

# Implementasi Metode Bilateral Filter Perbaikan Kualitas Citra RGB

Nurega Harahap

Program Studi Teknik Informatika, STMIK Budi Darma, Medan, Indonesia

Email: [negahrp@mail.com](mailto:negahrp@mail.com)

**Abstrak**—Dalam proses pengambilan citra menggunakan kamera atau sensor lainnya sering mengalami masalah dan penurunan mutu, seperti hasil capture yang tidak sesuai dengan yang diharapkan. Hal tersebut bisa terjadi karena kemungkinan lensa kamera yang tidak focus, pengaturan pencahayaan yang tidak sempurna, adanya debu atau kotoran lainnya yang menempel di lensa kamera. Memperbaiki kualitas suatu citra sangat diperlukan hingga agar seseorang dapat mengamati citra tersebut dengan jelas dan detail tanpa adanya gangguan. Metode bilateral filter merupakan sebuah filter yang berfungsi menggantikan nilai sebuah piksel dengan hasil perhitungan konvolusi dalam area yang di cakup oleh filter, operasi konvolusi ini sering kali di libatkan dalam operasi ketetanggaan pixel. Yang kemudian menghasilkan aplikasi perangkat lunak yang mampu mengurangi efek noise yang mengganggu pada citra digital dengan metode bilateral filter. Dengan metode ini diharapkan mampu menyelesaikan masalah perbaikan kualitas citra pada aplikasi perbaikan citra. Penelitian ini membahas agar proses yang dilakukan untuk menerapkan perbaikan citra berdasarkan metode bilateral filter, maka dibangun aplikasi dengan menggunakan matlab 2013 sebagai editor untuk mengedit program.

**Kata Kunci:** Citra Digital, Perbaikan Kualitas Citra, Bilateral Filter, Matlab

**Abstract**— In the process of capturing images using a camera or other sensors often experience problems and deterioration in quality, such as capture results that are not as expected. This can happen because of the possibility of a camera lens that is not focused, the lighting settings are not perfect, the presence of dust or other dirt on the camera lens. Improving the quality of an image is necessary so that someone can observe the image clearly and in detail without any disturbance bilateral filter method is a filter that functions to replace The value of a pixel with the convolution calculation results in the area covered by the filter, this convolution operation is often involved in neighboring pixel operations. Which then produces software applications that are able to reduce the effects of disturbing noise on digital images with the bilateral filter method. With this method, it is expected to be able to solve the problem of image quality improvement in image improvement applications. This study discusses that the process undertaken to implement image improvement based on the bilateral filter method, then an application is built using MATLAB 2013 as an editor to edit programs.

**Keywords:** Digital Image, Image Quality Improvement, Bilateral Filter, Matlab

## 1. PENDAHULUAN

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi sari suatu objek. Citra sebagai keluaran suatu sistem perekam data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisive. Citra digital merupakan salah satu bentuk citra yang sangat mudah dipergunakan jika dilihat dari segi pengiriman citra sebagai data, pengolahan dan pemrosesan citra. Disamping itu juga, citra digital menjadi hal yang sangat penting dan berguna dan berbagai hal bidang kehidupan sehingga akses terhadap citra digital ini semakin banyak. Berdasarkan penelitian yang dilakukan oleh Sandy Kosasi mengatakan bahwa tindakan pengambilan (mengabadikan) citra tidak selalu memberikan hasil yang baik (mengalami penurunan mutu)[1]. Citra dapat mengalami derau (*noise*) pada saat pengiriman melalui saluran transmisi sehingga dapat menyebabkan citra terlalu terang/gelap, citra kurang tajam, kabur, mengandung cacat, warnanya terlalu kontras [2]. Oleh karena itu proses perbaikan kualitas citra ini sangat penting untuk menghasilkan kualitas citra yang lebih baik dari kualitas sebelumnya.

Implementasi adalah perluasan aktivitas yang saling menyesuaikan proses interaksi antara tujuan dan tindakan untuk mencapainya serta memerlukan jaringan pelaksanaan, birokrasi yang efektif perbaikan kualitas citra bertujuan untuk mendapatkan tampilan citra dengan bentuk visualisasi yang lebih baik, dengan cara memaksimalkan kandungan informasi di dalam citra masukan.

Citra RGB merupakan model warna adatif, yaitu ketiga berkas cahaya yang ditambahkan bersama-sama, dengan menambahkan panjang gelombang, untuk membuat *spectrum* warna akhir, sehingga dapat disimpulkan bahwa warna dasar dari tiap warna adalah *red*, *green* dan *blue*. Rekonstruksi degradasi citra, dapat dilakukan dengan memanipulasi nilai-nilai RGB pada suatu piksel.

