

Analisis Citra Digital *Printed Circuit Board* dan *Gerber Data* Menggunakan Deteksi Tepi Canny dan Sobel

Ajar Rohmanu^[1], Andi Jumardi^[2]

Pasca Sarjana Magister Komputer Universitas Budi Luhur
Jln. Ciledug Raya, Petukangan Utara, Jakarta Selatan
12260 DKI Jakarta INDONESIA
ajarrohmanu@gmail.com^[1]
ardhypixel.ptiq@gmail.com^[2]

Arief Maulana^[3], Diko Muhammad Adam^[4]

Pasca Sarjana Magister Komputer Universitas Budi Luhur
Jln. Ciledug Raya, Petukangan Utara, Jakarta Selatan
12260 DKI Jakarta INDONESIA
ariefmaulanasugiantoro@gmail.com^[3]
dikomuhammadadam@gmail.com^[4]

Abstrak— Salah satu tahapan utama dalam pengolahan citra adalah proses pendeteksian tepi, dengan adanya proses ini batas antara objek dengan latar belakang dapat ditentukan dengan baik. Banyaknya metode yang bisa digunakan, akan tetapi jika tidak tepat dan sesuai dengan kondisi citra akan menimbulkan keraguan dalam pengambilan keputusan. Pada penelitian ini akan menggunakan pendeteksian tepi operator Canny dan Sobel untuk menganalisa dan mendeteksi tepi pada citra digital *Printed Circuit Board* dan citra digital *Gerber Data*, sehingga mampu memberikan informasi yang akurat sesuai citra yang digunakan. Dari hasil penelitian akan didapatkan operator mana yang cocok untuk mendeteksi tepi citra digital *Printed Circuit Board* dan citra digital *Gerber Data*. Hasil eksperimen menunjukkan bahwa operator Sobel kurang tepat jika digunakan untuk mendeteksi tepi citra yang mengandung noise, sedangkan operator Canny dapat mereduksi noise dengan baik.

Kata kunci : *Pendeteksian Tepi, Operator, citra, digital, noise, printed circuit board, gerber data*

I. PENDAHULUAN

Aplikasi pengolahan citra saat ini telah banyak digunakan hampir di segala bidang, antara lain adalah di bidang elektronika, kedokteran, fotografi, militer, dan geofisika. Salah satu tahapan utama dalam pengolahan citra tersebut adalah proses pendeteksian tepi citra, dengan proses ini tepi batas suatu objek dengan latar belakangnya dapat ditentukan dengan baik.

Menentukan letak tepi suatu citra adalah mudah, jika kondisi citra jelas dan tajam (*sharp*), namun ketepatan keberadaan tepi menjadi sulit ditentukan apabila dalam citra terdapat gangguan, seperti adanya *noise*. Kenyataannya citra hasil akuisisi, tidak selalu dalam keadaan bersih dari *noise*. Dengan alasan untuk mempermudah dan membantu menentukan keberadaan tepi telah banyak dikembangkan metode pendeteksian tepi citra, antara lain operator Sobel dan operator Canny. Sehingga para pengguna mempunyai banyak pilihan mengenai operator pendeteksi tepi tersebut.

Oleh karena banyaknya operator pendeteksi tepi yang tersedia, perlu pengetahuan yang tepat untuk

menentukan operator yang sesuai dengan kondisi citra yang akan dideteksi. Dengan demikian tepi yang dihasilkan merupakan tepi yang optimal dan tepat. Apabila pemilihan operator pendeteksi tepi tidak sesuai, akibatnya dapat menimbulkan tidak tepat tepi yang dihasilkan, efek lain yang timbul dapat mempengaruhi proses analisis selanjutnya. Dalam bidang elektronika khususnya proses pembuatan *Printed Circuit Board*, perlu adanya pengecekan hasil produksi dengan desain sehingga kualitas bisa terjaga. Berdasarkan keadaan inilah maka diperlukan pengetahuan atau informasi mengenai kinerja dari setiap operator, sehingga kesalahan dalam memilih operator untuk pendeteksian tepi citra dapat dihindari.

