

# Implementasi Algoritma *Canny Edge Detection* untuk Identifikasi *Scratch* pada *Liquid Cristal Display Case* (LCD Case)

Andri Safari  
Universitas Buana Perjuangan  
Karawang, Indonesia  
if15.andrisafari@mhs.ubpkarawang.ac.id

Hanny Hikmayanti  
Universitas Buana Perjuangan  
Karawang, Indonesia  
hanny.hikmayanti@ubpkarawang.ac.id

Deden Wahiddin  
Universitas Buana Perjuangan  
Karawang, Indonesia  
deden.wahiddin@ubpkarawang.ac.id

## Abstract—

LCD (*Liquid Cristal Display*) adalah bagian dari salah satu rangkaian elektronik yang berfungsi untuk menampilkan suatu data, berupa huruf, grafik atau karakter. Material LCD adalah lapisan dari campuran organik antara lapisan kaca bening dengan elektroda transparan *indium oksida*, sedangkan LCD Case adalah tempat di mana LCD diletakkan dalam rangkaian elektronika. Sedangkan *Computer Vision* merupakan cabang ilmu pengolahan citra *digital* yang memungkinkan komputer dapat melihat seperti manusia, sehingga dapat mengambil keputusan, melakukan aksi, dan mengenali suatu objek. Bentuk implementasi dari *Computer Vision* adalah *Edge Detection*. Pada penelitian ini dibangun sebuah aplikasi pengolahan *image* yang dibutuhkan untuk mengidentifikasi LCD Case *Scratch*. Aplikasi ini dapat digunakan untuk mengidentifikasi LCD Case *scratch* sehingga LCD Case yang diterima sesuai dengan kualitas yang ditetapkan. Aplikasi identifikasi LCD Case *Scratch* ini dirancang dengan metode *canny edge detection* berbasis *computer vision* dengan menggunakan bahasa pemrograman *Python*. Aplikasi ini mampu mengidentifikasi *scratch* pada LCD Case dan menampilkan hasil identifikasi *scratch* secara *realtime* dengan cara menampilkan hasil identifikasi pada layar komputer, sehingga aplikasi ini diharapkan dapat mempermudah penerimaan dan identifikasi LCD Case *scratch*.

**Kata kunci — *CannyDetection, Identifikasi Lcd, Method Scratch***

## I. PENDAHULUAN

LCD Case adalah salah satu dari sekian banyak *parts* (komponen) *videotape* mobil. LCD Case dengan kualitas yang baik akan menghasilkan produk *videotape* mobil dengan kualitas yang tinggi. Namun, kualitas *videotape* mobil yang baik adalah tujuan utama dari proses produksi. Selanjutnya, *defect* adalah *parts* dengan kualitas di luar spesifikasi standar kualitas yang telah ditetapkan[1]. *Scratch, flashes, out to spec, dimensions out, bending, oily, gass mark, dirty, over buffing* merupakan kategori *parts defect*. Dari kasus *defect* yang telah dipaparkan, maka dilakukan penelitian pada kategori *defect scratch*. *Scratch* merupakan *defect* berupa celah (garis) pada permukaan *Flatbar* (tidak rata) yang dapat mengurangi kualitas *videotape* mobil.

Pembangunan aplikasi ini merupakan salah satu solusi yang dapat diterapkan untuk mengidentifikasi LCD Case *scratch*. Masalah terkait dengan *scratch* telah diteliti oleh Nursabillah muhammad alie[2] dengan judul *Quality checking and inspection based on machine vision technique to determine tolerance value using single ceramic cup* dengan meneliti *scratch* pada gelas keramik dengan cara mengolah *image* objek dari sebuah foto objek, untuk menghasilkan bentuk tepi pada objek kemudian menganalisa *scratch* pada keramik dengan ketajaman dari tepi yang dihasilkan dan Ayu ambarwati[3] dengan judul Segmentasi Citra lahan bekas tambang terbuka menggunakan metode deteksi tepi *canny* dengan fokus penelitian mendeteksi tepi citra foto udara lahan bekas tambang, sedangkan untuk penelitian yang dilakukan adalah mendeteksi dan mengidentifikasi objek secara *realtime* dengan mengolah video objek menggunakan metode *canny* dan beberapa *Library OpenCV* untuk mengidentifikasi *scratch* pada LCD Case, kemudian menampilkan hasil identifikasi secara *realtime* pada monitor komputer.