Permasalahan lain yang sering terjadi saat pengambilan citra ialah masalah kontras. Saat pengambilan citra yang tidak focus, pencahayaan yang kurang dan tidak merata sehingga menghasilkan citra kurang baik. Pengambilan citra saat pagi, siang, maupun sore hari juga tentunya berbeda perubahan intensitas cahaya matahari saat pagi, siang, maupun sore hari. *Noise salt and pepper* ini disebabkan oleh gangguan bentuknya berupa bintik-bintik hitam atau putih di dalam citra. Pengolahan citra digital merupakan bagian penting yang mendasari berbagai aplikasi nyata, seperti pengolahan pola,

Adapun solusi untuk penghilangan *noise* atau memperlihatkan citra yang kurang jelas seperti yang diuraikan diatas, maka dilakukan Proses perbaikan kualitas citra agar terlihat lebih baik. Suatu citra juga dapat diperbaiki dengan membuat setiap piksel menjadi beberapa piksel. Perubahan suatu nilai piksel citra melalui perbaikan resolusi sering dibutuhkan untuk keperluan memperlihatkan detail citra agar lebih jelas.

Perbaikan adalah untuk mendapatkan citra yang lebih baik dan lebih berguna saat diterapkan dalam aplikasi tertentu, karena citra asli memiliki kualitas citra yang kurang baik. Adapun metode yang digunakan dalam perbaikan kualitas citra ini dengan metode *bilateral filter*.

Safriadi dalam penelitiannya mengatakan bahwa metode *bilateral filter* dapat digunakan sebagai salah satu metode restorasi citra digital khususnya penapis derau (*noise*) [3]. Metode untuk peningkatan kualitas citra pun sudah mengalami banyak kemajuan, salah satu metode perbaikan kualitas citra yang terbaru adalah *bilateral filter*. Metode ini merupakan salah satu gabungan dari dua metode, yaitu *domain filter* dan *range filter* [4]. Adapun kelebihan dari metode *bilateral filter* ini adalah menghilangkan *noise* pada matriks *spectrogram* sambil menjaga kandungan ciri atau informasi suara[5].

## 2. METODOLOGI PENELITIAN

### 2.1 Citra

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra sebagai keluaran suatu sistem perekaman data dapat bersifat *optic* berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan [6]. Sebuah citra dapat mewakili oleh sebuah matriks yang terdiri dari *M* kolom dan *N* baris, dimana perpotongan anantara kolom dan baris disebut *pixel*, yaitu elemen terkecil dari suatu citra. *Pixel* mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang dapat pada koordinat (x,y) adalah *f(x,y)* yaitu besar *intensitas* atau warna dari *pixel* di titik itu.

### 2.2 Peningkatan Kualitas Citra

Peningkatan kualitas citra adalah suatu proses untuk mengubah sebuah citra menjadi citra baru sesuai dengan kebutuhan melalui cara. Cara-cara bisa dilakukan misalnya dengan fungsi transformasi, operasi matematis, pemfilteran, dan lain-lain[6]. Tujuan utama dari peningkatan kualitas citra adalah untuk memproses citra hingga citra yang dihasilkan lebih baik daripada citra aslinya untuk aplikasi tertentu Peningkatan kualitas citra dibagi dalam dia kategori, yaitu metode domain spasial (ruang atau waktu) dan metode domain frekuensi. Teknik pemrosesan metode domain spasial adalah berdasarkan manipulasi langsung dari piksel di dalam citra. Sedangkan teknik pemrosesan metode domain frekuensi adalah berdasarkan perubahan transformasi fourier pada citra. Perbaikan kualitas citra yang dijadikan objek mempunyai kualitas yang buruk, misalnya citra mengalami derau(*noise*), citra terllu gelap/terang, citra kurang tajam kabur, dan sebagainya.

### 2.3 Bilateral Filter

Metode *bilateral filter* merupakan salah satu metode perbaikan kualitas citra yang bersifat *non-linier* dan *adaftif* yang di perkenalkan oleh Tomasi. *bilateral filter* disebut *non-linier* karena fungsi-fungsi yang digunakan dalam metode ini bukan fungsi linier. Sedangkan di sebut *adaftif* karena metode ini tidak tergantung pada tipe derau (*noise*) yang ada[4]. *Bilateral filter* ini merupakan pengembangan dan konsep penghalusan *Gaussian* dengan menitik beratkan *koefisien filter* dengan hubungannya kepada *intensitas relative piksel*. Piksel akan berbeda dalam hal *intensitas* sentralnya walaupun terlihat mirip dengan tetangganya. *Filter* ini merupakan *konvolusi efektif* dengan *filter Gaussian non-linier* yang menekankan pada *intensitas* piksel. Langkah-langkah untuk memperbaiki kualitas citra berdasarkan metode *bilateral filter* [4] ini adalah :

1. Pembobotan dilakukan dengan mencari jarak spasial dan radiometrik dari tiap piksel pada citra *g* yang bertetangga dengan piksel yang menjadi pusat analisis *g(x,y)*.
2. Nilai-nilainya digabungkan menjadi satu nilai bobot (kernel)
3. Setelah itu, dapat di cari nilai piksel hasil *f(x,y)* dengan menggunakan rumus
4. Kernel akan berjalan terus sampai semua piksel pada citra yang terdegradasi dilalui.

