parsial dihitung dengan :

$$S_x = (a_2 + ca_3 + a_4) - (a_0 + ca_2 + a_6)$$

$$S_y = (a_0 + ca_1 + a_2) - (a_6 + ca_5 + a_4)$$

Dengan konstanta  $c = 2$ . Dalam bentuk  $s_x$  dan  $s_y$  dapat dinyatakan sebagai :

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

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

Arah tepi dihitung dengan persamaan :

$$\alpha(x,y) = \tan^{-1} \left( \frac{S_y}{S_x} \right)$$

Contoh penerapan Sobel sederhana :

$$\begin{bmatrix} \begin{bmatrix} 3 & 4 & 2 \\ 2 & \textcircled{1} & 6 \\ 3 & 5 & 7 \end{bmatrix} & 5 & 1 \\ 4 & 2 & 5 & 7 & 1 \\ 2 & 5 & 1 & 7 & 3 \end{bmatrix} \quad \begin{array}{ccccc} x & x & x & x & x \\ x & q_{2,2} & & & \end{array}$$

input

Jika diberikan input seperti matriks diatas maka perhitungan  $q_{2,2}$  adalah

$$\Delta x = -3 - 4 - 3 + (4) + (12) + (7) = +13$$

$$\Delta y = -3 - 8 - 2 + 3 + 10 + 7 = +7$$

$$q_{2,2} = ((-13)^2 + (+7)^2)^{1/2}$$

$$= (169 + 49)^{1/2}$$

= 15

### 2.2.6 Peak Signal to Noise Ratio (PSNR)

PSNR adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR digunakan untuk mengetahui perbandingan kualitas *image* sebelum dan sesudah pengolahan dengan edge detection. Selain itu PSNR juga dapat digunakan untuk mengukur ketahanan pada steganografi yang nantinya akan digunakan sebagai parameter.

Untuk menghitung PSNR dari suatu citra, terlebih dahulu harus menentukan MSE (*Mean Square Error*). MSE adalah nilai *error* kuadrat rata-rata antara citra asli dengan citra hasil edge detection. Semakin kecil nilai MSE, semakin bagus prosedur perbaikan citra yang digunakan. Artinya, kualitas citra setelah mengalami perbaikan noise hampir sama dengan kualitas citra asalnya. Sedangkan PSNR bernilai sebaliknya, jika nilai PSNR semakin besar maka kualitas citra hasil semakin mirip dengan kualitas citra hasilnya

Perhitungan MSE adalah sebagai berikut.[10]

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n [I(i, j) - K(i, j)]^2$$

Dimana:

MSE = Nilai *Mean Square Error* dari citra

m = panjang citra (dalam piksel)

n = lebar citra (dalam piksel)

(i,j) = koordinat masing-masing piksel

I = nilai intensitas citra asli

K = nilai intensitas citra hasil

Nilai PSNR didapat dengan rumus

$$PSNR = 20 \times \log_{10} \left( \frac{b}{rms} \right)$$

Dimana:

PSNR = nilai PSNR citra

B = nilai pixel terbesar (255)

Rms = root mean square

$$rms = \sqrt{\frac{1}{M \times N} \sum_{i=1}^N \sum_{j=1}^M (f_{ij}^o - f_{ij}^i)^2}$$

Contoh perhitungan PSNR :

7	1	1
2	3	4
5	0	6

(a)

6	1	1
3	3	3
5	1	6

(b)

Tabel 2.2 (a) Tabel piksel citra awal (b) Tabel piksel citra akhir

Tabel 2.2 adalah tabel piksel citra yang digunakan untuk perhitungan PSNR.

$$MSE = \frac{(7-6)^2 + (1-1)^2 + (1-1)^2 + (2-3)^2 + (3-3)^2 + (4-3)^2 + (5-5)^2 + (0-1)^2 + (6-6)^2}{3 \times 3}$$