Berdasarkan masalah yang telah dipaparkan maka penelitian ini mengimplementasikan beberapa kesimpulan yaitu memastikan mengetahui spesifikasi *defect* LCD Case *scratch*, membuat aplikasi untuk implementasi algoritma *Canny edge detection* untuk identifikasi *Scratch* pada LCD Case, memastikan tingkat akurasi aplikasi implementasi algoritma *Canny edge detection* untuk identifikasi *Scratch* pada LCD Case.

## II. STUDI LITERATUR

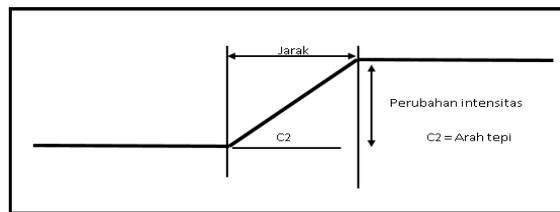
### A. Citra Digital

Citra adalah gambar atau objek dua dimensi. Citra yang dihasilkan dari gambar analog dua dimensi dan kontinu menjadi gambar *diskrit*, melalui proses sampling gambar *analog* dibagi menjadi M baris

dan N kolom sehingga menjadi gambar *diskrit*[4]. Salah satu bagian yang mempunyai peranan sangat penting pada bagian *multimedia* adalah *citra digital*, *citra digital* mampu memberikan informasi *visual* dari suatu objek. *Citra digital* mempunyai ciri khas yang tidak dimiliki oleh grafik, huruf dan data teks, sebuah citra bisa saja memiliki berbagai informasi. Dengan melihat sebuah citra kita mendapatkan berbagai informasi dari apa yang kita lihat. Suatu citra dapat dikatakan bahwa citra tersebut adalah *citra digital* jika mempunyai fungsi  $f(x,y)$  dan memiliki titik *koordinat*  $(x, y)$  dan memiliki *intensitas* atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai  $x, y$  dan nilai amplitude  $f$  secara keseluruhan berhingga (*finite*) dan bernilai diskrit[5].

B. *Deteksi Tepi*

Tepi adalah nilai sebuah intensitas dari tingkat keabuan yang memiliki perubahan besar dalam waktu yang sangat cepat. Deteksi tepi digunakan untuk memperoleh tepi-tepi dari sebuah objek citra[3]. Sebuah tepi mengandung berbagai informasi penting berupa bentuk maupun ukuran sebuah objek tersebut. Citra memiliki titik  $x$  dan  $y$  disebut dengan sebuah tepi dan ditandai tingginya perubahan nilai *pixel* sangat kontras terhadap *pixel* lainnya, suatu citra yang ingin garis batasnya nampak terlihat jelas atau meningkatkan garis batas objek adalah salah satu operasi dari deteksi tepi. *Operator gradien* pertama yang digunakan untuk mendeteksi tepi di dalam citra, yaitu *operator gradien selisih terpusat*, *operator Sobel*, *operator Prewitt*, *operator Roberts*, *operator Canny*. Gambar 1 merupakan alur perubahan sebuah nilai tepi pada objek.

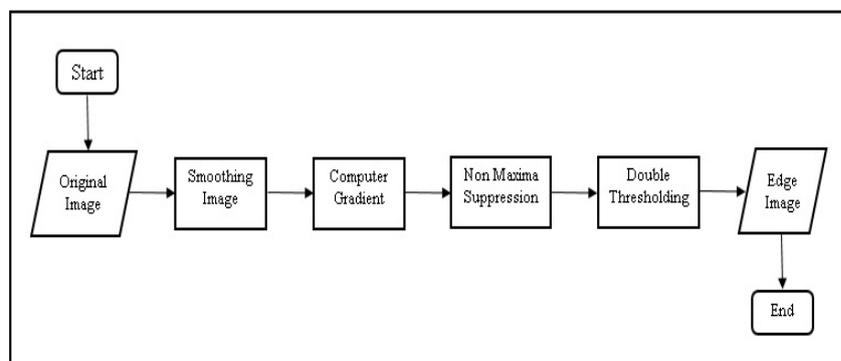


Gambar.1 Model Tepi Satu Dimensi

C. *Canny Edge Detection*

Metode *Canny Edge Detection* dikembangkan oleh *John F. Canny* pada tahun 1986, dan telah menjadi salah satu metode deteksi tepi standar yang digunakan dalam sebuah penelitian[6] [7]. Deteksi tepi *Canny* terdiri dari empat proses yaitu dan Gambar 2 adalah diagram alir untuk proses *Canny Edge Detection*:

1. *smoothing image* : Citra yang masuk dilakukan *smoothing image* dengan tujuan untuk memperbaiki kualitas citra dengan cara menghilangkan derau.
2. *compute gradient* : Merupakan proses untuk menghitung potensi gradien suatu tepi citra.
3. *Non-maxima Suppression* : Proses *non-maxima suppression* bertujuan untuk melokalisasi tepi obyek secara presisi.
4. *Double Thresholding* : Proses *double thresholding* bertujuan untuk menentukan kategori *pixel edge*.



Gambar.2 Flow chart Canny

D. *Open Computer Vision ( OpenCV)*

*Intel Open Source Computer Vision Library* adalah kepanjangan dari *OpenCV. Library OpenCV* memiliki lebih dari 300 fungsi-fungsi C (program bahasa C dan turunannya). *OpenCv* merupakan *library* gratis dan dapat dipakai untuk kebutuhan pribadi ataupun kepentingan bisnis, tanpa harus membayar atau membeli ijin pada pihak *intel*. Sistem operasi yang sudah *teinstall linux* atau *windows*

dapat menjalankan *OpenCV*. *Software OpenCV* menyediakan berbagai fungsi-fungsi *image processing*, seperti halnya dengan fungsi-fungsi analisis citra atau gambar dan pola [8].

Identifikasi Objek, deteksi benda, interaksi Manusia dan Komputer, segmentasi objek, dan Pengenalan wajah, pengenalan Gerak Isyarat atau *gesture*, pengenalan Gerakan objek, gerakan Dari Struktur objek, gerakan Ego, dan Pemahaman Gerakan objek, dan Robot-Robot Yang bergerak atau *Mobile Robotics*, menghitung jarak antar suatu objek merupakan hasil penerapan *Software Open Computer Vision*.

E. *Computer Vision*

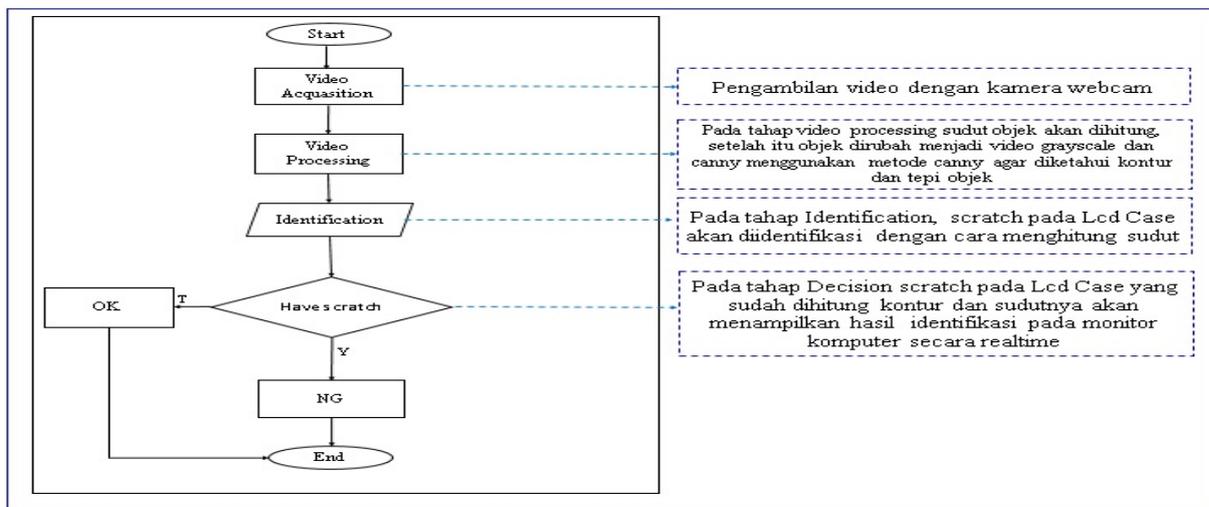
*Computer vision* mempunyai tujuan ingin meniru cara kerja *visual* manusia (*Human Vision*). Pekerjaan yang biasa dilakukan oleh manusia seperti melihat mengontrol dan lain-lain dapat digantikan dengan *computer vision*, pada era *digital* saat ini *computer vision* terus dikembangkan. *Computer Vision* adalah suatu sistem yang mempunyai kemampuan untuk menganalisis sebuah obyek secara *visual*, setelah data suatu obyek yang bersangkutan dimasukkan dalam bentuk citra[9]. [10]Dalam melakukan atau mengambil suatu keputusan *computer vision* memperoleh data dari hasil interpretasi. Tahapan-tahapan dalam *computer vision* terbagi 3 (tiga) tahapan yaitu :