Dalam penelitian ini, penerapan rumus pembobotan pada *bilateral filter* menggunakan rumus fungsi *Gaussian* , karena fungsi *Gaussian* merupakan kasus yang sering dijumpai dan mudah diimplementasikan[4]. Adapun rumus-rumusnya yang digunakan dalam metode *bilateral filter* ini adalah :

$$r(a_i) = e^{\frac{[f(a_1) - F(a_0)]}{2\sigma^2}} \dots \dots \dots (1)$$

Keterangan :

- R(ai) = Elemen matriks kernel
- e = Konstanta
- A = Indeks tengah dari matriks kernel

$$g(x,y) = e^{\frac{x^2+y^2}{2\sigma^2}} \dots \dots \dots (2)$$

Keterangan :

- G (x,y) = Kernel Gaussian
- E dan  $\sigma$  = Konstanta
- X dan Y = Index tengah dari matriks

Dalam hal ini  $\sigma$  adalah deviasi standart piksel pada pusat (x,y) mendapatkan bobot terbesar berupa 1. *Gaussian filter* paling tidak berukuran 5x5 sebagai contoh, bobot-bobotnya dapat di peroleh dengan membuat  $\sigma^2$  bernilai 1.

$$H(a_0) = k^{-1} \sum_{i=0}^{n-1} f(a_i) \times g(a_i) \times r(a_i) \dots \dots \dots (3)$$

Keterangan :

- $H(a_0)$  = bobot hasil perkalian pada posisi ( $a_0$ )
- $N$  = Jumlah kolom matriks kernel
- $K$  = Penjumlahan semua bobot di  $F$
- $F(a,i)$  = elemen matriks kernel *gaussian* pada posisi (a,i)

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Analisa Masalah

Analisa berfungsi sebagai kerangka yang digunakan untuk mereduksi gambar yang terkena *noise salt and pepper* pada citra RGB yang akan diproses. Analisa merupakan kegiatan untuk memperhatikan, mengamati sesuatu yang dilakukan dengan maksud untuk mengidentifikasi dan mengevaluasi permasalahan-permasalahan yang diharapkan dapat diusulkan sebagai perbaikan.

Gambar yang dianalisa adalah gambar dari citra yang berwarna yang dimana gambar tersebut mengalami derau (*noise*). Citra dapat mengalami derau (*noise*) pada saat pengiriman melalui saluran transmisi sehingga dapat menyebabkan citra terlalu terang/gelap, citra kurang tajam, kabur, mengandung cacat, warnanya terlalu kontras, kemudian proses perbaikan kualitas citra ini sangat penting untuk menghasilkan kualitas citra yang lebih baik dari kualitas sebelumnya. Tujuan dari hasil penelitian ini berupa sebuah perangkat lunak yang bermanfaat untuk pengurangan *noise* pada citra RGB, untuk melakukan perbaikan citra yang terkena *noise* dengan menggunakan metode *bilateral filter* dalam pengolahan citra.

Perbaikan citra RGB dapat diperbaiki dengan salah satu metode perbaikan citra yaitu *bilateral filter* di mana hasil perbaikan citra dilakukan di dalam obyek JPEG. Proses *bilateral filter* adalah suatu proses untuk menentukan kernel pada posisi antara suatu sampel dengan sampel tetangganya.

Proses awal perbaikan kualitas citra dengan metode *bilateral filter* ini adalah melakukan proses penyeleksian terhadap gambar RGB yang dijadikan sebagai sampel. Setelah hasil penyeleksian hasil gambar RGB di dapat proses selanjutnya membaca resolusi citra masukan yang berupa JPG objek yang berguna untuk mengetahui ukuran/resolusi citra tersebut. Nilai tiap *pixel* pada gambar dapat diketahui dari properties *pixel*. Metode *bilateral filter* adalah salah satu metode perbaikan kualitas citra. Tahap-tahap yang dilakukan untuk memperbaiki citra berdasarkan metode *bilateral filter* ini adalah salah satu *filter* yang bekerja dengan cara menghitung kernel dan menghitung konvolusi dalam area yang dicakup oleh *filter*. Di dalam melakukan operasi perbaikan *noise* citra dengan metode *bilateral filter* operasi ini dilakukan dengan cara konvolusi, konvolusi sering kali dilibatkan dalam oprasi ketetanggan *pixel*. Dalam hal ini peneliti memakai persamaan formulasi dalam metode *bilateral filter* ini dengan tiga tahap untuk menyelesaikannya

#### 3.2 Penerapan Algoritma Camellia

Berikut adalah citra RGB yang terkena *noise*, gambar tersebut diambil dari kamera contoh kasus yang dibahas adalah *noise* yang terjadi pada citra RGB, *noise* tersebut ada karena terjadi gangguan pada kamera pada saat proses pengambilan gambar.

