

## PENERAPAN PENGOLAHAN CITRA UNTUK PERBAIKAN GAMBAR 2 DIMENSI DENGAN MENGGUNAKAN MATLAB

**Ahmad Ridho'i**

*Program Studi Teknik Elektro, Fakultas Tekni, Universitas 17 Agustus 1945 Surabaya*

*Jl. Semolowaru 45 Surabaya*

*Telp : 031-5931800*

*E-mail: [ridhoi@untag-sby.ac.id](mailto:ridhoi@untag-sby.ac.id)*

### **ABSTRAK**

Citra adalah sebuah gambar 2 dimensi yang dapat menghasilkan atas beberapa informasi. Banyak informasi yang bisa diperoleh dari sebuah citra apabila citra tersebut menampilkan kondisi dari obyek yang telah di ubah menjadi gambar 2 dimensi, sehingga warna, bentuk, ukuran, dan parameter-parameter lainnya dapat diketahui. Apabila citra tidak baik atau tidak jelas serta tidak sesuai yang diharapkan maka perlu adanya perbaikan citra. Pada dalam beberapa citra kadang ditemukan citra yang tidak sesuai dengan harapan karena rusak atau terjadi perubahan warna dan cacat pixel. Hal ini dapat disebabkan adanya noise pada citra ataupun kerusakan pada warna di sebabkan lamanya gambar tersebut. Maka dari itu untuk mengatasi berbagai masalah kerusakan pada citra perlu adanya perbaikan citra. Untuk perbaikan pada citra terdapat dua hal yaitu warna atau keabuan dari citra dan pola (patern) dari citra. Untuk perbaikan citra dengan kerusakan pada warna dengan area pixel melebar tidak berbentuk pola tapi cenderung menempati luasan dari pixel sedangkan kerusakan pola umumnya berbentuk terputusnya kesamaan nilai dari pixel. Untuk melakukan pengolahan citra atau perbaikan citra pada tingkat pixel dapat menggunakan matlab karena pada matlab tersedia fungsi-fungsi atau perintah/coding yang berkaitan dengan pengolahan citra. Selain perintah/coding matlab juga menyediakan form yang lebih representatif untuk membuat tampilan citra. GUI merupakan graphical user interface yang dapat digunakan untuk menampilkan citra dari sebelum perbaikan dan setelah dilakukan perbaikan.

**Kata Kunci:** Citra, foto, GUI, Matlab, Pixel, Pola (patern), pengolahan.

### **ABSTRACT**

*Image is a 2-dimensional image that can produce some information. A lot of information can be obtained from an image if the image displays the condition of the object that has been converted into a 2-dimensional image, so that the color, shape, size and other parameters can be known. If the image is not good or not clear and is not as expected, it is necessary to improve the image. In some images, images are sometimes found that are not in accordance with expectations because they are damaged or there is a change in color and pixel defects. This can be caused by noise in the image or damage to the color caused by the length of the image. Therefore, to overcome various problems of damage to the image, it is necessary to improve the image. To improve the image, there are two things, namely the color or gray of the image and the pattern (pattern) of the image. For image repair with damage to color with a widened pixel area, it does not form a pattern but tends to occupy the area of the pixel, while pattern damage generally takes the form of breaking the similarity of the pixel values. To perform image processing or image repair at the pixel level, you can use Matlab because Matlab provides functions or commands/coding related to image processing. In addition to commands/coding, Matlab also provides a more representative form for displaying images. The GUI is a graphical user interface that can be used to display images from before and after repairs.*

**Keywords** : Image, photo, GUI, Matlab, Pixel, Pattern, processing.

### **1. PENDAHULUAN**

Pada industri modern saat ini teknologi sangat memiliki peranan penting dalam mempermudah proses pekerjaan manusia. Terutama di industri banyak unit-unit yang dulunya dikerjakan manusia banyak digantikan teknologi sehingga peran teknologi sangat mempengaruhi proses di industri. Teknologi yang tidak kalah pentingnya adalah teknologi vision dengan menerapkan teknologi vision

