

RANCANG BANGUN APLIKASI *FILTER NEIGHBORHOOD PROCESSING* DAN *NOISE REDUCTION* UNTUK *IMAGE PROCESSING* MENGGUNAKAN MATLAB

Ratih Novita Sari¹, Nurfitri Anbarsanti²

¹Akademi Teknik Telekomunikasi Sandhy Putra Jakarta Jakarta, Indonesia
Email: novitasariratih@gmail.com; anbarsanti@yahoo.com

Abstrak

Perbaikan kualitas citra merupakan salah satu proses awal dalam pengolahan citra (*image processing*). Perbaikan kualitas citra diperlukan karena sering kali citra yang dijadikan objek pembahasan mempunyai kualitas yang buruk, misalnya citra mengalami derau (*noise*) pada saat pengiriman melalui saluran transmisi, citra yang gelap, citra yang kurang tajam, kabur dan sebagainya. Penelitian ini mempelajari teknik-teknik pengurangan *noise* dengan menggunakan filter *neighborhood processing*. Pengembangan *filter* untuk mengurangi *noise* selanjutnya dilakukan, dan diuji coba terhadap beberapa citra. Citra yang diuji diberikan *noise* Gaussian, *Salt and Pepper*, dan *Speckle* kemudian dianalisa performanya secara kualitatif dengan membandingkan citra asli, citra bernoise, dan citra *output filter* secara kasat mata. Secara kuantitatif diukur melalui nilai MSE, SNR, dan PSNR-nya. Hasil pengujian dapat disimpulkan dengan karakteristik *noise* yang dapat dikurangi secara signifikan dengan menggunakan *filter* LPF dan HPF. *Filter* LPF (*filter* Gaussian dan *filter* Average) dapat membuat gambar menjadi lebih halus dibandingkan dengan HPF (*filter* Laplacian, *filter* Prewitt, dan *filter* Sobel) yang mempertajam hasil gambar.

Kata Kunci: *Neighborhood Filter, Noise Reduction*

Abstract

Improved image quality is one of the early process in image processing (*image processing*). Improved image quality is required because often the image used as the subject of discussion has poor quality, for example, image is noise during transmission through transmission line, dark image, image less sharp, blur and so on. This study studied noise reduction techniques by using filter neighborhood processing. The development of filters to reduce noise is further done, and tested against several images. The tested image was given Gaussian, Salt and Pepper, and Speckle noise and then analyzed its performance qualitatively by comparing original image, image bernoise, and filter output image by naked eye. Quantitatively measured through the value of MSE, SNR, and PSNR it. Test results can be concluded with noise characteristics that can be significantly reduced by using LPF and HPF filters. LPF filters (Gaussian filter and filter Average) can make images smoother than HPF (Laplacian filter, Prewitt filter, and Sobel filter) that sharpens the image.

Keywords: *Neighborhood Filter, Noise Reduction*

PENDAHULUAN

Peningkatan mutu citra adalah suatu proses mendapatkan citra yang lebih mudah diinterpretasikan oleh mata manusia. Proses pengolahan citra yang termasuk dalam kategori peningkatan mutu citra terdiri dari proses-proses yang bertujuan memperbaiki mutu citra untuk memperoleh keindahan gambar, untuk kepentingan analisis citra, dan untuk

mengoreksi citra. Teknik yang digunakan, yaitu antara lain *image processing*, berupa proses perbaikan citra dengan meningkatkan kualitas citra baik kontras maupun kecerahan. *image restoration*, proses memperbaiki model citra, *color image*, suatu proses yang melibatkan citra berwarna, baik berupa *image enhancement*, *image restoration* atau yang lainnya.

Dengan permasalahan tersebut, maka penulis membuat penelitian ini dengan judul “Rancang Bangun Aplikasi Filter Neighborhood Processing dan Noise Reduction untuk Image Processing Menggunakan MATLAB“. Dengan adanya aplikasi tersebut diharapkan dapat membantu menghilangkan *noise* dan memperbaiki kualitas citra pada gambar.

Maksud dan tujuan dari penulisan proyek akhir ini adalah membuat suatu aplikasi yang dapat membantu dalam peningkatan mutu citra dengan baik dan mudah untuk citra dengan gambar warna (RGB) atau hitam-putih (*grey scale*).

Permasalahan yang akan dibahas di dalam proyek akhir ini, antara lain: apa itu *image processing*, bagaimana cara memperbaiki gambar yang terkena *noise*, *noise* apa saja yang terjadi pada gambar, bagaimana cara untuk memfilter gambar, dan *filter* apa saja yang digunakan untuk proses perbaikan gambar.