Gambar di bawah ini merupakan gambar berukuran 178 x162 *pixels* dan di ubah menjadi berukuran 5 x 5 *pixel* yang diambil pada gambar ber *noise*. karena citra yang digunakan adalah citra berwarna RGB maka akan terdapat 3 matrix citra untuk tiap-tiap komponen yaitu *Red, Green, dan Blue*.



Citra Bernoise 178 x 162 Piksel



Resolusi 5x5 Piksel

**Gambar 1.** Pengambilan *Pixel* yang Akan Diproses

Tabel 1. Nilai Citra RGB

Nilai Citra Red, Green, Blue (RGB)				
R = 33	R = 25	R = 31	R = 15	R = 30
G = 0	G = 1	G = 24	G = 19	G = 34
B = 15	B = 14	B = 31	B = 22	B = 37
R = 43	R = 35	R = 31	R = 0	R = 16
G = 6	G = 9	G = 21	G = 1	G = 17
B = 23	B = 22	B = 29	B = 5	B = 21
R = 47	R = 63	R = 57	R = 0	R = 14
G = 8	G = 35	G = 46	G = 0	G = 13
B = 26	B = 49	B = 54	B = 4	B = 18
R = 67	R = 64	R = 63	R = 32	R = 48
G = 26	G = 33	G = 50	G = 30	G = 46
B = 44	B = 48	B = 59	B = 35	B = 51
R = 64	R = 23	R = 22	R = 29	R = 45
G = 23	G = 0	G = 9	G = 27	G = 43
B = 41	B = 7	B = 18	B = 32	B = 48

Berdasarkan piksel di atas diketahui nilai piksel citra tersebut diambil menggunakan aplikasi bantuan *matlab*, dimana nilai piksel *red* = nilai piksel *green* = nilai piksel *blue*. Nilai-nilai piksel tersebut akan diproses dengan menerapkan metode *Bilateral filter* untuk mengurangi *noise* pada citra tersebut. Nilai piksel dari citra di atas diproses sesuai dengan ketentuan dari metode *Bilateral filter*.

Langkah 1

Untuk memudahkan semua perhitungan nilai R,G,B ditambah kan dulu semua nilai ascii nya, adapun perhitungannya sebagai berikut :

$$R + G + B$$

- $\frac{33+0+15}{3} = \frac{48}{3} = 16$
- $\frac{25+1+14}{3} = \frac{40}{3} = 13$
- $\frac{31+24+31}{3} = \frac{86}{3} = 29$
- $\frac{15+19+31}{3} = \frac{65}{3} = 22$
- $\frac{30+34+37}{3} = \frac{107}{3} = 34$
- $\frac{43+6+23}{3} = \frac{72}{3} = 24$
- $\frac{35+9+22}{3} = \frac{66}{3} = 22$
- $\frac{31+21+29}{3} = \frac{81}{3} = 27$
- $\frac{0+1+22}{3} = \frac{23}{3} = 8$
- $\frac{16+17+21}{3} = \frac{54}{3} = 18$
- $\frac{47+8+26}{3} = \frac{81}{3} = 27$
- $\frac{63+33+49}{3} = \frac{145}{3} = 48$
- $\frac{57+46+54}{3} = \frac{157}{3} = 52$
- $\frac{0+0+4}{3} = \frac{4}{3} = 1$
- $\frac{14+13+18}{3} = \frac{45}{3} = 15$
- $\frac{67+26+44}{3} = \frac{137}{3} = 46$
- $\frac{64+33+48}{3} = \frac{145}{3} = 48$
- $\frac{63+50+59}{3} = \frac{172}{3} = 57$
- $\frac{32+30+35}{3} = \frac{97}{3} = 32$
- $\frac{48+46+51}{3} = \frac{145}{3} = 48$
- $\frac{64+0+15}{3} = \frac{79}{3} = 26$
- $\frac{23+0+7}{3} = \frac{30}{3} = 10$
- $\frac{22+9+18}{3} = \frac{49}{3} = 16$
- $\frac{29+27+32}{3} = \frac{88}{3} = 29$
- $\frac{45+43+48}{3} = \frac{136}{3} = 45$