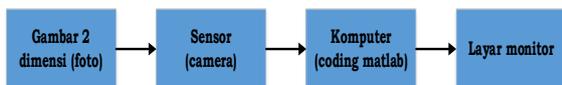
maka proses bisa lebih akurat dalam karena secara visual objek bisa diketahui lebih jelas. Bagian vision diantaranya adalah pengolahan citra yang dihasilkan dari sensor visual dari kamera. Sehingga hal-hal yang terkait dengan pengolahan citra menjadi hal penting yaitu: perbaikan, analisa, kompresi citra, restorasi, transmission citra, rekontruksi citra, dan represntasi citra(Jain, 1989). Salah satu sistem perbaikan citra tersebut adalah sistem pengenalan dan pengolahan

citra dengan spesifikasi ukuran pixel dan format file. Sistem ini meniru algoritma pemikiran manusia untuk mengenal citra, yaitu proses mengamati dan membandingkan citra sebelum dan sesudah mengalami perbaikan. Pengolahan citra merupakan salah satu teknik yang digunakan untuk memproses gambar dan memanipulasinya agar diperoleh gambar yang diinginkan. Hal tersebut diperlukan karena setiap elemen gambar atau pixel membawa paling tidak dua informasi, yaitu letak dan warna. Jika sebuah gambar mempunyai ukuran 320 x 200 pixel, maka setidaknya 320 x 200 x 2 informasi atau 128.000 informasi (muharom syahri, 2021). Sehingga citra yang merupakan gambar 2 dimensi rentan terjadi perubahan terhadap warna gambar maupun rusak gambar karena noise. Sehingga untuk melakukan perbaikan atau rekonstruksi citra diperlukan agar dapat mendekati original dari citra.

Salah satu perangkat/software yang dapat digunakan untuk melakukan pengolahan citra yaitu matlab karena di matlab sudah tersedia fungsi-fungsi/coding yang terkait dengan citra/image. Selain fungsi yang sudah tersedia di matlab juga tersedia GUI (graphic user interface) yang memungkinkan untuk berkreasi sebuah tampilan untuk citra tidak hanya informasi nilai bisa ditampilkan berupa gambar (Gonzalez Rafael C, 2002).

## 2. PEMBAHASAN

Pada penelitian kali ini karena yang diteliti hanya berupa coding maka perencanaan hanya terkait pada alur dari coding yang digunakan untuk perbaikan citra pada gambar 2 dimensi. Untuk itu dalam sistem dapat digambarkan dalam bentuk diagram blok seperti di bawah:



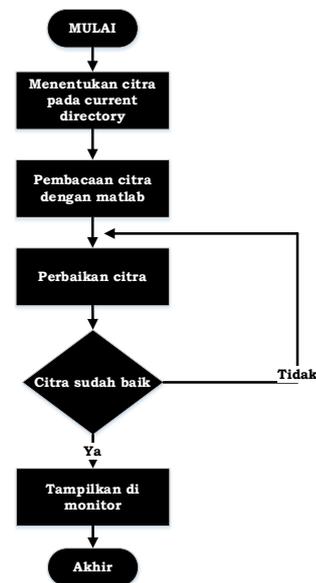
**Gambar 1. Sistem perbaikan citra dengan matlab**

Dengan demikian dari gambar 1. untuk masing-masing blok dapat dijelaskan sebagai berikut :