Berdasarkan pada masalah yang telah disebutkan di atas penelitian ini bertujuan melakukan analisis terhadap kinerja pendeteksian tepi citra dari operator Canny dan operator Sobel, sehingga dapat memberikan informasi yang akurat mengenai kinerja masing-masing operator dalam penggunaan mendeteksi citra *Printed Circuit Board* dan citra *gerber data*.

II. LANDASAN TEORI

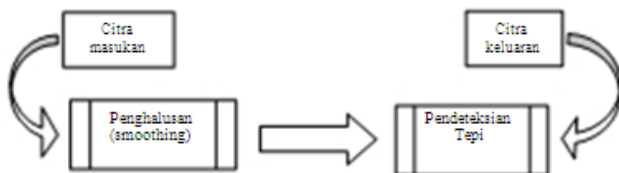
A. Metode Pendeteksian Tepi

Secara umum tepi dapat didefinisikan sebagai batas antara dua region (dua *pixel* yang saling berdekatan) yang memiliki perbedaan intensitas yang tajam (tinggi) [1]. Definisi ini tidak sepenuhnya berlaku bila tepi tersebut mengalami efek *blur*. Efek ini dapat mengakibatkan batas antara dua region mengalami perubahan secara landai sehingga tepi antara kedua region tersebut menjadi kurang jelas. Definisi di atas tidak pula berlaku sepenuhnya bila pada batas antara dua region mengalami gangguan *noise*. Karena intensitas *noise* bersifat acak, maka keberadaan *noise* dapat memunculkan tepi-tepi lain di sekitar tepi yang sebenarnya, selain itu dapat pula menggeser posisi tepi yang sebenarnya.

Untuk mendefinisikan tepi dapat ditentukan dengan cara sederhana, yaitu menghitung perbedaan intensitas antara dua piksel yang saling berdekatan. Metode ini sering disebut dengan turunan pertama atau gradien citra. Salah satu contoh metode gradien ini

adalah *metode gradien Roberts*. Selain metode gradien, tepi citra juga dapat ditentukan dengan melalui turunan kedua citra, dikenal dengan *metode Laplacian*, yang menyatakan tepi berada pada *zero-crossing*, sementara untuk turunan pertama terletak pada nilai maksimum lokalnya. Dibandingkan metode gradien, *metode Laplacian* menghasilkan tepi yang lebih baik karena menghasilkan tepi yang cukup tipis, sedangkan kelemahan *Laplacian* sangat sensitif terhadap adanya *noise*.

Seperti diketahui bahwa citra hasil akuisisi dapat mengandung berbagai karakteristik, diantaranya *noise* dan *blur*, maka sebagai solusi dari kendala tersebut telah dikembangkan metode lain dengan cara menambahkan proses penghalusan (*smoothing*) sebelum proses pendeteksian tepi. Secara garis besar pengembangan metode ini seperti bagan pada gambar 1:



Gambar 1. Pengembangan Proses Pendeteksian Tepi

B. Operator Canny

Pada tahun 1986 John Canny mengusulkan 3 kriteria yang menjadi basis pengembangan filter untuk mengoptimalkan pendeteksian tepi pada citra memiliki *noise*. Ke tiga kriteria tersebut adalah [2] :

- *Good detection*, kriteria ini bertujuan memaksimalkan nilai *signal to noise ration* (SNR) sehingga semua tepi dapat terdeteksi dengan baik atau tidak ada yang hilang.
- *Good localisation*, tepi yang terdeteksi berada pada posisi yang sebenarnya, atau dengan kata lain bahwa jarak antara posisi tepi yang terdeteksi oleh detektor dengan posisi tepi sebenarnya adalah seminimum mungkin (idealnya = 0).
- *Low multiplicity of the response atau "one response to single edge"* detektor tidak memberikan tepi yang bukan tepi sebenarnya. Berdasarkan pada kriteria ini Canny berhasil melakukan optimalisasi dari ke 3 kriteria tersebut dan menghasilkan persamaan (1), namun persamaan ini cukup sulit untuk diimplementasikan.

$$h(x) = a_1 e^{\cos(\omega x)} + a_2 e^{\sin(\omega x)} + a_3 e^{-\cos(\omega x)} + a_4 e^{-\sin(\omega x)}$$

sehingga pada implementasinya, Canny tetap menggunakan *filter Gaussian* untuk mereduksi *noise* dan dilanjutkan dengan penghitungan turunan pertama dan *thresholding hysteresis*.