Dalam proyek akhir ini terdapat batasan-batasan masalah, antara lain: simulasi dilakukan dengan menggunakan MATLAB, hanya membahas gambar tertentu untuk di-filter, hanya menggunakan *noise salt & pepper*, *gaussian*, *speckle*. *filter* yang digunakan yaitu *filter LPF* dan *HPF*.

TINJAUAN PUSTAKA

Citra

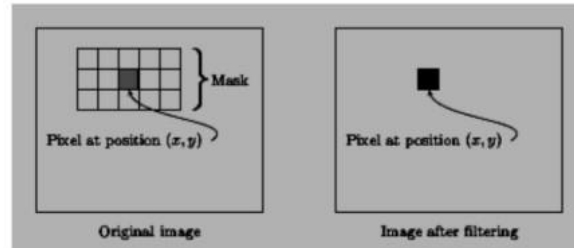
Citra adalah suatu representasi (gambaran), kemiripan atau imitasi dari suatu objek. Citra terbagi dua yaitu ada citra bersifat analog dan ada citra yang bersifat digital. Sebuah citra dapat mewakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut pixel, yaitu elemen terkecil dari suatu citra. Pixel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna (Iqbal, 2017).

Image Processing

Image atau gambar adalah representasi spasial dari suatu objek yang sebenarnya dalam bidang dua dimensi yang biasanya ditulis dalam koordinat kartesian x-y, dan setiap koordinat merepresentasikan satu sinyal terkecil dari objek yang biasanya koordinat terkecil ini disebut sebagai piksel (Iqbal, 2017). Salah satu metode pada *image processing* adalah pendeteksian tepi berdasarkan *image gradient* (Image Gradient, 2017) dan *histogram equalization* (Elia, 2017).

Teknik Pengolahan Gambar Neighborhood Processing

Pada operasi titik, citra dimodifikasi dengan menerapkan suatu fungsi pada setiap nilai piksel. *Neighborhood processing* dapat dipandang sebagai perluasan dari operasi pengolahan titik, dimana fungsi diterapkan pada sejumlah piksel yang bertetangga (Anbarsanti, 2017). *Neighborhood processing* dilakukan dengan menggeser sebuah “*mask*” (yang dapat berbentuk segiempat atau bentuk lain) pada citra yang akan diolah. Konfigurasi *mask* atau *kernel* tersebut tergantung dari tipe pengaplikasiannya, apakah korelasi atau konvolusi. Dengan demikian dapat diperoleh citra baru yang piksel-pikselnya dihitung dari nilai piksel-piksel citra yang berada di bawah “*mask*”, lihat gambar berikut. *Mask* juga sering disebut sebagai *window* (Aditya, 2017).



Gambar 1. Mask pada pixel

Noise Reduction

Noise adalah efek samping dari penggunaan sensor elektronik yang dipakai untuk mengumpulkan cahaya (Dyrad, 2017). *Noise* merupakan sesuatu yang tidak diinginkan, namun akan selalu muncul sebagai akibat dari ketidaksempurnaan kinerja sensor (Tn, 2017) Berikut berbagai jenis *noise reduction*, antara lain :

- 1) *Noise Gaussian*, disebut juga *white noise* karena seluruh frekuensi spektral nya adalah sebagai cahaya putih (Ketutrare, 2017)
- 2) *Noise Salt and Pepper*, disebut derau impuls (*impulse noise*), *shot noise*, atau derau biner (*binary noise*) (Susilawati, 2017)
- 3) *Noise Speckle*, *speckle* merupakan *noise* ganda.

Macam-macam Filter Neighborhood Processing

Perbaikan citra dapat dilakukan dengan beberapa cara (Cloud, 2017):

1. LPF (*Low Pass Filter*), adalah suatu proses pada gambar/citra dari bentuk *filter* yang mengambil data pada frekuensi rendah dan membuang frekuensi tinggi yang mempunyai tujuan mengurangi *noise* pada suatu *image* atau pelembutan citra (*image smoothing*).



Gambar 2. Hasil citra setelah difilter dengan LPF

2. HPF (*High Pass Filter*), adalah suatu proses pada gambar/citra dari bentuk *filter* yang mengambil komponen frekuensi tinggi dan menurunkan komponen frekuensi rendah.



Gambar 3. Hasil citra setelah difilter dengan HPF

Beberapa jenis HPF, diantaranya:

1. Laplacian, *filter* deteksi tepi laplacian akan menampilkan tepian dalam segala arah.
2. Prewitt, menggunakan *filter* HPF yang diberi satu angka nol penyangga.
3. Sobel, digunakan untuk mengurangi *noise* sebelum melakukan perhitungan deteksi tepi.