- Gambar 2 dimensi, merupakan gambar dalam bentuk cetak (printed) atau foto, untuk itu gambar 2 dimensi memiliki resolusi yang merupakan hasil dari cetak atau printed. Sehingga ketajaman cetak terhadap gambar 2 dimensi dari soft file menjadi gambar 2 dimensi cetak mempengaruhi baik dan tidaknya resolusi gambar (Jähne, 2005).
- Sensor (kamera/scanner), Scanner merupakan alat untuk membaca gambar 2 dimensi atau foto yang hasilnya berupa file gambar yang selanjutnya dapat disimpan dalam directory komputer. Dalam melakukan pembacaan gambar 2 dimensi menjadi soft file (file gambar) juga ditentukan oleh resolusi yang dipilih makin tinggi resolusi makin besar ukuran soft file, dan juga berbanding lurus dengan banyaknya pixel yang dimiliki oleh gambar 2 dimensi yang berupa soft file (muharom syahri, 2021).

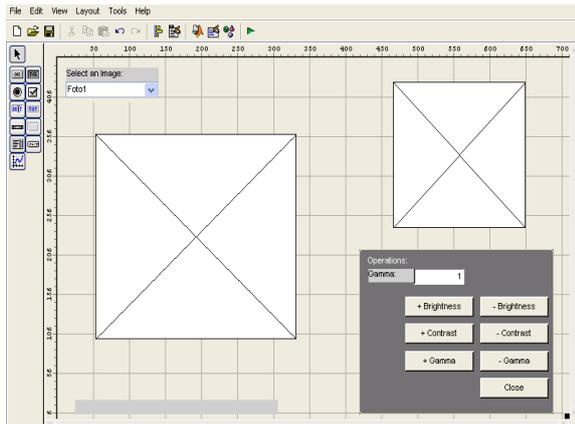
- Komputer, komputer merupakan tempat menyimpan file gambar hasil scanner/kamera dalam directoy tertentu dan juga tempat untuk melakukan proses pengolahan citra. Pada proses pengolahan citra ini dilakukan dengan menggunakan perangkat lunak, GUI pada matlab (Pratt, 2001).
- Monitor, merupakan perangkat yang berfungsi menampilkan berupa gambar 2 dimensi. Untuk itu hasil dari pengolahan oleh processor yang mengikuti coding matlab untuk pengolahan citra (Woods, 2001).

Dengan demikian untuk system perbaikan citra gambar 2 dimensi menggunakan alur seperti diagram blok di atas yang mana menjadi urutan untuk melakukan pengolahan citra pada gambar 2 dimensi. Masing-masing blok memiliki peran yang penting, sehingga dari blok-blok hanya blok pengolahan citra dengan matlab yang ada dikomputer memerlukan perancangan software, sedangkan yang lain tidak memerlukan pengaturan apapun. Untuk itu diperlukan perencanaan software agar menghasilkan algoritma logika program yang mendekati apa yang diperlukan untuk perbaikan citra pada gambar 2 dimensi (Sundararajan, 2017). Flowchart coding untuk perbaikan citra di bawah:



**Gambar 2. Flowchart pengolahan citra**

Setelah pengambilan objek foto dengan kamera atau scanner file gambar didapat selanjutnya menentukan current directory yang merupakan tempat file yang akan diolah oleh matlab. Hal tersebut bisa dilakukan dengan dua cara yaitu dengan meletakkan file ke current directory yang aktif di matlab atau mengarahkan panggilan file ke directory mana letak file tersebut berada (jensen John r, 2015). Langkah awal dalam penggunaan matlab untuk perbaikan citra dengan merancang GUI (graphic user interface) yang ada di dalam matlab, dengan menentukan komponen-komponen yang diperlukan seperti di bawah:



**Gambar 3. Rancangan GUI**

Dari perancangan GUI di atas secara otomatis terbentuk M-file yang berisikan fungsi dari masing-masing komponen secara otomatis memiliki fungsi yang dapat diisi coding yang diinginkan dan juga properties yang dapat digunakan untuk mengatur tampilan dari komponen tersebut (Blanchet gerard & Charbit maurice, 2006).