C. Operator Sobel

Sobel termasuk yang mengembangkan metode ini.

Proses penghalusan yang digunakan merupakan proses konvolusi dari jendela yang ditetapkan terhadap citra yang dideteksi. Agar perkiraan gradien tepat ditengah jendela, dalam konvolusi Sobel menggunakan jendela 3x3, dan susunan piksel-piksel disekitar piksel (x,y) seperti bagan berikut [3] :

p_1	p_2	p_3
p_6	(x,y)	p_4
p_7	p_8	p_5

Sehingga besar gradien dihitung dengan menggunakan persamaan:

$$s_x = (p_3 + cp_4 + p_5) - (p_1 + cp_6 + p_7)$$

$$s_y = (p_1 + cp_2 + p_3) - (p_7 + cp_8 + p_5)$$

dengan c konstanta yang bernilai 2. Sehingga matriks *operator Sobel* seperti yang ditunjukkan di bawah ini:

$$s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \text{ dan}$$

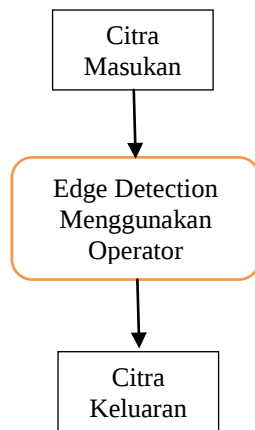
$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Dari matriks di atas terlihat bahwa *Sobel* memberikan pembobotan pada piksel-piksel yang lebih dekat dengan titik pusat.

III. METODE PENELITIAN

Dalam melakukan penelitian deteksi tepi *Printed Circuit Board* dan *Gerber data* ini menggunakan aturan sebagai berikut:

- 1) Penelitian ini dimulai dengan mengambil citra digital *Printed Circuit Board* dan *Gerber data*.
- 2) Citra digital yang sudah diperoleh selanjutnya dilakukan pendeteksian tepi dengan metode *Operator Canny* dan *Operator Sobel* dengan merubah nilai ambang.
- 3) Langkah-langkah penelitian tersebut dapat digambarkan sesuai skema berikut ini:



Gambar 2. Tahapan Penelitian Pendeteksian Tepi Citra

- 4) Menampilkan histogram untuk citra sebelum deteksi tepi dan setelah dilakukan deteksi tepi, selanjutnya membandingkan hasil dari *Operator Canny* dan *Operator Sobel* mana yang lebih sesuai untuk mendeteksi tepi citra digital *PrintedCircuit Board* dan *Gerber data*.

IV. HASIL PEMBAHASAN

Berikut ini adalah hasil dari uji coba penelitian yang dilakukan:

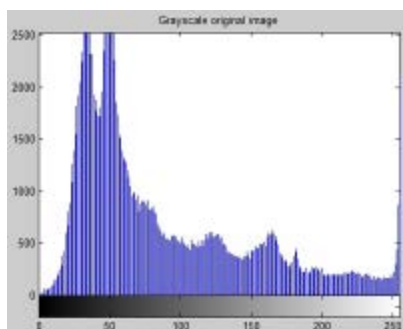
A. Sobel Detection

Berikut ini adalah citral digital yang akan dilakukan deteksi tepi operator *Sobel*:

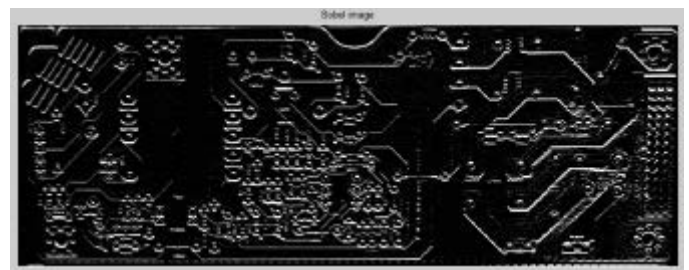
1) Printed Circuit Board



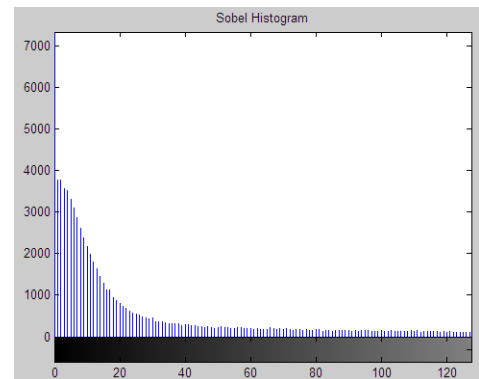
Gambar 3. Citra Digital Printed Circuit Board



Gambar 4. Histogram Printed Circuit Board



Gambar 5. Hasil citra Deteksi Sobel



Gambar 6. Histogram Sobel Printed Circuit Board

Hasil *Operator Sobel* yang ditambahkan perhitungan gradient dari citra digital dan threshold untuk mengatur kualitas deteksi tepi dengan Script dan gambar berikut:

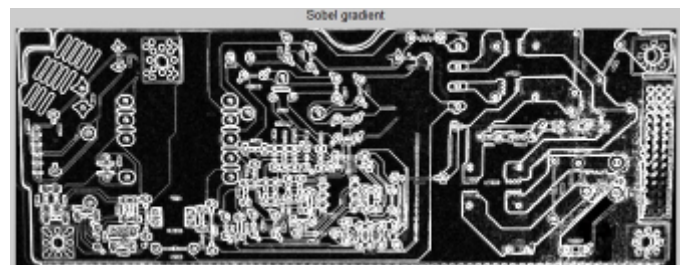
Script Gradien [4]

```

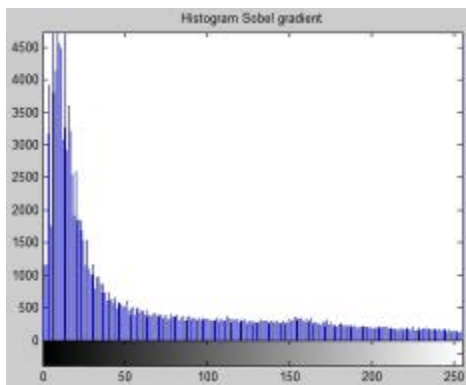
for i=1:size(C,1)-2
    for j=1:size(C,2)-2
        %Sobel mask for x-direction:
        Gx=((2*C(i+2,j+1)+C(i+2,j)+C(i+2,j+2))-(2*C(i,j+1)+C(i,j)+C(i,j+2)));
        %Sobel mask for y-direction:
        Gy=((2*C(i+1,j+2)+C(i,j+2)+C(i+2,j+2))-(2*C(i+1,j)+C(i,j)+C(i+2,j)));
        %The gradient of the image
        B(i,j)=sqrt(Gx.^2+Gy.^2);
    end
end
    
```

```

Script Threshold:
%Define a threshold value
Thresh=100;
B=max(B,Thresh);
B(B==round(Thresh))=0;
B=uint8(B);
    
```



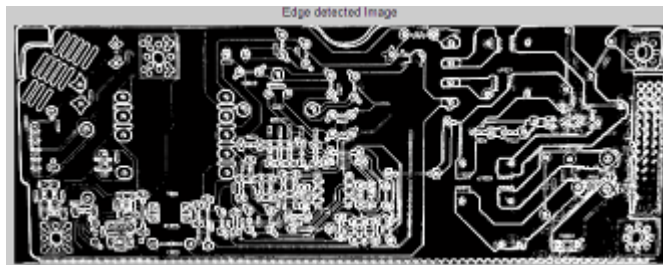
Gambar 7. Hasil Citra Dektesi Sobel dengan gradient