PSNR dan SNR

PSNR adalah perbandingan antara nilai maksimum dari sinyal yang diukur dengan besarnya derau yang berpengaruh pada sinyal tersebut. PSNR digunakan untuk mengetahui perbandingan kualitas citra *cover* sebelum dan sesudah disisipkan pesan (Ketutrare, 2017). PSNR diformulasikan menjadi:

$$PSNR = 10 \log_{10} \left(\frac{C_{max}^2}{MSE} \right)$$

Sedangkan MSE dinyatakan sebagai *mean square error* yang diformulasikan menjadi:

$$MSE = \frac{1}{MN} \sum_{x=1}^M \sum_{y=1}^N (S_{xy} - C_{xy})^2$$

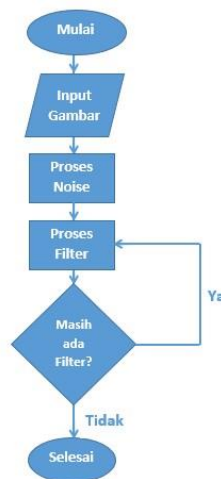
SNR atau *Signal To Noise Ratio* merupakan salah satu parameter yang digunakan untuk menentukan kualitas citra setelah dilakukan proses pengolahan citra. Citra yang telah diproses nantinya akan dibandingkan dengan citra asli untuk memberi perkiraan kasar kualitas citranya. SNR diformulasikan sebagai berikut:

$$SNR = 10 \log_{10} \left[\frac{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x,y)]^2}{\sum_0^{n_x-1} \sum_0^{n_y-1} [r(x,y) - t(x,y)]^2} \right]$$

METODE PENELITIAN

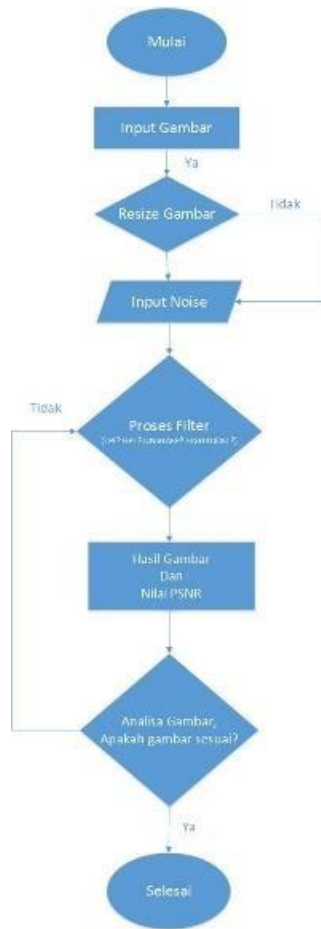
Diagram Alir Program

Berikut ini adalah diagram alir (*flowchart*) utama yang menjelaskan tentang gambaran umum dari simulasi yang dibangun. Flowchart utama ini juga akan menghasilkan flowchart untuk masing-masing program filter untuk menghilangkan noise pada gambar.



Gambar 4. Diagram Alir Utama Program

Diagram alir utama dijelaskan secara lebih spesifik oleh diagram alir program lengkap yang mengakumulasi variabel-variabel dalam program Matlab sehingga menghasilkan nilai output yang menyesuaikan parameter output.

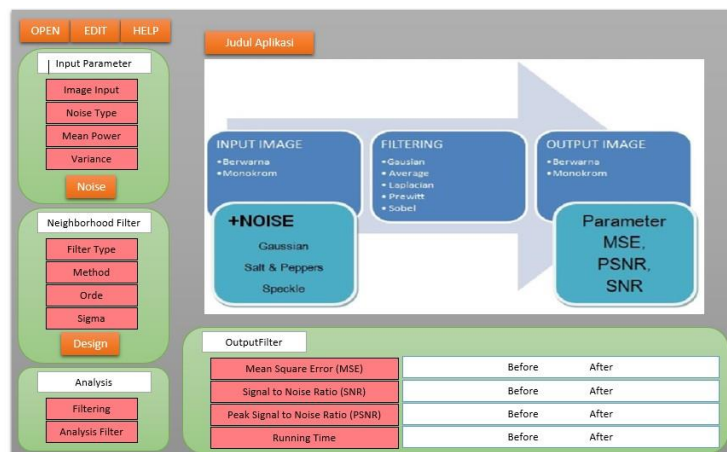


Gambar 5. Diagram Alir Program Lengkap

HASIL DAN PEMBAHASAN

Desain Interface pada MATLAB

Interface program pada penelitian ini adalah sebagai berikut:



Gambar 6. Tampilan awal aplikasi

Skenario Pengujian

Analisa aplikasi dilakukan dengan mengaplikasikan pada MATLAB (Dzil, 2017; Riazal, 2017; Sabrina, 2017) lalu membandingkan beberapa *filter*, dan analisa ini mencakup proses gambar yang akan di-*filter*. Gambar yang akan di uji coba terdapat 2 macam yaitu gambar Lena dan gambar pas foto. Untuk pengujian filter yang dibuat dalam tugas akhir ini, dilakukan 3 macam skenario pengujian yaitu sebagai berikut:

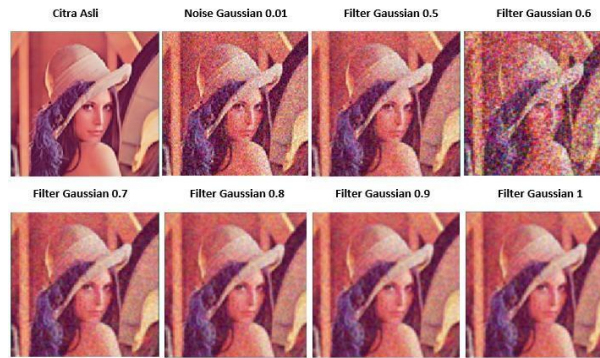
1. Skenario 1 :
 - a. *Noise Impulse Gaussian*
 - b. Filter yang diuji : LPF (*Gaussian, Average, Median*) dan HPF (*Laplacian, Prewitt, Sobel*)
2. Skenario 2 :
 - a. *Noise Impulse salt & pepper*
 - b. Filter yang diuji : LPF (*Gaussian, Average, Median*) dan HPF (*Laplacian, Prewitt, Sobel*)
3. Skenario 3 :
 - a. *Noise Impulse speckle*
 - b. Filter yang diuji : LPF (*Gaussian, Average, Median*) dan HPF (*Laplacian, Prewitt, Sobel*)

Hasil Pengujian dan Analisis

1. Skenario 1
 - a) Berikut ini hasil pengujian setelah citra diberi *noise Impulse Gaussian* dengan kepadatan 0,01. *Filter* yang digunakan adalah LPF Gaussian dengan parameter sigma dari 0.5 sampai 1.

Tabel 1. Hasil Pengujian pada Skenario 1 untuk *Noise Gaussian* dan *Filter Gaussian*

Nilai	Sigma	Nilai sesudah di- <i>filter</i>		
		Red	Green	Blue
MSE	0.5	68.34216	216.8164	64.28936
	0.6	68.15019	216.8164	65.03408
	0.7	69.56919	216.8146	54.82234
	0.8	62.53176	216.8146	52.1143
	0.9	66.59251	216.8146	52.45148
	1	71.95791	216.8146	54.23578
SNR	0.5	4.3537	0.69362	4.3053
	0.6	5.7198	0.69362	5.9231
	0.7	6.1609	0.69362	6.6641
	0.8	6.0935	0.69362	6.8849
	0.9	5.8203	0.69362	6.8569
	1	5.4837	0.69362	6.7116
PSNR (dB)	0.5	28.43	24.7699	28.3816
	0.6	29.7961	24.7699	29.9994
	0.7	30.2372	24.7699	30.7404
	0.8	30.1698	24.7699	30.9612
	0.9	29.8965	24.7699	30.9632
	1	29.56	24.7699	30.7879



Gambar 7. Perbandingan Citra pada Skenario 1 untuk *Noise Gaussian* dan *Filter Gaussian*

b) Berikut ini hasil pengujian setelah citra diberi *noise* Gaussian dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* LPF Average dengan nilai parameter sigma dari 0.5 sampai 1.



Gambar 8. Perbandingan Citra pada Skenario 1 untuk *Noise Gaussian* dan *Filter Average*

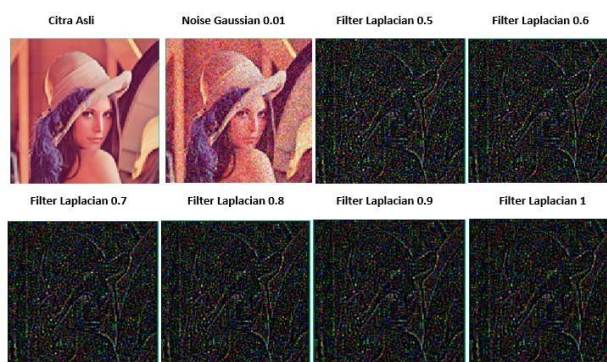
Tabel 2. Hasil Pengujian pada Skenario 1 untuk *Noise Gaussian* dan *Filter Average*