1. Tahap akuisisi citra *digital*, pada tahap ini dapat juga disebut sebagai tahap awal atau proses *image preprocessing*.
2. Tahap teknik komputasi di mana *pixel* dari sebuah data citra diolah dan dilakukan modifikasi.
3. Tahap menganalisis yaitu citra yang telah melewati proses analisis hasilnya dapat digunakan untuk proses lain dengan tujuan tertentu, misalnya mengontrol atau mengarahkan robot, mengawasi alat, mengawasi lingkungan pabrik dan sebagainya.

III. METODE PENELITIAN

Pada penelitian ini dilakukan penelitian dengan fokus penelitian mengidentifikasi LCD Case, yaitu dengan mengidentifikasi *scratch* pada LCD Case. Proses identifikasi *scracth* pada LCD Case menggunakan pengolahan citra *digital* dengan metode *Canny Edge Detection*.

Proses implementasi algoritma *Canny edge detection* untuk identifikasi *Scratch* pada LCD Case dengan cara mengolah video secara *realtime* yang diambil menggunakan kamera *webcam*, video LCD Case kemudian diproses dengan menggunakan perangkat lunak sistem operasi *Linux mint* yang sudah *terinstall software Open Computer Vision (OpenCV)* dan beberapa *Library* yaitu *cv2, numpy as np* dan *math* dengan menggunakan bahasa pemrograman *Python*. Hasil dari proses ini yaitu *scratch* yang ada pada LCD Case dapat teridentifikasi secara *realtime*.



Gambar.3 Diagram Alir

A. *Pengambilan Data*

Adapun metode yang digunakan untuk mendapatkan data yang benar-benar dapat dipercaya, akurat dan relevan yaitu :

- Akusisi citra

Kamera *webcam* digunakan untuk proses pengambilan data citra secara langsung, dengan mengarahkan kamera *webcam* ke LCD Case.

- *Observasi*  
Teknik peninjauan langsung ini sangat diperlukan guna memastikan permasalahan yang terjadi dan metode apa yang akan diambil guna menyelesaikan permasalahan tersebut.

B. *Rancangan Penelitian*

Perancangan sistem merupakan desain atau alur dari sistem yang akan dibuat, metode yang akan diterapkan adalah metode *Canny Edge Detection* yang secara umum tahapan yang dilakukan pada metode ini adalah : *VideoAcquasition, Canny edge detection, Identification* dan *Result*.

Untuk proses pada masing-masing tahapannya dapat dilihat pada penjelasan dibawah ini :

- *Video Acquasition* yaitu Pada proses ini pengambilan video atau image LCD Case dengan menggunakan kamera *webcam*.
- *Video processing method* yaitu Pada tahap *video processing* sudut atau *angle* objek pada image atau video akan dihitung menggunakan potongan kode yang ditunjukkan pada Gambar 3.