Tabel 2. Perhitungan Nilai RGB

16	13	29	22	34
24	22	27	8	18
27	48	52	1	15
46	48	57	32	48
26	10	16	29	45

$$R(ai) = e \frac{[f(a_1) - f(a_0)]}{2\sigma^2}$$

- $\frac{16-13}{2.1^2} = \frac{3}{2} = 1,5$   
 $2,718 \times 1,5 = 4,4816$
- $\frac{13-29}{2.1^2} = \frac{-16}{2} = -8$   
 $2,718 \times -8 = 0,0003$
- $\frac{29-22}{2.1^2} = \frac{7}{2} = 3,5$   
 $2,718 \times 3,5 = 33,115$
- $\frac{22-34}{2.1^2} = \frac{-12}{2} = -6$   
 $2,718 \times -6 = 0,0024$

5.  $\frac{34-16}{2.1^2} = \frac{18}{2} = 9$   
 $2,718 \times 9 = 8,103$
6.  $\frac{24-22}{2.1^2} = \frac{2}{2} = 1$   
 $2,718 \times 1 = 2,7182$
7.  $\frac{22-27}{2.1^2} = \frac{-5}{2} = -2,5$   
 $2,718 \times -2,5 = 0,0820$
8.  $\frac{27-8}{2.1^2} = \frac{19}{2} = 9,5$   
 $2,718 \times 9,5 = 13,359$
9.  $\frac{8-18}{2.1^2} = \frac{-10}{2} = -5$   
 $2,718 \times -5 = 0,0067$
10.  $\frac{18-24}{2.1^2} = \frac{-6}{2} = -3$   
 $2,718 \times -3 = 0,0497$
11.  $\frac{27-48}{2.1^2} = \frac{-21}{2} = -10,5$   
 $2,718 \times -10,5 = 0,00027$
12.  $\frac{48-52}{2.1^2} = \frac{-4}{2} = -2$   
 $2,718 \times -2 = 0,1353$
13.  $\frac{52-1}{2.1^2} = \frac{51}{2} = 25,5$   
 $2,718 \times 25,5 = 118,716$
14.  $\frac{1-15}{2.1^2} = \frac{-14}{2} = -7$   
 $2,718 \times -7 = 1,12535$
15.  $\frac{15-27}{2.1^2} = \frac{-12}{2} = -6$   
 $2,718 \times -6 = 0,0024$
16.  $\frac{46-48}{2.1^2} = \frac{-2}{2} = -1$   
 $2,718 \times -1 = 2,718$
17.  $\frac{48-57}{2.1^2} = \frac{-9}{2} = -4,5$   
 $2,718 \times -4,5 = 0,01110$
18.  $\frac{57-32}{2.1^2} = \frac{25}{2} = 12,5$   
 $2,718 \times 12,5 = 268,337$
19.  $\frac{32-48}{2.1^2} = \frac{-16}{2} = -8$   
 $2,718 \times -8 = 1,5229$
20.  $\frac{48-46}{2.1^2} = \frac{2}{2} = 1$   
 $2,718 \times 1 = 2,718$
21.  $\frac{26-10}{2.1^2} = \frac{16}{2} = 8$   
 $2,718 \times 8 = 2,980$
22.  $\frac{10-16}{2.1^2} = \frac{-6}{2} = -3$   
 $2,718 \times -3 = 0,04978$
23.  $\frac{16-29}{2.1^2} = \frac{-13}{2} = -6,5$   
 $2,718 \times -6,5 = 0,00150$
24.  $\frac{29-45}{2.1^2} = \frac{-16}{2} = -8$   
 $2,718 \times -8 = 2,980$
25.  $\frac{45-26}{2.1^2} = \frac{19}{2} = 9,5$   
 $2,718 \times 9,5 = 13,359$

Tabel 3. Hasil Akhir Perhitungan RGB

4,48116	0,0003	33,115	0,0024	8,103
2,7182	0,0820	13,359	0,0067	0,0497
0,00027	0,1353	118,716	1,12535	0,0024
2,718	0,01110	268,337	1,5229	2,718
2,980	0,04978	0,00150	2,980	13,359

## Langkah 2

Dilakukan perhitungan untuk mencari kernel nya, kernel atau *mask* memberikan petunjuk tentang apa yang harus dilakukan filter terhadap data. adapun cara perhitungannya sebagai berikut :

1. Perhitungan pencarian nilai kernel pertama  

$$g(1,1) = e^{\frac{1^2 + 1^2}{2 \cdot \sigma^2}}$$

$$g(1,1) = 2,718 \frac{1 + 1}{2 \cdot 1^2}$$

$$g(1,1) = 2,718 \frac{2}{2}$$

$$2,718 \times 1 = 1$$
 Hasil yang didapat adalah = 1
2. Perhitungan pencarian nilai kernel kedua  

$$g(1,2) = e^{\frac{1^2 + 2^2}{2 \cdot \sigma^2}}$$

$$g(1,2) = 2,718 \frac{1 + 4}{2 \cdot 1^2}$$

$$g(1,2) = 2,718 \frac{5}{2}$$

$$2,718 \times 3 = 20,0855$$
 Hasil yang didapat adalah = 20,0855
3. Perhitungan pencarian nilai kernel ketiga  