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	134.0108	216.8164	86.62223
	0.6	134.0108	216.8164	86.62223
	0.7	134.0108	216.8146	86.62223
	0.8	134.0108	216.8146	86.62223
	0.9	134.0108	216.8146	86.62223
	1	134.0108	216.8146	86.62223
SNR	0.5	2.7831	0.69362	4.6782
	0.6	2.7831	0.69362	4.6782
	0.7	2.7831	0.69362	4.6782
	0.8	2.7831	0.69362	4.6782
	0.9	2.7831	0.69362	4.6782
	1	2.7831	0.69362	4.6782
PSNR	0.5	26.8594	24.7699	28.7545
	0.6	26.8594	24.7699	28.7545
	0.7	26.8594	24.7699	28.7545
	0.8	26.8594	24.7699	28.7545
	0.9	26.8594	24.7699	28.7545
	1	26.8594	24.7699	28.7545

c) Berikut ini hasil pengujian setelah citra diberi *noise* Gaussian dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Laplacian dengan nilai parameter sigma dari 0.5 sampai 1.

Tabel 3. Hasil Pengujian pada Skenario 1 untuk *Noise* Gaussian dan *Filter* Laplacian.

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3296.4613	216.81455	3002.7469
	0.6	3304.0313	216.81455	3016.0274
	0.7	3315.0059	216.8146	3031.4322
	0.8	3325.3442	217.02732	3053.0258
	0.9	3336.3258	217.02732	3067.4325
	1	3346.7741	217.02732	3080.0919
SNR	0.5	-11.126	0.69362	-10.7207
	0.6	-11.1359	0.69362	-10.7398
	0.7	-11.1503	0.69362	-10.7717
	0.8	-11.1639	0.68936	-10.7928
	0.9	-11.1782	0.68936	-10.8132
	1	-11.1918	0.68936	-10.8311
PSNR	0.5	12.9503	24.7699	13.3556
	0.6	12.9404	24.7699	13.3366
	0.7	12.9262	24.7699	13.2835
	0.8	12.9124	24.7657	13.2835
	0.9	12.8981	24.7657	13.2631
	1	12.8845	24.7657	13.2452

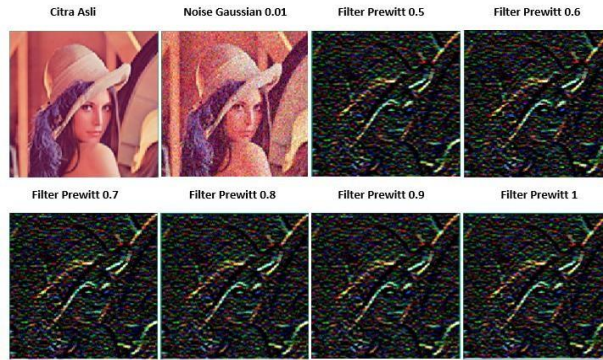


Gambar 9. Perbandingan Citra pada Skenario 1 untuk *Noise* Gaussian dan *Filter* Laplacian

d) Berikut ini adalah hasil pengujian setelah citra diberi *noise* Gaussian dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Prewitt dengan nilai parameter sigma dari 0.5 sampai 1.

Tabel 4. Hasil Pengujian Citra pada Skenario 1 untuk *Noise* Gaussian dan *Filter* Prewitt

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3167.8537	217.02732	2937.4505
	0.6	3167.8537	217.02732	2937.4505
	0.7	3167.8537	217.02732	2937.4505
	0.8	3167.8537	217.02732	2937.4505
	0.9	3167.8537	217.02732	2937.4505
	1	3167.8537	217.02732	2937.4505
SNR	0.5	-10.9531	0.68936	-10.6252
	0.6	-10.9531	0.68936	-10.6252
	0.7	-10.9531	0.68936	-10.6252
	0.8	-10.9531	0.68936	-10.6252
	0.9	-10.9531	0.68936	-10.6252
	1	-10.9531	0.68936	-10.6252
PSNR	0.5	13.1232	24.7657	13.4511
	0.6	13.1232	24.7657	13.4511
	0.7	13.1232	24.7657	13.4511
	0.8	13.1232	24.7657	13.4511
	0.9	13.1232	24.7657	13.4511
	1	13.1232	24.7657	13.4511

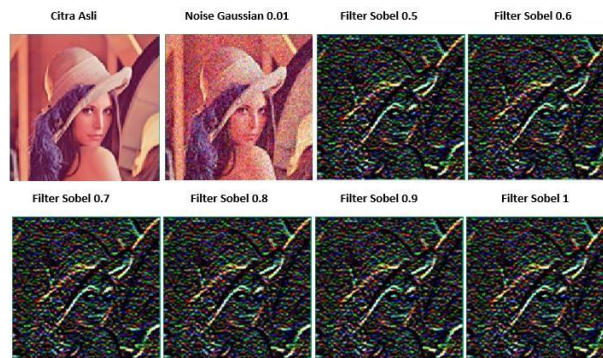


Gambar 10. Perbandingan Citra pada Skenario 1 untuk *Noise Gaussian* pada *Filter Prewitt*

e) Berikut ini adalah hasil pengujian setelah citra diberi *noise* Gaussian dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Sobel dengan nilai parameter sigma dari 0.5 sampai 1.