Pada proses perbaikan citra ini di mulai dengan pembacaan citra atau informasi tentang citra yang biasa disebut proses pembacaan. Pembacaan citra ini sangat diperlukan untuk mengetahui citra sebelum perbaikan supaya dapat mempermudah di dalam melakukan perbaikan citra. Di bawah ini merupakan struktur fungsi dalam membaca citra, dimana citra yang digunakan sebagai object adalah foto yang mengalami kerusakan dan tersimpan pada directory komputer (D:\MATLAB6p5\demos) dengan nama file foto.jpg. Berikut coding untuk melakukan pembacaan citra.

```
function coba1_OpeningFcn(hObject, eventdata, handles, varargin)
set(hObject, 'Units', 'pixels');
handles.foto = imread([matlabroot filesep 'demos' filesep 'foto.jpg']);
info = imfinfo([matlabroot filesep 'demos' filesep 'foto.jpg']);
position = get(hObject, 'Position');
set(hObject, 'Position', [position(1:2) info.Width + 10 info.Height + 100]);
axes(handles.axes1);
image(handles.foto)
set(handles.axes1, ...
    'Visible', 'off', ...
    'Units', 'pixels', ...
    'Position', [50 50 info.Width info.Height]);
```

Ada tiga hal yang perlu diperhatikan dalam proses perbaikan citra, yaitu brightness, colormap, dan contrast. Ketiga hal tersebut mempunyai fungsi dan struktur fungsi masing-masing yang sangat berguna untuk mengolah gambar sesuai yang perlukan.

Proses **brightness** bertujuan untuk mendapatkan warna gambar terang dan gelap. Proses brightness mengubah komposisi warna pada gambar menjadi warna putih terang atau warna gelap sesuai dengan intensitas warna yang diinginkan. Fungsi brightness pertama-tama akan menentukan kecerahan dari gambar. Berikut ini merupakan struktur fungsi brightness seperti di bawah.

```
function newmap = brighten(map,beta)
if nargin < 2, beta = map; map = colormap; end
if nargin==2 & length(map)==1, % brighten(fig,beta) fig = map;
if floor(fig)~=fig | isempty(find(fig==get(0,'Children'))), return,
end
set(fig,'Colormap',brighten(get(fig,'Colormap'),beta),...
Color',brighten(get(fig,'color'),beta)
for h=get(fig,'Children'), if strcmp(get(h,'type'),'axes'),
set(h,'color',brighten(get(h,'color'),beta),...
'xcolor',brighten(get(h,'xcolor'),beta),...
'ycolor',brighten(get(h,'ycolor'),beta),...
'zcolor',brighten(get(h,'zcolor'),beta),...
'colororder',brighten(get(h,'colororder'),beta)) for
hh=[get(h,'title') get(h,'xlabel') get(h,'ylabel') get(h,'zlabel')],
set(hh,'color',brighten(get(hh,'color'),beta)) end
for hh = get(h,'Children'), if strcmp(get(hh,'type'),'text'),
set(hh,'color',brighten(get(hh,'color'),beta))
elseif strcmp(get(hh,'type'),'line'),
set(hh,'color',brighten(get(hh,'color'),beta))
ec = get(hh,'MarkerEdgeColor');
fc = get(hh,'MarkerFaceColor');
if ~isstr(ec), set(hh,'MarkerEdgeColor',brighten(ec,beta)), end
if ~isstr(fc), set(hh,'MarkerFaceColor',brighten(fc,beta)), end
elseif strcmp(get(hh,'type'),'patch'),ec = get(hh,'EdgeColor');
fc = get(hh,'FaceColor');
if ~isstr(ec), set(hh,'EdgeColor',brighten(ec,beta)), end
if ~isstr(fc), set(hh,'FaceColor',brighten(fc,beta)), end
end end end end
Return end
if max(max(map)) > 1
map = map/255; End
if (beta > 1) | (beta < -1)
error('Beta must be between -1 and 1.') end
tol = sqrt(eps);
if beta > 0
gamma = 1 - min(1-tol,beta);
else
gamma = 1/(1 + max(-1+tol,beta));end
map = map.^gamma;
if nargin == 0 colormap(map)
Else
newmap = map; end
```