$$g(1,3) = e^{\frac{1^2 + 3^2}{2 \cdot \sigma^2}}$$

$$g(1,3) = 2,718 \frac{1 + 9}{2 \cdot 1^2}$$

$$g(1,3) = 2,718 \frac{10}{2}$$

$$2,718 \times 5 = 13,59$$
 Hasil yang didapat adalah = 13,59
4. Perhitungan pencarian nilai kernel keempat  

$$g(2,1) = e^{\frac{2^2 + 1^2}{2 \cdot \sigma^2}}$$

$$g(2,1) = 2,718 \frac{4 + 1}{2 \cdot 1^2}$$

$$g(2,1) = 2,718 \frac{5}{2}$$

$$2,718 \times 3 = 20,0855$$
 Hasil yang didapat adalah = 20,0855
5. Perhitungan pencarian nilai kernel keempat  

$$g(2,2) = e^{\frac{2^2 + 2^2}{2 \cdot \sigma^2}}$$

$$g(2,2) = 2,718 \frac{4 + 4}{2 \cdot 1^2}$$

$$g(2,2) = 2,718 \frac{8}{2}$$

$$2,718 \times 4 = 10,872$$
 Hasil yang didapat adalah = 10,872

6. Perhitungan pencarian nilai kernel keenam

$$g(8,1) = e^{\frac{2^2 + 3^2}{2 \cdot \sigma^2}}$$

$$g(2,3) = 2,718 \frac{4 + 9}{2 \cdot 1^2}$$

$$g(2,3) = 2,718 \frac{13}{2}$$

$$2,718 \times 11 = 59,874$$

Hasil yang didapat adalah = 59,874

7. Perhitungan pencarian nilai kernel ketujuh

$$g(3,1) = e^{\frac{3^2 + 1^2}{2 \cdot \sigma^2}}$$

$$g(3,1) = 2,718 \frac{9 + 1}{2 \cdot 1^2}$$

$$g(3,1) = 2,718 \frac{10}{2}$$

$$2,718 \times 8 = 2,980$$

Hasil yang didapat adalah = 2,980

8. Perhitungan pencarian nilai kernel kedelapan

$$g(3,2) = e^{\frac{3^2 + 2^2}{2 \cdot \sigma^2}}$$

$$g(3,2) = 2,718 \frac{9 + 4}{2 \cdot 1^2}$$

$$g(3,2) = 2,718 \frac{13}{2}$$

$$2,718 \times 11 = 59,874$$

Hasil yang didapat adalah = 59,874

9. Perhitungan pencarian nilai kernel kesembilan

$$g(3,3) = e^{\frac{3^2 + 3^2}{2 \cdot \sigma^2}}$$

$$g(3,3) = 2,718 \frac{9 + 9}{2 \cdot 1^2}$$

$$g(3,3) = 2,718 \frac{18}{2}$$

$$2,718 \times 16 = 8,886$$

Hasil yang didapat adalah = 8,886

**Tabel 4.** Hasil Perhitungan Kernel Citra

1	20,0855	2,980
20,0855	403,428	59,874
2,980	59,874	8,886

**Langkah 3**

Menghitung konvolusi, konvolusi didefinisikan secara sederhana sebagai perhitungan kernel digeser sepanjang baris dan kolom dalam citra sehingga diperoleh nilai baru pada citra keluaran. Berikut operasi konvolusi antara citra dengan kernel dapat diilustrasikan sebagai berikut :

asci (:, : , 1)				
33	25	31	15	30
43	35	31	0	16
47	63	57	0	14
67	64	63	32	48
64	23	22	29	45
asci (:, : , 2)				
0	1	24	19	34
6	9	21	1	17
8	35	46	0	13
26	33	50	30	46
23	0	9	27	43
asci (:, : , 3)				
15	14	31	22	37
23	22	29	5	21
26	49	54	4	18
44	48	59	35	51
41	7	18	32	48

**Tabel 5.** Kernel Citra 3x3

1	20,085	2,980
20,0855	403,428	59,874
2,980	59,874	8,886

**Langkah 4**

Perhitungan Matrix Komponen Red, Tempatkan kernel pada sudut kiri atas, kemudian hitung nilai pixel pada posisi (209) dari kernel, berikut prosesnya.

33	25	31	15	30
43	35	31	0	16

0	0	0	0	0
0	36			0

47	63	57	0	14
67	64	63	32	48
64	23	22	29	45

0				0
0				0
0	0	0	0	0

Hasil konvolusi = 43 Nilai ini di hitung dengan cara berikut :

$$G(1,1) = (33 \times 1) + (25 \times 20,0855) + (31 \times 2,980) + (15 \times 20,0855) + (35 \times 403,428) + (31 \times 59,874) + (47 \times 2,980) + (63 \times 59,874) + (57 \times 8,886) = 33 + 1,2445 + 0,01023 + 2,14054 + 0,08645 + 0,000496 + 0,01551 + 0,001008 + = 36,498734$$

Geser kernel satu *pixel* ke kanan, kemudian hitung nilai *pixel* pada posisi (185) dari kernel.