Tabel 5. Hasil Pengujian Citra pada Skenario 1 untuk *Noise* Gaussian dan *Filter* Sobel

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3254.7845	217.02732	2949.439
	0.6	3254.7845	217.02732	2949.439
	0.7	3254.7845	217.02732	2949.439
	0.8	3254.7845	217.02732	2949.439
	0.9	3254.7845	217.02732	2949.439
	1	3254.7845	217.02732	2949.439
SNR	0.5	-11.0707	0.68936	-10.6429
	0.6	-11.0707	0.68936	-10.6252
	0.7	-11.0707	0.68936	-10.6252
	0.8	-11.0707	0.68936	-10.6252
	0.9	-11.0707	0.68936	-10.6252
	1	-11.0707	0.68936	-10.6252
PSNR	0.5	13.0056	24.7657	13.4334
	0.6	13.0056	24.7657	13.4334
	0.7	13.0056	24.7657	13.4334
	0.8	13.0056	24.7657	13.4334
	0.9	13.0056	24.7657	13.4334
	1	13.0056	24.7657	13.4334



Gambar 11. Perbandingan Citra pada Skenario 1 untuk *Noise* Gaussian dan *Filter* Sobel

2. Skenario 2

a) Berikut ini hasil pengujian setelah citra diberi *Noise Salt and Pepper* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* LPF Gaussian dengan nilai parameter sigma dari 0.5 sampai 1.



Gambar 12. Perbandingan Citra pada Skenario 2 untuk *Noise Salt and Pepper* dan *Filter Gaussian*

Tabel 6. Hasil pengujian Citra pada Skenario 2 untuk *Noise Salt and Pepper* dan *Filter Gaussian*

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	33.0066	52.478	26.6245
	0.6	30.8656	52.478	24.0564
	0.7	33.9679	52.478	25.7579
	0.8	37.8081	52.478	28.2144
	0.9	41.146	52.478	30.4228
	1	43.8353	52.478	32.2721
SNR	0.5	8.8685	6.8547	9.8017
	0.6	9.15975	6.8547	9.9454
	0.7	8.7438	6.8547	9.9454
	0.8	8.2787	6.8547	9.5498
	0.9	7.9112	6.8547	9.2225
	1	7.6363	6.8547	8.9662
PSNR	0.5	32.9448	30.931	33.878
	0.6	33.2361	30.931	34.3185
	0.7	32.8201	30.931	34.0217
	0.8	32.355	30.931	33.6261
	0.9	31.9875	30.931	33.2988
	1	31.7126	30.931	33.0425

b) Berikut ini adalah hasil pengujian setelah citra diberi *noise Salt and Pepper* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* LPF Average dengan nilai parameter sigma dari 0.5 sampai 1.



Gambar 13. Perbandingan Citra pada Skenario 2 untuk *NoiseSalt and Pepper* dan *Filter Average*

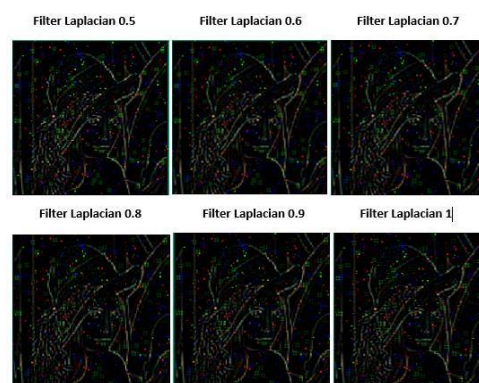
Tabel 7. Hasil Pengujian Citra di Skenario 2 (*Noise Salt and Pepper* dan *Filter Average*)

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	57.1358	52.478	41.4349
	0.6	57.1358	52.478	41.4349
	0.7	57.1358	52.478	41.4349
	0.8	57.1358	52.478	41.4349
	0.9	57.1358	52.478	41.4349
SNR	0.5	6.4854	6.8547	7.8808
	0.6	6.4854	6.8547	7.8808
	0.7	6.4854	6.8547	7.8808
	0.8	6.4854	6.8547	7.8808
	0.9	6.4854	6.8547	7.8808
PSNR	0.5	30.5617	30.931	31.9571
	0.6	30.5617	30.931	31.9571
	0.7	30.5617	30.931	31.9571
	0.8	30.5617	30.931	31.9571
	0.9	30.5617	30.931	31.9571

c) Berikut ini adalah hasil pengujian setelah citra diberi *noise Salt and Pepper* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Laplacian dengan nilai parameter sigma dari 0.5 sampai 1.