Proses **color** pada gambar bertujuan untuk mengatur elemen warna gambar sesuai yang diinginkan. Proses ini diperlukan untuk memperbaiki elemen warna pada gambar yang mengalami kerusakan. Warna pada gambar diubah dengan menggunakan colorbar, karena dengan colorbar dapat mengatur warna yang diinginkan hanya dengan menggeser colorbar yang telah tersedia. Struktur fungsi dari proses colorbar dapat dilihat seperti di bawah ini.

```
function handle=colorbar(arg1, arg2, arg3)
narg = nargin;
GCF =(gcf);
GCA = gca;
if narg<2
haxes = gca; hfig = gcf;
if narg<1
loc = 'vert';
else
if strcmp(arg1,'peer')
error('Parameter "peer" must be followed by an axes handle.');
```

```
End
loc = arg1; end
elseif narg == 2
if strcmp(arg1,'peer')
if ishandle(arg2) & strcmp(get(arg2, 'type'), 'axes')
haxes = arg2;
hfig = get(haxes, 'parent');
loc = 'vert';
narg = 0;
else
error('Second argument must be a scalar axes handle.');
```

```
end
Else
error('Unknown command option.');
```

```
end
Else
loc = arg1;
if strcmp(arg2,'peer')
if ishandle(arg3) & strcmp(get(arg3, 'type'), 'axes')
haxes = arg3;
hfig = get(haxes, 'parent');
narg = 1;
else
error('Third argument must be a scalar axes handle.');
```

```
end
else error('Unknown command option.');
```

```
end end
if narg==1 & strcmp(loc,'delete')
ax = gcbo;
if strcmp(get(ax,'tag'),'ColorbarDeleteProxy')
cbo = ax;
ax = get(cbo,'userdata');
if ishandle(ax)
ud = get(ax,'userdata');
if isfield(ud,'ColorbarDeleteProxy') & ...
isequal(ud.ColorbarDeleteProxy,cbo) & ...
ishandle(ax)
delete(ax) end end end
.....
```

Proses contrast pada gambar bertujuan untuk mengatur color gray scale pada gambar. Contrast akan bekerja dengan baik ketika warna gambar dipadukan dengan intensitas kecerahan. Berikut ini struktur fungsi dari proses contrast.

```
function cmap = contrast(x,m);
if nargin < 2, m = size(colormap,1); end
xmin = min(min(x));
xmax = max(max(x));
x = round((m-1)*(x-xmin)/(xmax-xmin));
f = find(diff(sort([x(:); (0:m)'])))';
f = f/max(f);
cmap = [f f];
```

Pengujian ini dilakukan untuk mengetahui tingkat keberhasilan. Selain itu dengan adanya pengujian, dapat diketahui adanya kelemahan atau kekurangan yang ada pada citra, sehingga dapat dilakukan beberapa perbaikan yang diperlukan.

Titik berat pengujian terletak pada pengujian perangkat lunak saja. Sedangkan untuk perangkat kerasnya yang merupakan piranti pendukung, tidak akan dibahas, karena perangkat keras sudah ada dalam bentuk jadi. Ada beberapa langkah-langkah Di

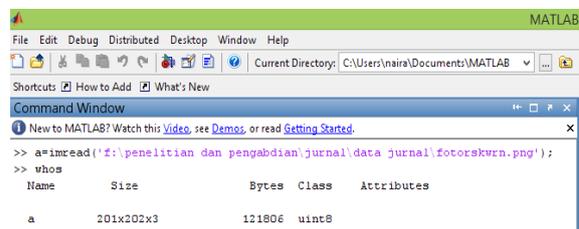
dalam melakukan pengujian citra. Dibawah ini akan dijelaskan langkah-langkah pengujian perangkat lunak pengolahan citra dari awal pembacaan citra sampai hasil perbaikan citra.