Tabel 1 Potongan kode program 1

```

20 #calculate angle
21 def angle(pt1,pt2,pt0):
22     dx1 = pt1[0][0] - pt0[0][0]
23     dy1 = pt1[0][1] - pt0[0][1]
24     dx2 = pt2[0][0] - pt0[0][0]
25     dy2 = pt2[0][1] - pt0[0][1]
26     return float((dx1*dx2 + dy1*dy2)/math.sqrt(float((dx1*dx1 + dy1*dy1)*(dx2*dx2 + dy2*dy2)
27     + 1e-10)
28 while(cap.isOpened()):
29     #Capture frame-by-frame
30     ret, frame = cap.read()
31     if ret==True:
32         #grayscale
33         gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
34         #Canny
35         canny = cv2.Canny(frame,80,240,3)
36
37         #contours
38         canny2, contours, hierarchy =
39         cv2.findContours(canny,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
40         for i in range(0,len(contours)):
41             #approximate the contour with accuracy proportional to
42             #the contour perimeter
43             approx = cv2.approxPolyDP(contours[i],cv2.arcLength(contours[i],True)*0.02,True)
44
45             if(abs(cv2.contourArea(contours[i]))<100 or not(cv2.isContourConvex(approx))):
46                 continue
47

```

- *Identification* yaitu Pada tahapan *Identification* video LCD Case yang sudah diketahui sudut dan tepi objeknya akan mengidentifikasi *scratch* yang ada pada LCD Case kemudian menghitung sudut dan kontur *scratch* pada LCD Case menggunakan potongan kode yang ditunjukkan pada Tabel 1.

Tabel.2 Potongan kode program 2

```

48 #Scratch
49 if(len(approx) == 3):
50     x,y,w,h = cv2.boundingRect(contours[i])
51     cv2.putText(frame,'SCRATCH',(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,(255,255,255),2,cv2.LINE_AA)
52     elif(len(approx)>=4 and len(approx)<=6):
53         #nb vertices of a polygonal curve
54         vtc = len(approx)
55         #get cos of all corners
56         cos = []
57         for j in range(2,vtc+1):

```

```

58     cos.append(angle(approx[j% vtc],approx[j-2],approx[j-1]))
59     #sort ascending cos
60     cos.sort()
61     #get lowest and highest
62     mincos = cos[0]
63     maxcos = cos[-1]
64
65     #Use the degrees obtained above and the number of vertices
66     #to determine the shape of the contour
67     x,y,w,h = cv2.boundingRect(contours[i])
68     if(vtc==4):
69
70         cv2.putText(frame,'SCRATCH',(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,(255,255,255),2,cv2.LINE_AA)
71         img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)
72         elif(vtc==5):
73
74         cv2.putText(frame,'SCRATCH',(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,(255,255,255),2,cv2.LINE_AA)
75
76         else:
77             #Scratch
78             area = cv2.contourArea(contours[i])
79             x,y,w,h = cv2.boundingRect(contours[i])
80             radius = w/2
81             if(abs(1 - (float(w)/h))<=2 and abs(1-(area/(math.pi*radius*radius)))<=0.2):
82
83                 cv2.putText(frame,'SCRATCH',(x,y),cv2.FONT_HERSHEY_SIMPLEX,scale,(255,255,255),2,cv2.LINE_AA)
84                 img = cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)
85
86             #Display the resulting frame
87             out.write(frame)
88             cv2.imshow('Identification Scratch',frame)
89             cv2.imshow('canny edge detection method',canny)
90             if cv2.waitKey(1) == 1048689: #if q is pressed
91                 break
92
93         #When everything done, release the capture
94         cap.release()
95         cv2.destroyAllWindows()

```

- Berdasarkan hasil perhitungan potongan kode program pada tabel 2 akan menentukan apakah *scratch* melebihi ambang batas yang dibolehkan atau masih dibawah ambang batas.
- *Decision* yaitu Hasil identifikasi kontur dan tepi menggunakan kode program diatas, akan menampilkan hasil identifikasi *scratch* pada LCD Case dengan menampilkan tulisan *scratch* diatas objek secara *realtime* pada monitor komputer.

### C. Implementasi

Aplikasi identifikasi LCD Case *scratch* dengan metode *Canny edge detection* ini merupakan aplikasi yang hanya bisa digunakan untuk mengidentifikasi *scratch* yang ada pada LCD Case dengan menggunakan perangkat lunak sistem operasi *Linux mint* yang sudah *terinstall software Open Computer Vision (OpenCV)* dan beberapa *Library* yaitu *cv2*, *numpy as np* dan *math* dengan menggunakan bahasa pemrograman *Python*, proses pengambilan video objek menggunakan kamera webcam dengan resolusi 5 megapixel.