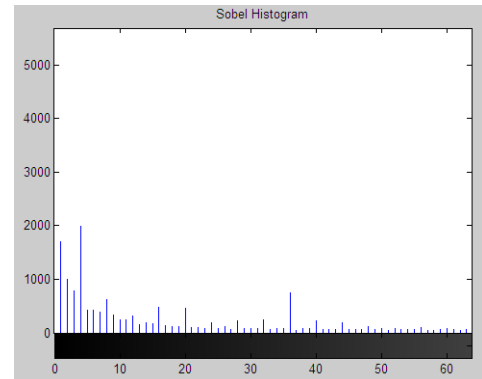
Gambar 8. Histogram Sobel dengan Gradient



Gambar 11. Hasil citra deteksi Sobel



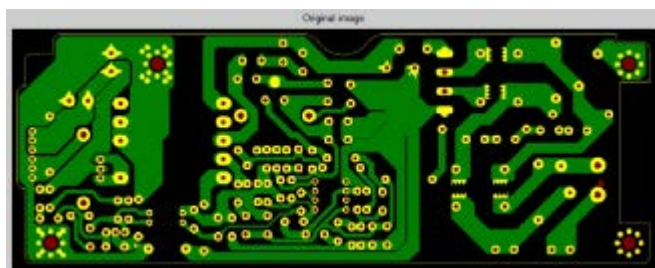
Gambar 9. Hasil citra deteksi sobel dengan threshold



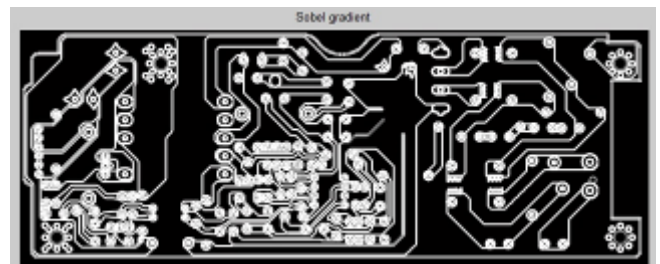
Gambar 12. Histogram Sobel Gerber data

Pada *Printed Circuit Board* memiliki noise yang sangat berpengaruh hasil deteksi tepi dengan operator Sobel.

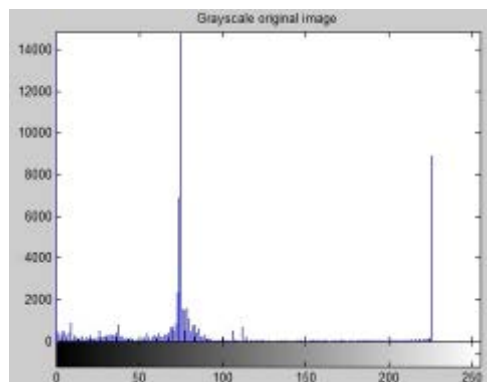
2) Gerber data



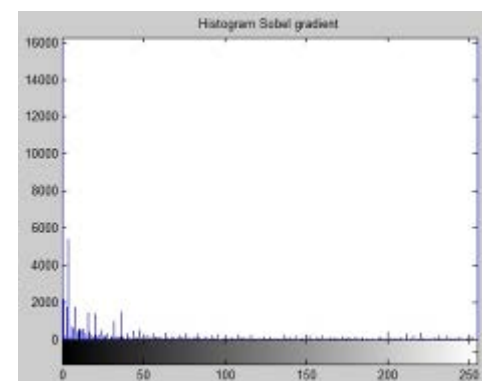
Gambar 10. Citra Digital Gerber data



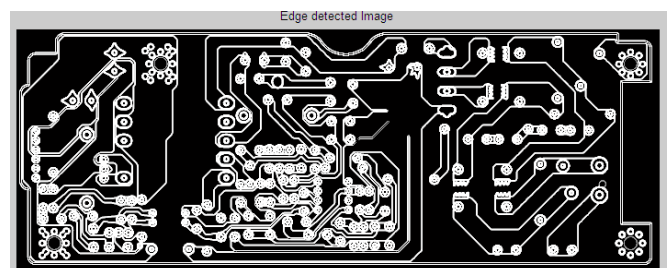
Gambar 13. Hasil Citra Dekteksi Sobel dengan gradient



