

BAB III

METODE PENELITIAN

3.1 Gambaran Umum Penelitian

Pada penelitian tugas akhir penulis melakukan penelitian di *PT. JEIN*, Dengan fokus penelitian mengidentifikasi *Lcd Case*, yaitu dengan mengidentifikasi *scratch* pada *Lcd Case* agar tidak masuk atau lolos ke proses produksi, proses identifikasi *scracth* pada *Lcd Case* menggunakan pengolahan citra *digital* dengan metode *Canny Edge Detection*.

Proses identifikasi *Lcd Case scratch* menggunakan metode *Canny edge detection* 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*.

3.2 Pengambilan Data

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

1. akusisi citra

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

2. *Observasi*

Teknik peninjauan langsung ini sangat diperlukan guna memastikan permasalahan yang terjadi dan metode apa yang akan diambil guna menyelesaikan permasalahan tersebut. Disini penulis meninjau langsung area dan situasi yang ada di *Department Materian Control*.

3. *Interview*

Penulis melakukan *interview* dengan beberapa karyawan yang ada di *Department Material Control*, dari Pertanyaan yang penulis ajukan adalah :

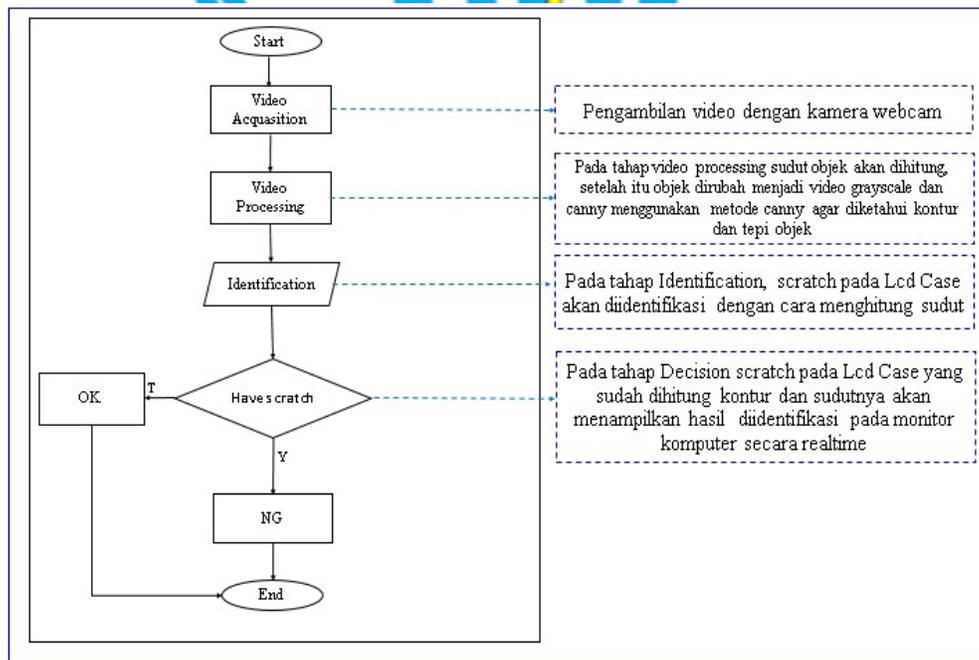
- a. Berapa banyak spesifikasi parts *defect*
- b. *Defect* yang paling sering ditemukan
- c. *Defect* yang sering lolos ke *department production*
- d. Apa kendala dalam mendeteksi *parts scratch*

Dari beberapa pertanyaan diatas penulis dapat memperoleh informasi mengenai kendala yang dihadapi *operator Department Material Control*.

3.3 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 : *Video Acquisition*, *Canny edge detection*, *Identification* dan *Result* seperti yang ditunjukkan pada gambar 3.1.

Berdasarkan dari hasil penelitian kemudian dilakukan analisa untuk mengetahui beberapa hal yang akan menjadi kesimpulan dari penelitian ini, yaitu pengujian akan dilakukan dengan menggunakan beberapa *Lcd Case* diantaranya *Lcd Case scratch* dan *Lcd Case* baik atau *ok*.



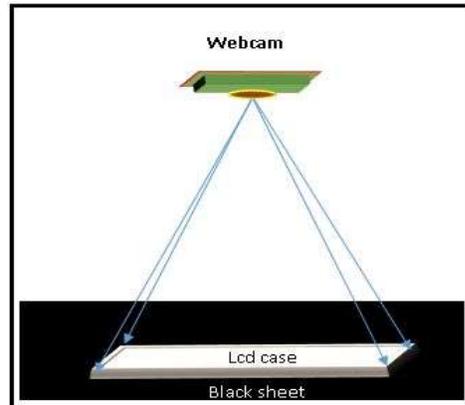
Gambar 3.1 Diagram Blok Rancangan sistem

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

ini :

1. *Video Acquisition*

Pada proses ini pengambilan video atau image *Lcd Case* dengan menggunakan kamera *webcam*.