### D. Pengujian

Pada tahap proses pengujian, akan menguji berapa besar tingkat akurasi pada proses identifikasi LCD Case *scratch* dengan metode *canny edge detection*, di mana pada proses pengujian ini akan diketahui apakah aplikasi ini mampu mengidentifikasi LCD Case *scratch* secara *realtime*. Perangkat Pengujian

Perangkat pengujian menggunakan laptop dengan spesifikasi berikut ini :

1. Intel(R) celeron(R) CPU 1000M.@1.80GHZ.
2. 4.00 GB RAM
3. 32-bit Operating sysyem
4. Sistem Operasi Linux mint
5. Bahasa pemrograman python 2.7
6. Webcam dengan resolusi kamera 5 megapixel
7. LCD Case
8. Black Sheet

#### IV. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini yaitu membangun aplikasi implementasi algoritma *Canny edge detection* untuk identifikasi *Scratch* pada LCD Case, dengan menggunakan perangkat lunak sistem operasi *Linux mint* yang sudah *terinstall software Open Computer Vision (OpenCV)* dan beberapa *library* yaitu *cv2, numpy as np* dan *math* dengan menggunakan bahasa pemrograman *python*. Aplikasi identifikasi LCD Case *scratch* dengan metode *canny edge detection* ini mampu mengidentifikasi LCD Case *scratch* secara *realtime*.

##### A. Identifikasi Masalah

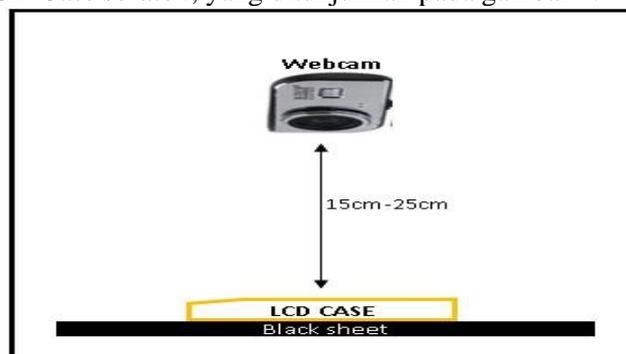
Dari hasil analisa sistem pada objek penelitian dapat dianalisa masih adanya aktifitas manual dari operator ketika melakukan proses identifikasi LCD Case *scratch*. Operator akan memastikan diameter atau ketebalan *scratch* pada LCD Case tidak melebihi ambang batas maksimal yang dibolehkan dengan menggunakan alat pembanding yaitu *template mold check gauge*. Dengan cara mengukur dan membandingkan secara visual antara *scratch* pada LCD Case dengan ukuran ambang batas yang tertera pada *mold check*.

##### B. Program

Berdasarkan analisa sistem yang berjalan penulis mengimplementasikan dengan rancangan aplikasi. Pada pemrograman aplikasi implementasi algoritma *Canny edge detection* untuk identifikasi *Scratch* pada LCD Case menggunakan sistem operasi *Linux mint* yang sudah *terinstal software Open Computer Vision (OpenCV)* dan beberapa *Library* yaitu *cv2, numpy as np* dan *math*, aplikasi ini menggunakan bahasa pemrograman *python*.

##### C. Pengujian

Proses pengujian dilakukan langsung dengan mengidentifikasi LCD Case *scratch* dengan kondisi jarak antara LCD Case dan *webcam* adalah 15cm sampai dengan 25cm dengan diameter atau ketebalan *scratch* 2mm dan panjang *scratch* 10mm guna mengetahui tingkat akurasi aplikasi dalam mendeteksi dan mengidentifikasi LCD Case *scratch*, yang ditunjukkan pada gambar 4.

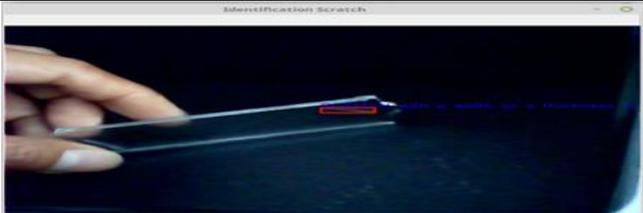


Gambar.4 Kondisi Pengujian

Hasil dari pengujian identifikasi LCD Case *scratch* termuat dalam sebuah matriks pada Tabel 3