**a. Pembacaan Citra**

Pembacaan citra adalah mengambil citra yang tersimpan di directory tertentu, selanjutnya diterjemahkan menjadi matrik dan dari matrik tersebut baru bisa dilakukan pengolahan citra.



**Gambar 4. Pembacaan citra pada GUI**



Dari cara pembacaan citra dengan matlab dapat menghasilkan matrik dari citra di atas seperti di bawah:

**Gambar 5. Pembacaan citra dengan matlab.**

**b. Perbaikan citra**

Setelah melakukan pembacaan citra, kemudian perlu dilakukan perbaikan citra sesuai keperluan. Perbaikan citra ini dilakukan dengan 4 cara, yaitu brightness, contrast, gamma, dan color. Di bawah ini akan dijelaskan masing-masing langkah yang diambil dalam perbaikan citra (Ridho'i et al., 2022).

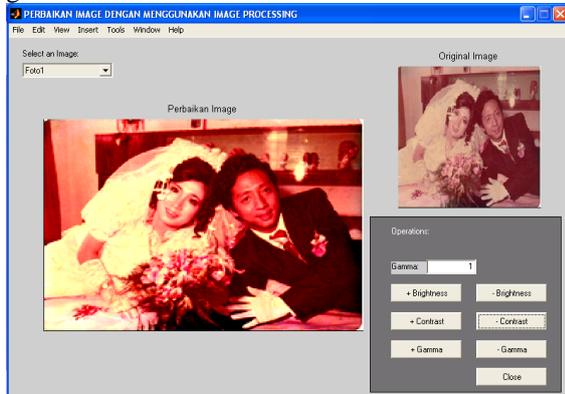
**Brightness**, Proses ini akan memberikan tingkat kecerahan pada suatu citra. Perubahan ini dapat dilihat seperti gambar di bawah ini.



**Gambar 6. Perubahan citra dengan brightness**

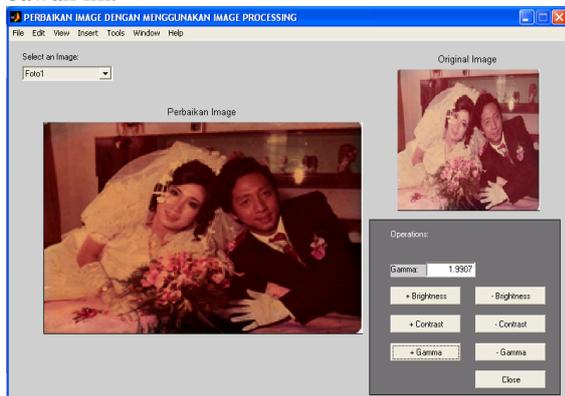
Dengan menaikkan nilai brightness menjadi tinggi grayscale mengarah ke 255 sehingga citra menjadi lebih terang.

**Contrast**, Proses ini mengatur color gray scale pada gambar. Contrast akan bekerja dengan baik ketika warna gambar dipadukan dengan intensitas kecerahan. Perubahan citra ini dapat dilihat seperti gambar di bawah ini.



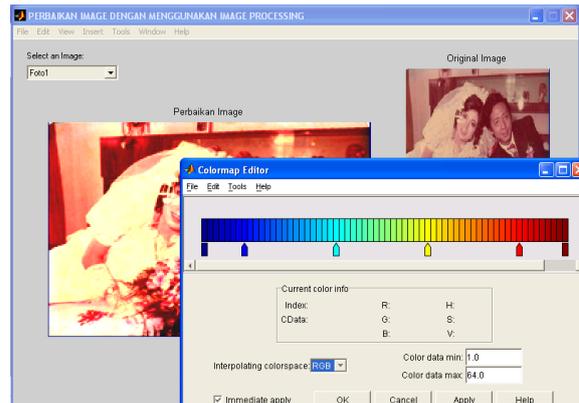
Gambar 7. Perubahan citra melalui contrast