Gambar 10. Histogram Gerber data



Gambar 14. Histogram Sobel dengan Gradient



Gambar 15. Hasil citra deteksi sobel dengan threshold

Pada *Gerber data* memiliki noise lebih sedikit dari pada *Printed Circuit Board* sehingga deteksi tepi dengan operator *Sobel* menghasilkan kualitas yang lebih bagus.

B. Canny Detection

Berikut ini adalah Script dan hasil deteksi tepi menggunakan *Operator Canny* dengan sumber citra digital yang sama dengan deteksi tepi operator *Sobel*.

Script Threshold dan Gaussian [5]

```
%Value for Thresholding
T_Low = 0.015;
T_High = 0.075;
%Gaussian Filter Coefficient
B = [2, 4, 5, 4, 2; 4, 9, 12, 9, 4; 5, 12, 15, 12, 5; 4, 9, 12, 9, 4; 2, 4, 5, 4, 2];
B = 1/159.* B;
%Convolution of image by Gaussian Coefficient
A=conv2(img, B, 'same');
%Filter for horizontal and vertical direction
KGx = [-1, 0, 1; -2, 0, 2; -1, 0, 1];
KGy = [1, 2, 1; 0, 0, 0; -1, -2, -1];
```

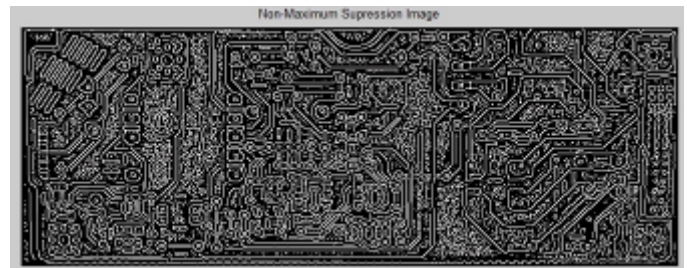
Script Negative directions:

```
for i=1:pan
    for j=1:leb
        if (arah(i,j)<0)
            arah(i,j)=360+arah(i,j);
        end; end; end;
    arah2=zeros(pan, leb);
    for i = 1 : pan
        for j = 1 : leb
            if ((arah(i, j) >= 0) && (arah(i, j) < 22.5) || (arah(i, j) >= 157.5) && (arah(i, j) < 202.5) || (arah(i, j) >= 337.5) && (arah(i, j) <= 360))
                arah2(i, j) = 0;
            elseif ((arah(i, j) >= 22.5) && (arah(i, j) < 67.5) || (arah(i, j) >= 202.5) && (arah(i, j) < 247.5))
                arah2(i, j) = 45;
            elseif ((arah(i, j) >= 67.5 && arah(i, j) < 112.5) || (arah(i, j) >= 247.5 && arah(i, j) < 292.5))
                arah2(i, j) = 90;
            elseif ((arah(i, j) >= 112.5 && arah(i, j) < 157.5) || (arah(i, j) >= 292.5 && arah(i, j) < 337.5))
                arah2(i, j) = 135;
            end; end; end;
```