**Tabel 3 Pengujian**

No	Skenario	Identifikasi LCD Case <i>Scratch</i> dengan metode <i>Canny edge detection</i>		Result
		Manual	Sistem	
1	implementasi algoritma <i>Canny edge detection</i> untuk identifikasi <i>Scratch</i> pada LCD Case	Scratch NG	Scratch NG	Sesuai

No	Skenario	Identifikasi LCD Case Scratch dengan metode Canny edge detection		Result
		Manual	Sistem	
				
2	implementasi algoritma <i>Canny edge detection</i> untuk identifikasi <i>Scratch</i> pada LCD Case			
		Scratch NG	Scratch NG	Sesuai
3	implementasi algoritma <i>Canny edge detection</i> untuk identifikasi <i>Scratch</i> pada LCD Case			
		Scratch NG	Scratch NG	Sesuai
4	implementasi algoritma <i>Canny edge detection</i> untuk identifikasi <i>Scratch</i> pada LCD Case			
		Scratch NG	Scratch NG	Sesuai
5	implementasi algoritma <i>Canny edge detection</i> untuk identifikasi <i>Scratch</i> pada LCD Case			
		Scratch NG	Scratch NG	Sesuai

Tahap pengujian selanjutnya dilakukan 30 kali pengujian, dan Tabel 4 adalah matriks dari hasil pengujian akhir.

**Tabel 4 Pengujian Akhir**

No	Iterasi pengujian	implementasi algoritma <i>Canny edge detection</i> untuk identifikasi <i>Scratch</i> pada LCD Case		Result
		Manual	Sistem	
1	Pengujian ke 1	Scratch NG	Scratch NG	Sesuai
2	pengujian ke 2	Scratch NG	Scratch NG	Sesuai
3	Pengujian ke 3	Scratch NG	Scratch NG	Sesuai
4	Pengujian ke 4	Scratch NG	Scratch NG	Sesuai
5	Pengujian ke 5	Scratch NG	Scratch NG	Sesuai
6	Pengujian ke 6	Scratch NG	Scratch NG	Sesuai

No	Iterasi pengujian	implementasi algoritma <i>Canny edge detection</i> untuk identifikasi <i>Scratch</i> pada LCD Case		Result
		Manual	Sistem	
7	Pengujian ke 7	Scratch NG	Scratch NG	Sesuai
8	Pengujian ke 8	Scratch NG	Not identified	X
9	Pengujian ke 9	Scratch NG	Scratch NG	Sesuai
10	Pengujian ke 10	Scratch NG	Not identified	X
11	Pengujian ke 11	Scratch NG	Scratch NG	Sesuai
12	Pengujian ke 12	Scratch NG	Not identified	X
13	Pengujian ke 13	Scratch NG	Not identified	X
14	Pengujian ke 14	Scratch NG	Scratch NG	Sesuai
15	Pengujian ke 15	Scratch NG	Not identified	X
16	Pengujian ke 16	Scratch NG	Scratch NG	Sesuai
17	Pengujian ke 17	Scratch NG	Scratch NG	Sesuai
18	Pengujian ke 18	Scratch NG	Not identified	X
19	Pengujian ke 19	Scratch NG	Not identified	X
20	Pengujian ke 20	Scratch NG	Scratch NG	Sesuai
21	Pengujian ke 21	Scratch NG	Scratch NG	Sesuai
22	Pengujian ke 22	Scratch NG	Scratch NG	Sesuai
23	Pengujian ke 23	Scratch NG	Not identified	X
24	Pengujian ke 24	Scratch NG	Scratch NG	Sesuai
25	Pengujian ke 25	Scratch NG	Scratch NG	Sesuai
26	Pengujian ke 26	Scratch NG	Scratch NG	Sesuai
27	Pengujian ke 27	Scratch NG	Scratch NG	Sesuai
28	Pengujian ke 28	Scratch NG	Scratch NG	Sesuai
29	Pengujian ke 29	Scratch NG	Scratch NG	Sesuai
30	Pengujian ke 30	Scratch NG	Scratch NG	Sesuai

Dari sebanyak 30 kali pengujian yang dilakukan maka dapat diketahui tingkat akurasi yaitu : Hasil pengujian Aplikasi identifikasi LCD Case *scratch* dengan metode *canny edge detection* memiliki tingkat akurasi sebesar 73% dengan data uji sebanyak 30 data.