**Gamma**, Proses ini memiliki fungsi yang sama dengan brightness, yaitu mengatur tingkat kecerahan suatu citra. Tetapi ada yang membedakan antara gamma dengan brightness, yaitu gamma makin bertambah ( $\text{gamma} > 1$ ) akan menghasilkan warna makin gelap dan sebaliknya gamma makin berkurang ( $\text{gamma} < 1$ ) akan menghasilkan warna makin terang. Selain itu gamma memiliki nilai yang lebih detail dibandingkan brightness dengan nilai normal adalah 1. Perubahan citra ini dapat dilihat seperti gambar di bawah ini.



Gambar 8. Perubahan citra dengan gamma

**Color**, Proses ini digunakan untuk mengatur elemen warna gambar sesuai yang diinginkan. Proses ini diperlukan untuk memperbaiki elemen warna pada gambar yang mengalami kerusakan. Perubahan citra ini dapat dilihat seperti gambar di bawah ini.

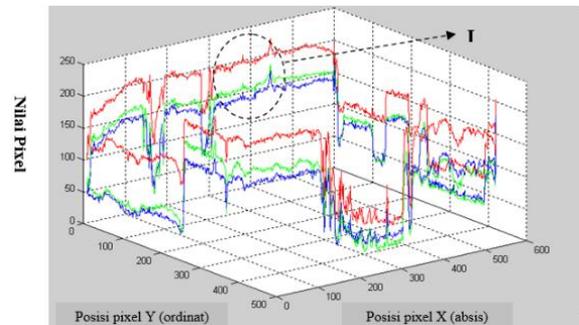


Gambar 9. Perubahan citra dengan color

### c. Perbandingan citra

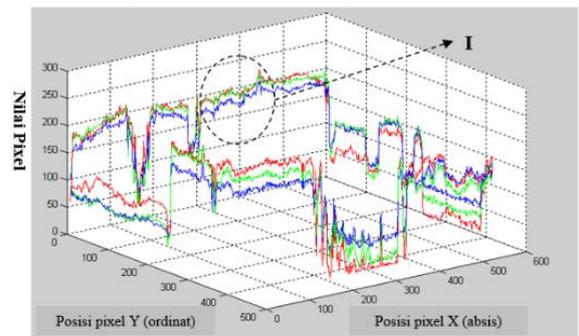
Setelah melakukan perbaikan citra, maka dilakukan perbandingan citra dari citra sebelum perbaikan dengan citra hasil sesudah perbaikan. Di bawah ini dapat dilihat hasil perbandingan citra original dengan citra hasil perbaikan beserta nilainya (muharom syahri, 2021).

**Citra yang mengalami kerusakan warna**, citra yang mengalami kerusakan pada warnanya sebelum dilakukan perbaikan mempunyai nilai pixel seperti dibawah ini.



Gambar 10. Grafik nilai pixel sebelum perbaikan

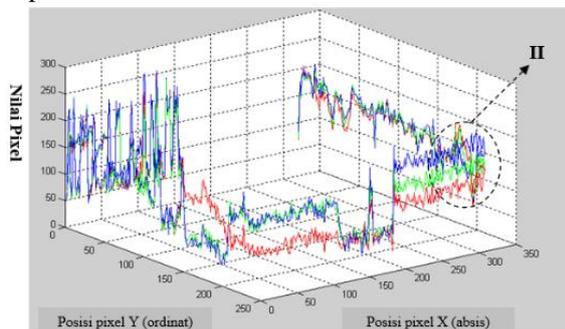
Sedangkan sesudah dilakukan perbaikan citra, maka nilai pixelnya berubah seperti di bawah ini.