33	25	31	15	30
43	35	31	0	16
47	63	57	0	14
67	64	63	32	48
64	23	22	29	45

0	0	0	0	0
0	36	28		0
0				0
0				0
0	0	0	0	0

Hasil konvolusi = 28 Nilai ini di hitung dengan cara berikut :

$$G(1,2) = (25 \times 1) + (31 \times 20,0855) + (15 \times 2,980) + (35 \times 20,0855) + (31 \times 403,428) + (0 \times 59,874) + (63 \times 2,980) + (57 \times 0,000016) + (0 \times 8,886) = 25 + 1,54318 + 0,00495 + 1,7423 + 0,07657 + 0 + 0,02079 + 0,000912 + 0 = 28,388702$$

Dengan cara yang sama seperti di atas, maka nilai *pixel* elemen *red* dari citra awal setelah dikonvolusi adalah sebagai berikut :

**Tabel 6.** Hasil Perkalian Matriks Red

0	0	0	0	0
0	36	28	43	0
0	47	75	87	0
0	78	101	110	0
0	0	0	0	0

Perhitungan Matrix Komponen *Green*, Tempatkan kernel pada sudut kiri atas, kemudian hitung nilai *pixel* pada posisi (120) dari kernel, berikut prosesnya.

0	1	24	19	34
6	9	21	1	17
8	35	46	0	13
26	33	50	30	46
23	0	9	27	43

0	0	0	0	0
0	21			0
0				0
0				0
0	0	0	0	0

Hasil konvolusi = 21 Nilai ini di hitung dengan cara berikut :

$$G(1,1) = (0 \times 1) + (1 \times 20,0855) + (24 \times 2,980) + (6 \times 20,0855) + (9 \times 403,428) + (21 \times 59,874) + (8 \times 2,980) + (35 \times 59,874) + (46 \times 8,886) = 0 + 0,04978 + 0,00792 + 0,29868 + 0,02223 + 0,000336 + 0,00264 + 0,00056 + 51,7638 = 21,145946$$

Geser kernel satu *pixel* ke kanan, kemudian hitung nilai *pixel* pada posisi (100) dari kernel

0	1	24	19	34
6	9	21	1	17
8	35	46	0	13
26	33	50	30	46
23	0	9	27	43

0	0	0	0	0
0	52	42		0
0				0
0				0
0	0	0	0	0

Hasil konvolusi = 42 Nilai ini di hitung dengan cara berikut :

$$G(1,2) = (1 \times 1) + (24 \times 20,0855) + (19 \times 2,980) + (9 \times 20,0855) + (21 \times 403,428) + (1 \times 59,874) + (35 \times 2,980) + (46 \times 59,874) + (0 \times 8,886) = 1 + 1,19472 + 0,00627 + 0,44802 + 0,05187 + 0,000016 + 0,01155 + 0,000736 + 0 = 41,9427$$

Dengan cara yang sama seperti di atas, maka nilai *pixel* elemen *green* dari citra awal setelah dikonvolusi adalah sebagai berikut :

**Tabel 7.** Hasil Perkalian Matriks Green

0	0	0	0	0
0	21	42	17	0
0	46	72	99	0
0	53	73	66	0
0	0	0	0	0

Perhitungan Matrix Komponen *Blue*, Tempatkan kernel pada sudut kiri atas, kemudian hitung nilai *pixel* pada posisi (112) dari kernel, berikut prosesnya.

15	14	31	22	37
23	22	29	5	21
26	49	54	4	18
44	48	59	35	51
41	7	18	32	48

0	0	0	0	0
0	25			0
0				0
0				0
0	0	0	0	0

Hasil konvolusi = 25 Nilai ini di hitung dengan cara berikut :

$$G(1,1) = (15 \times 1) + (14 \times 20,0855) + (31 \times 2,980) + (23 \times 20,0855) + (22 \times 403,428) + (29 \times 59,874) + (26 \times 2,980) + (49 \times 59,874) + (54 \times 8,886) = 15 + 452 + 1,242 + 442 + 824 + 1,694 + 1,176 + 783 + 2,153 = 24,72243$$

Geser kernel satu *pixel* ke kanan, kemudian hitung nilai *pixel* pada posisi (93) dari kernel

15	14	31	22	37
23	22	29	5	21
26	49	54	4	18
44	48	59	35	51
41	7	18	32	48

0	0	0	0	0
0	25	54		0
0				0
0				0
0	0	0	0	0

Dengan cara yang sama seperti di atas, maka nilai *pixel* elemen *blue* dari citra awal setelah dikonvolusi adalah sebagai berikut :