Gambar 3.2 Image Acquisition

2. Video processing method

Pada tahap *video processing* sudut atau *angle* objek pada image atau video akan dihitung menggunakan rumus :

$$dx1 = pt1[0][0]-pt0[0][0]$$

$$dy1 = pt1[0][1]-pt0[0][1]$$

$$dx2 = pt2[0][0]-pt0[0][0]$$

$$dy2 = pt2[0][1]-pt0[0][1]$$

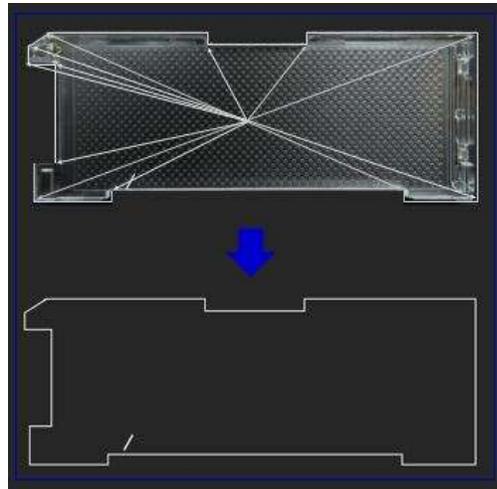
return

$$float((dx1*dx2+dy1*dy2))/math.sqrt(float((dx1*dx1+dy1*dy1))*(dx2*dx2+dy2*dy2)+1e-10)$$



Gambar 3.3 Image processing 1

setelah itu objek pada video dirubah menjadi video *grayscale* dan *canny* agar dapat di ketahui kontur dan tepi objek menggunakan metode *Canny edge detection*.



Gambar 3.4 Image processing 2

3. Identification

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 rumus :

```
if(len(approx) == 3): x,y,w,h = cv2.boundingRect(contours[i])
```

```
if(len(approx)>=4 and len(approx)<=6):
```

```
    vtc = len(approx)
```

```
    cos = []
```

```
    for j in range(2,vtc+1):
```

```
        cos.append(angle(approx[j%vtc],approx[j-2],approx[j-1]))
```

```
    cos.sort()
```

```
    mincos = cos[0]
```

```
    maxcos = cos[-1]
```

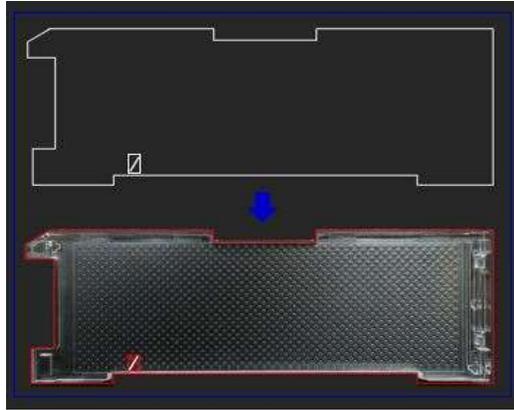
```
    area = cv2.contourArea(contours[i])
```

```
    x,y,w,h = cv2.boundingRect(contours[i])
```

```
    radius =w/2
```

```
if(abs(1 - (float(w)/h))<=2 and abs(1-area/(math.pi*radius*radius))<=0.2):
```

dari hasil perhitungan diatas akan menentukan apakah *scratch* melebihi ambang batas yang dibolehkan atau masih dibawah ambang batas.



Gambar 3.5 Identification scratch

4. *Decision*

Pada tahapan *decision* ini video *Lcd Case* yang sudah di hitung sudut dan diidentifikasi kontur dan tepinya menggunakan rumus dan perhitungan diatas, akan menampilkan hasil identifikasi, jika ada *scratch* pada *Lcd case* melebihi ambang batas yang ditentukan yaitu diameter atau ketebalan *scratch* 2mm dan panjang 10mm maka *scratch* akan teridentifikasi dengan menampilkan tulisan *scratch* diatas objek secara *realtime* pada monitor komputer.



Gambar 3.6 Result identification scratch

3.4 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.

3.5 Pengujian

Pada tahap proses pengujian, penulis akan menguji berapa besar tingkat akurasi pada proses identifikasi *Lcd case scratch* dengan metode *canny edge detection* di *Department Material Control*, dimana pada proses pengujian ini akan diketahui apakah aplikasi ini mampu mengidentifikasi *Lcd case scratch* secara realtime.