Gambar 11. Grafik nilai pixel sesudah perbaikan

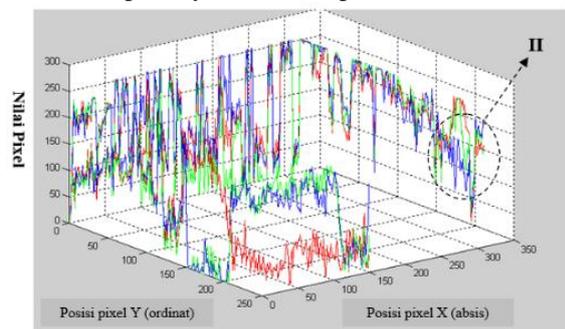
Dari gambar grafik di atas dapat dilihat perubahan nilai pixel sebelum dan sesudah perbaikan. Warna grafik diatas menunjukkan warna RGB, dimana

warna tersebut tersusun dari warna merah (Red), hijau (Green), dan biru (Blue). Citra Yang Terkena Noise, citra yang terkena noise sebelum dilakukan perbaikan mempunyai nilai pixel seperti dibawah ini.



Gambar 12. Grafik nilai pixel sebelum perbaikan

Sedangkan sesudah dilakukan perbaikan citra, maka nilai pixelnya berubah seperti di bawah ini.



Gambar 13. Grafik nilai pixel sesudah perbaikan

Seperti halnya gambar grafik yang telah diterangkan sebelumnya, gambar grafik di atas juga dapat dilihat perubahan nilai pixel sebelum dan sesudah perbaikan. Warna grafik diatas menunjukkan warna RGB, dimana warna tersebut tersusun dari warna merah (Red), hijau (Green), dan biru (Blue).

### 3. KESIMPULAN

Berdasarkan hasil-hasil yang telah dicapai selama perencanaan, pembuatan, dan pengujian perangkat lunak pada pengolahan citra ini, maka dapat disimpulkan sebagai berikut : perangkat lunak ini dapat digunakan untuk menunjang kegiatan pembacaan file gambar dengan menggunakan algoritma kerja yang mirip dengan pikiran manusia pada saat melakukan proses *pembacaan*, yaitu proses membaca gambar 2 dimensi atau informasi tentang gambar 2 dimensi. Kelemahan perangkat lunak ini terletak pada kebutuhannya akan memori kecepatan pengolahan oleh processor.

### PUSTAKA

Blanchet gerard, & Charbit maurice. (2006). *Digital Signal and Image Processing using MATLAB* ®. www.TechnicalBooksPdf.com  
Gonzalez Rafael C, W. R. E. (2002). *Digital Image Processing*.

Jähne, B. (2005). *Bernd Jähne Digital Image Processing*.  
Jain, A. K. (1989). *Fundamentals\_of\_Digital\_Image\_Processing\_-Anil\_K\_Jain*.  
jensen John r. (2015). *introductory digital image processing, a remote sensing perspective*.  
muharom syahri, asnawi saiful, bachri affan. (2021). robot pengikut target berdasarkan bentuk dan warna menggunakan metode hsv untuk aplikasi assistant robot. *JE*, 6, 415–423.  
Pratt, W. K. (2001). *Digital image processing : PIKS inside*. Wiley.  
Ridho'i, A., Setyadjit, K., & Hariadi, B. (2022). *MENENTUKAN KUALITAS BUAH APEL MALANG BERDASARKAN KULITNYA MEMANFAATKAN PENGOLAHAN CITRA DIGITAL* (Vol. 25, Issue 2). <http://univ45sby.ac.id/ejournal/index.php/industri/index>  
Sundararajan, D. (2017). *Digital Image Processing A Signal Processing and Algorithmic Approach*. www.TechnicalBooksPDF.com  
Woods, richard e. (2001). *Digital Image Processing Second Edition Richard E. Woods(1)*.