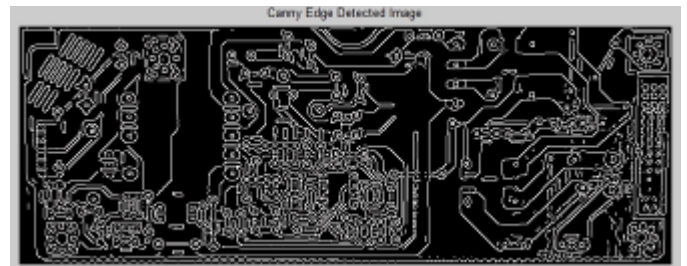
Script Non_maximum Supression [6]

```
for i=2:pan-1
    for j=2:leb-1
        if (arah2(i,j)==0)
            BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i,j+1), magnitude2(i,j-1)]));
        elseif (arah2(i,j)==45)
            BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i+1,j-1), magnitude2(i-1,j+1)]));
        elseif (arah2(i,j)==90)
            BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i+1,j), magnitude2(i-1,j)]));
        elseif (arah2(i,j)==135)
            BW(i,j) = (magnitude2(i,j) == max([magnitude2(i,j), magnitude2(i+1,j+1), magnitude2(i-1,j-1)]));
        end; end; end;
```

1) Printed Circuit Board



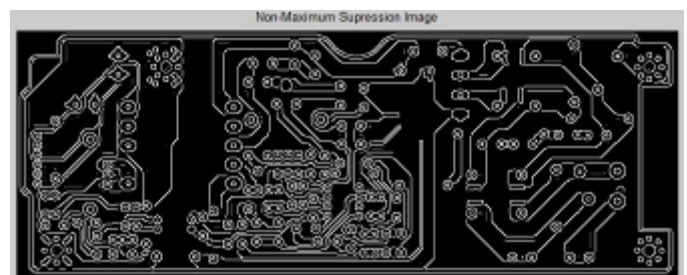
Gambar 16. Hasil citra deteksi Canny Non-maximum Supression



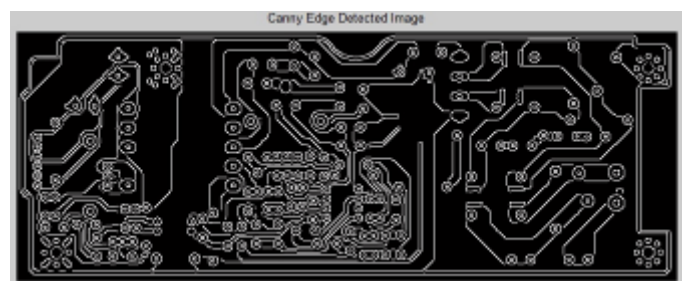
Gambar 17. Hasil citra deteksi Canny

Pada gambar 16 diperoleh citra yang terlihat memiliki noise yang banyak karena citra digital *Printed Circuit Board* sebagai sumber sehingga dilakukan penghilangan noise dan melakukan setting threshold dimana akan menghasilkan citra deteksi tepi yang memiliki kualitas lebih bagus seperti terlihat pada gambar 17.

2) Gerber data



Gambar 18. Hasil citra deteksi Canny Non-maximum Supression

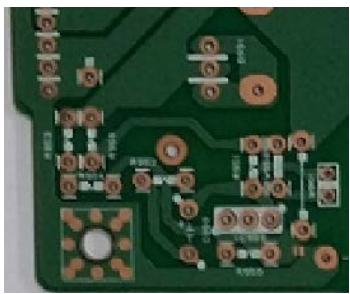
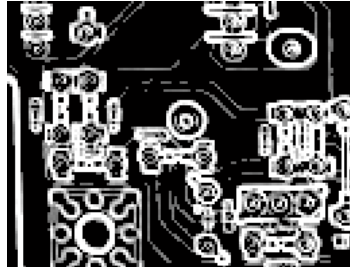
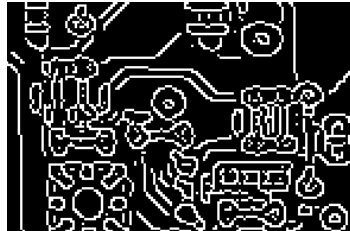


Gambar 19. Hasil citra deteksi Canny

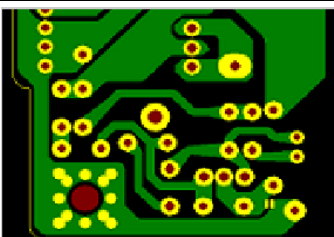
Pada gambar 18 terlihat hasil citra memiliki sedikit noise sehingga ketika dilakukan perubahan nilai threshold akan menghasilkan citra yang lebih baik dan lebih jelas, seperti terlihat pada gambar 19.