## V. KESIMPULAN DAN SARAN

Penelitian yang telah dilakukan mendapatkan beberapa hasil kesimpulan yang diperoleh yaitu spesifikasi *defect* LCD Case *Scratch* berdasarkan hasil wawancara dengan *operator* memiliki ambang batas maksimal sebesar 2mm untuk diameter atau ketebalan *scratch* dan 10mm untuk panjang *scratch*, di mana jika diameter atau ketebalan *scratch* melebihi 2mm dan panjang *scratch* melebihi 10mm maka *NG (NO GOOD)*. Kemudian, aplikasi implementasi *algoritma Canny edge detection* untuk identifikasi *Scratch* pada LCD Case dibangun dengan menggunakan sistem operasi *Linux mint* yang sudah terinstall *software Open Computer Vision (OpenCV)* dan beberapa *Library* yaitu *cv2*, *numpy as np* dan *math* dengan menggunakan bahasa pemrograman *Python*, aplikasi implementasi *algoritma Canny edge detection* untuk identifikasi *Scratch* pada LCD Case memiliki tingkat keberhasilan atau akurasi sebesar 73% dengan data uji sebanyak 30 data.

Penelitian selanjutnya disarankan dapat mengidentifikasi *item part* atau tipe *NG* dan *defect* lainnya, dan menggunakan *webcam* dengan resolusi kamera diatas dari 5 *megapixel* untuk hasil yang lebih baik. Kemudian, untuk *scratch* yang tidak teridentifikasi karena sudut *scratch* yang sulit dijangkau, maka disarankan untuk menambah pencahayaan dan menggunakan metode lain.

## PENGAKUAN

Naskah ilmiah ini adalah sebagian dari penelitian Tugas Akhir milik Andri Safari dengan judul Identifikasi LCD Case dengan metode *Canny edge detection* yang dibimbing oleh Hanny Hikmayanti dan Deden Wahiddin .

## DAFTAR PUSTAKA

- [1] M. S. Arif, C. F. Putri, and N. Tjahjono, "Peningkatan Grade Kain Sarung dengan Mengurangi Cacat Menggunakan Metode Kaizen dan Siklus PDCA pada PT . X," vol. 26, no. 2, pp. 222–231, 2018.
- [2] N. M. Alie *et al.*, "Quality checking and inspection based on machine vision technique to determine tolerancevalue using single ceramic cup," *ARNP J. Eng. Appl. Sci.*, vol. 12, no. 8, pp. 2737–2742, 2017.

- [3] A. Ambarwati and R. Passarela, "Menggunakan Metode Deteksi Tepi Canny," vol. 7, no. 1, pp. 29–37, 2017.
- [4] Kumaseh et.al, "Segmentasi citra digital ikan menggunakan metode Tresholding," *J. Ilm. Sains*, vol. 13 No., no. 1, p. 6, 2013.
- [5] T. Citra, "Sx = -2," vol. 3, no. 6, pp. 25–29, 2016.
- [6] N. Afriliana and R. Valeria, "Pendeteksian Ruang Kosong Parkir di dalam Ruangan," vol. X, no. 1, pp. 34–40, 2018.
- [7] G. Ramadhan, E. C. Djamal, and T. Darmanto, "Klasifikasi Identitas Wajah Untuk Otorisasi Menggunakan Deteksi Tepi dan LVQ," *Semin. Nas. Apl. Teknol. Inf. 2016 Yogyakarta*, pp. 37–41, 2016.
- [8] K. D. Irianto, "Pendeteksi gerak berbasis kamera menggunakan opencv pada ruangan," vol. 2, no. 1, pp. 52–59, 2010.
- [9] W. S. Pambudi and A. N. Tompunu, "Aplikasi Sensor Vision untuk Deteksi MultiFace dan Menghitung Jumlah Orang," *Semin. Nas. Teknol. Inf. Komun. Terap. 2012 (Semantik 2012)*, vol. 2012, no. Semantik, pp. 26–33, 2012.
- [10] A. Tedeschi and F. Benedetto, "A real-time automatic pavement crack and pothole recognition system for mobile Android-based devices," *Adv. Eng. Informatics*, vol. 32, pp. 11–25, 2017.