Tabel 8. Hasil Pengujian Citra di Skenario 2 (*Noise Salt and Pepper* pada *Filter* Laplacian)

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3893.32729	52.4780273	3694.66168
	0.6	3895.83822	52.4780273	3700.87406
	0.7	3899.65662	52.4780273	3706.64941
	0.8	3904.67263	52.4780273	3712.76575
	0.9	3908.71956	52.4780273	3717.69039
SNR	0.5	-11.8487	6.8547	-11.6212
	0.6	-11.8515	6.8547	-11.6285
	0.7	-11.8558	6.8547	-11.6353
	0.8	-11.8613	6.8547	-11.6425
	0.9	-11.8658	6.8547	-11.6482
PSNR	0.5	12.2276	30.931	12.4551
	0.6	12.2248	30.931	12.4478
	0.7	12.2205	30.931	12.441
	0.8	12.215	30.931	12.4338
	0.9	12.2105	30.931	12.4281

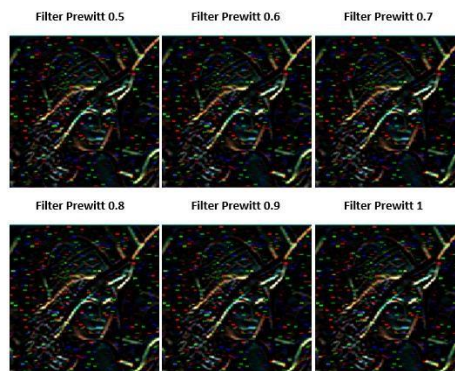


Gambar 14. Perbandingan Citra pada Pengujian Skenario 2 untuk *Noise Salt and Pepper* dan *Filter* Laplacian

d) Berikut ini hasil pengujian setelah citra diberi *noise Salt and Pepper* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Prewitt dengan nilai parameter sigma dari 0.5 sampai 1.

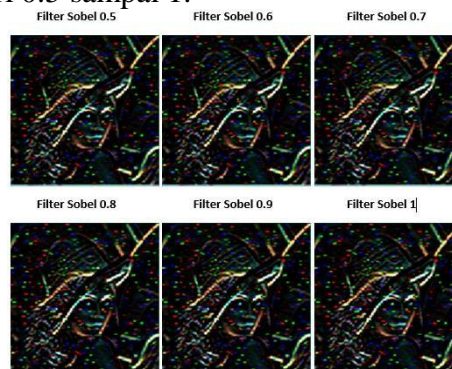
Tabel 9. Hasil Pengujian Citra pada Skenario 2 (Noise Salt and Pepper pada Filter Prewitt)

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3604.30017	68.5172119	3433.28318
	0.6	3604.30017	68.5172119	3433.28318
	0.7	3604.30017	68.5172119	3433.28318
	0.8	3604.30017	68.5172119	3433.28318
	0.9	3604.30017	68.5172119	3433.28318
	1	3604.30017	68.5172119	3433.28318
SNR	0.5	-11.5137	5.69651	-11.3026
	0.6	-11.5137	5.69651	-11.3026
	0.7	-11.5137	5.69651	-11.3026
	0.8	-11.5137	5.69651	-11.3026
	0.9	-11.5137	5.69651	-11.3026
	1	-11.5137	5.69651	-11.3026
PSNR	0.5	12.5626	29.7728	12.7737
	0.6	12.5626	29.7728	12.7737
	0.7	12.5626	29.7728	12.7737
	0.8	12.5626	29.7728	12.7737
	0.9	12.5626	29.7728	12.7737
	1	12.5626	29.7728	12.7737



Gambar 15. Perbandingan Citra di Pengujian Skenario 2 (*Noise Salt and Pepper* pada *Filter* Prewitt)

e) Berikut ini adalah hasil pengujian setelah citra diberi *Noise Salt and Pepper* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Sobel dengan nilai parameter sigma dari 0.5 sampai 1.