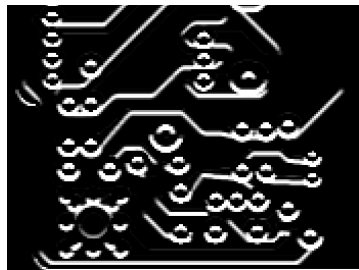
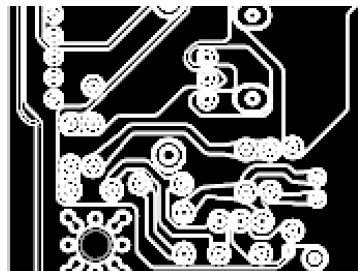
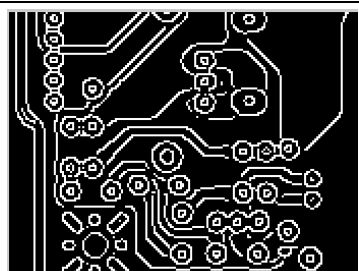
C. Perbandingan Sobel dan Canny Detection

1) Citra Printed Circuit Board

	Tampilan yang diperbesar dari citra asli dari <i>Printed Circuit Board</i>
	Tampilan yang diperbesar dari hasil operator Sobel
	Tampilan yang diperbesar dari hasil operator sobel dengan menambahkan perhitungan gradient citra digital dan nilai threshold
	Tampilan yang diperbesar dari hasil operator Canny yang ditambahkan Gaussian filtering, negative direction nearest, non-maximum suppression serta nilai threshold

2) Citra Gerber data

	Tampilan yang diperbesar dari citra asli dari <i>Data gerber</i>
---	--

	Tampilan yang diperbesar dari hasil operator Sobel
	Tampilan yang diperbesar dari hasil operator sobel dengan menambahkan perhitungan gradient citra digital dan nilai threshold
	Tampilan yang diperbesar dari hasil operator Canny yang ditambahkan Gaussian filtering, negative direction nearest, non-maximum suppression serta nilai threshold

V. KESIMPULAN

Dari hasil penelitian dan pengujian maka dapat diambil kesimpulan sebagai berikut:

1. Hasil pendeteksian tepi untuk *Printed Circuit Board* dan *Gerber Data* menggunakan operator Canny lebih baik dari pada operator Sobel.
2. Nilai *Threshold* akan mempengaruhi kualitas deteksi tepi pada Operator Canny dan operator Sobel.
3. Untuk meningkatkan output citra digital dari operator Canny ditambahkan *Gaussian filtering*, *negative direction to nearest*, *non-maximum suppression* serta *threshold* untuk mendapatkan hasil yang terbaik, untuk meningkatkan output dari operator Sobel ditambahkan perhitungan *gradient* citra digital.
4. Ukuran Citra digital sangat mempengaruhi hasil pendeteksian tepi objek, dimana semakin kecil ukuran citra hasil deteksi semakin jelas.

REFERENCES

- [1] Gonzalez, R.C and P.Wintz. "*Digital Image Processing*". Second Edition, Addison Wesley, 1987.
- [2] Canny, J. "A Computational *Approach To Edge Detection*". IEEE on PAMI. vol. 8, pp. 679-697, 1986.
- [3] Sobel, I. "*An isotropic image gradient operator*". In H. Freeman, editor, *Machine Vision for Three-Dimensional*, 1990.
- [4] Oge Marques," *Practical image and video processing using Matlab*". John Wiley & Sons, 2011.
- [5] Wijaya, M. C., Prijono, A., "*Pengolahan Citra Digital Menggunakan Mathlab*", Penerbit Informatika, Bandung, 2007.
- [6] Abdul kadir & Adhi Susanto., "*Teori dan Aplikasi Pengolahan Citra*", Andi, Jogjakarta, 2013.