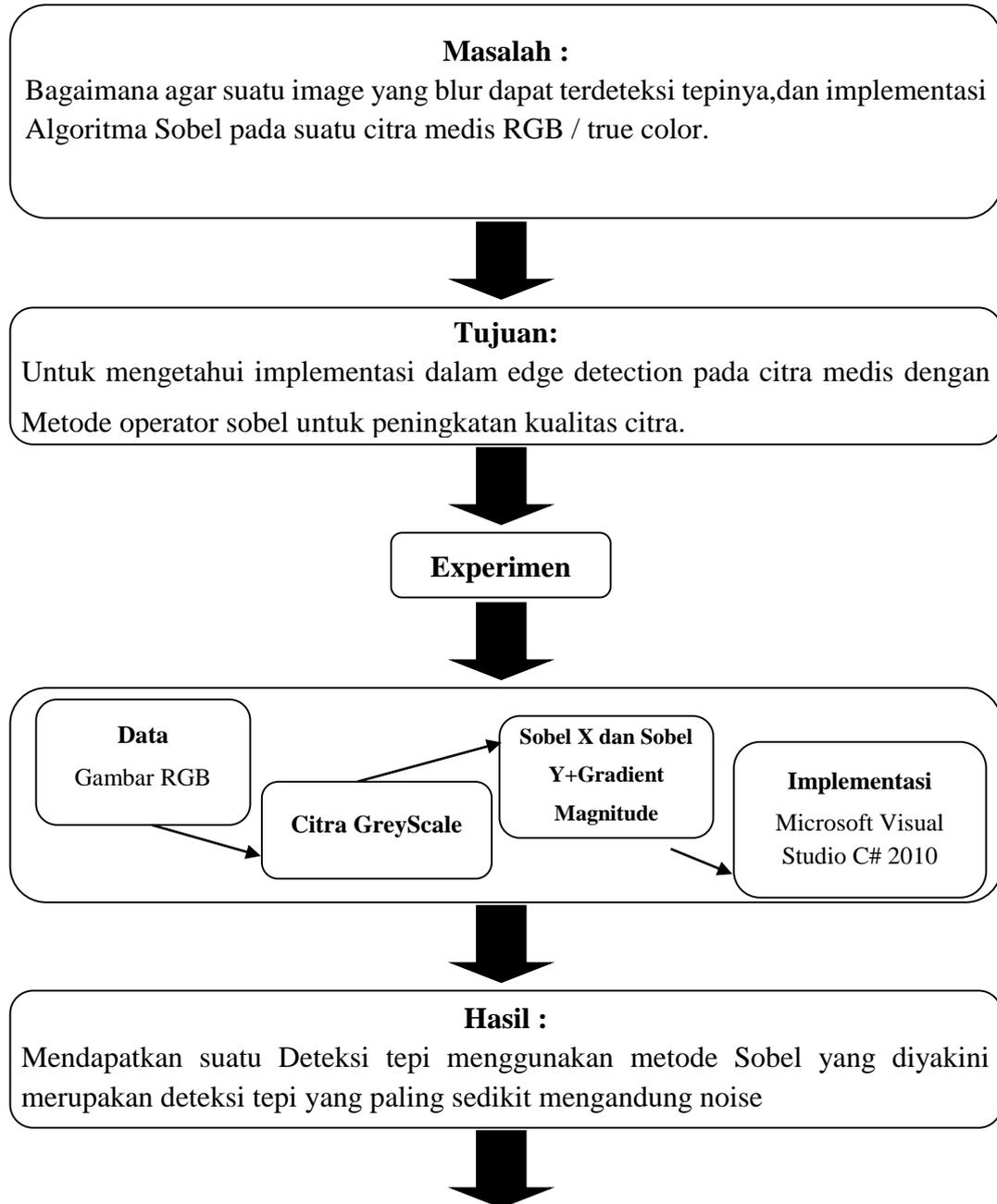
$$MSE = \frac{1+0+0+1+0+1+0+1+0}{9} = 0,444$$

$$Rms = \sqrt{0,444} = 0,666$$

$$PSNR = 20 * \log_{10} (255/0,666)$$

$$PSNR = 51,661$$

### 2.3 Kerangka Pemikiran



**Manfaat :**

- Mengetahui Deteksi Tepi satu Citra
- Dapat di implementasikan dalam berbagai kasus lain seperti MRI tulang , pengecekan plat nomor pada CCTV , dsb
- Dapat mengetahui keefektifan suatu algoritma deteksi tepi ketika kualitas gambar dirubah