Tabel 8. Hasil Perkalian Matriks Blue

0	0	0	0	0
0	25	54	19	0
0	65	65	22	0
0	63	95	75	0
0	0	0	0	0

Tabel 9. Nilai Hasil Citra *Red, Green, Blue* (RGB)

Nilai Hasil Citra <i>Red, Green, Blue</i> (RGB)		
R = 36	R = 28	R = 43
G = 21	G = 42	G = 17
B = 25	B = 54	B = 99
R = 47	R = 75	R = 87
G = 46	G = 72	G = 99
B = 65	B = 65	B = 22
R = 78	R = 101	R = 110
G = 53	G = 73	G = 66
B = 63	B = 95	B = 75

Setelah dilakukan proses metode *bilateral filter* dengan menggunakan kernel 3x3 akan menjadi hasil citra yang baru. Berdasarkan pengujian yang dilakukan metode *bilateral filter* dapat memperbaiki kualitas citra dengan memberikan filter kepada komponen citra tersebut.

#### 4. KESIMPULAN

Dari hasil penulisan dan analisa dari bab-bab sebelumnya, maka dapat di ambil kesimpulan, dimana kesimpulan-kesimpulan tersebut kiranya dapat berguna bagi para pembaca, sehingga penulis skripsi ini dapat lebih bermanfaat. Adapun kesimpulan-kesimpulan tersebut adalah:

1. Proses perbaikan kualitas citra RGB dilakukan dengan metode *bilateral filter* untuk memperbaiki kualitas citra terhadap kekurangan pada citra RGB. Proses perbaikan kualitas citra dilakukan dengan cara melakukan terhadap proses perbaikan citra hasil RGB dengan cara melakukan proses perbaikan citra.
2. Metode *bilateral filter* dapat diterapkan dalam perbaikan kualitas RGB dengan cara penghalusan citra. Aplikasi perbaikan kualitas citra RGB telah dapat meningkatkan resolusi spasial citra sehingga dapat memudahkan pengguna melakukan perbaikan citra untuk meningkatkan resolusi spasialnya.
3. Aplikasi perbaikan kualitas citra RGB telah selesai dirancang dengan menggunakan *matlab 2013* dan dapat dijalankan pada sistem operasi *windows*.

#### REFERENCES

[1] M. Informatika and S. Kosasi, "Rekonstruksi Degradasi dalam Pengolahan Rekonstruksi Degradasi dalam Pengolahan

- Citra Menggunakan Metode Konvolusi,” no. January 2018, 2015.
- [2] P. Metode and R. Untuk, “Penerapan metode retinex untuk meningkatkan kecerahan citra pada hasil screenshot,” vol. 1, no. 1, pp. 24–27, 2017.
  - [3] A. Essra, “ANALISIS DETEKSI TEPI CANNY PADA CITRA DENGAN GAUSSIAN FILTERING DAN BILATERAL,” vol. 2, no. 1, pp. 34–39.
  - [4] C. Tomasi, “Bilateral Filtering for Gray and Color Images,” no. Maret 1998.
  - [5] “Pengenalan Suara Paru Manusia Menggunakan Spektrogram Bilateral Filter 2D,” pp. 1–10, 2010
  - [6] M. T. Sutoyo, S. si, End Mulyanto, S. Si, *Teori Pengolahan Citra Digital*, Andi. Yogyakarta, 2009.
  - [7] D. Putra, *Pegolahan Citra Digital*, Andi. Yogyakarta, 2010.
  - [8] T. Fourier *et al.*, “METODE TRANSFORMASI FOURIER DAN FAST FOURIER TRANSFORM,” vol. 1, no. 3, pp. 94–99, 2016.
  - [9] S. Y. Riska, “Klasifikasi Level Kematangan Tomat Berdasarkan Perbedaan Perbaikan Citra Menggunakan Rata-Rata RGB Dan Index Pixel,” vol. 9, no. 2, pp. 18–26, 2015.
  - [10] S. Kurniawati, “PERANCANGAN APLIKASI REDUKSI NOISE PADA CITRA DIGITAL,” vol. 1, no. 3, pp. 122–129, 2016.
  - [11] M. Irwansyah, “IMPLEMENTASI ORDER-STATISTIC FILTER UNTUK MEREDUKSI NOISE PADA CITRA DIGITAL,” pp. 40–43, 2017.
  - [12] F. Rusdi *et al.*, “Point Operation,” vol. 2016, pp. 54–59, 2016.
  - [13] A. Kadir, *Pengenalan Algoritma*, Andi. Yogyakarta, 2013.
  - [14] M. S. Rosa A. S., *Rekayasa Perangkat Lunak*. Bandung, 2014.
  - [15] E. Prasetyo, *Pengolahan citra digital dan aplikasinya menggunakan matlab*, Andi. Yogyakarta, 2011.