Gambar 16. Perbandingan Citra pada Pengujian Skenario 2 untuk Noise Salt and Pepper pada Filter Sobel

Tabel 10. Hasil Pengujian Citra pada Skenario 2 untuk Noise Salt and Pepper pada Filter Sobel

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3633.8586	68.517212	3404.0626
	0.6	3633.8586	68.517212	3404.0626
	0.7	3633.8586	68.517212	3404.0626
	0.8	3633.8586	68.517212	3404.0626
	0.9	3633.8586	68.517212	3404.0626
	1	3633.8586	68.517212	3404.0626
SNR	0.5	-11.5492	5.69651	-11.2655
	0.6	-11.5492	5.69651	-11.2655
	0.7	-11.5492	5.69651	-11.2655
	0.8	-11.5492	5.69651	-11.2655
	0.9	-11.5492	5.69651	-11.2655
	1	-11.5492	5.69651	-11.2655
PSNR	0.5	12.5271	29.7728	12.8108
	0.6	12.5271	29.7728	12.8108
	0.7	12.5271	29.7728	12.8108
	0.8	12.5271	29.7728	12.8108
	0.9	12.5271	29.7728	12.8108
	1	12.5271	29.7728	12.8108

3. Skenario 3

a) Berikut ini adalah hasil pengujian setelah citra diberi *Noise Speckle* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* LPF Gaussian dengan nilai parameter sigma dari 0.5 sampai 1.



Gambar 17. Perbandingan Citra pada Pengujian Skenario 3 untuk *Noise Speckle* pada *Filter* Gaussian

Tabel 11. Hasil Pengujian Citra pada Skenario 3 untuk Noise Speckle pada Filter Gaussian

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	23.57847	40.47306	21.16467
	0.6	24.13888	40.47306	21.30743
	0.7	28.96653	40.47306	22.85968
	0.8	33.67419	40.47306	25.74323
	0.9	37.50736	40.47306	28.20056
	1	40.46749	40.47306	30.138
SNR	0.5	10.3293	7.98284	10.7984
	0.6	10.2273	7.98284	10.978
	0.7	9.43554	7.98284	10.4638
	0.8	8.7815	7.98284	9.9479
	0.9	8.3133	7.98284	9.5519
	1	7.9834	7.98284	9.2634
PSNR	0.5	34.4056	32.0591	34.8747
	0.6	34.3036	32.0591	34.6543
	0.7	33.5118	32.0591	34.5401
	0.8	32.8578	32.0591	34.0242
	0.9	32.3896	32.0591	33.6282
	1	32.0597	32.0591	33.3397

b) Berikut ini adalah hasil pengujian setelah citra diberi *noise Speckle* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter LPF Average* dengan nilai parameter sigma dari 0.5 sampai 1.



Gambar 17. Perbandingan Citra Pengujian Skenario 3 (*Noise Speckle* dan *Filter Average*)

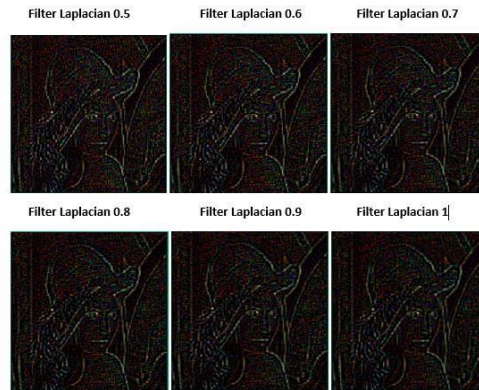
Tabel 12. Hasil Pengujian Citra pada Skenario 3 untuk *Noise Speckle* dan *Filter Average*

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	54.35429	40.47306	39.62392
	0.6	54.35429	40.47306	39.62392
	0.7	54.35429	40.47306	39.62392
	0.8	54.35429	40.47306	39.62392
	0.9	54.35429	40.47306	39.62392
	1	54.35429	40.47306	39.62392
SNR	0.5	6.7022	7.9828	8.0749
	0.6	6.7022	7.9828	8.0749
	0.7	6.7022	7.9828	8.0749
	0.8	6.7022	7.9828	8.0749
	0.9	6.7022	7.9828	8.0749
	1	6.7022	7.9828	8.0749
PSNR	0.5	30.7785	32.0591	32.1512
	0.6	30.7785	32.0591	32.1512
	0.7	30.7785	32.0591	32.1512
	0.8	30.7785	32.0591	32.1512
	0.9	30.7785	32.0591	32.1512
	1	30.7785	32.0591	32.1512

c) Berikut ini adalah hasil pengujian setelah citra diberi *noise Speckle* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter HPF Laplacian* dengan nilai parameter sigma dari 0.5 sampai 1.

Tabel 13. Hasil Pengujian pada Skenario 3 untuk *Noise Speckle* dan *Filter Laplacian*

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3445.2999	40.473063	3253.2552
	0.6	3476.1146	40.473063	3286.8063
	0.7	3503.2621	40.473063	3315.7654
	0.8	3526.2616	40.473063	3340.4159
	0.9	3545.9423	40.473063	3361.6968
	1	3562.8373	40.473063	3380.0223
SNR	0.5	-11.3178	7.9828	-11.0687
	0.6	-11.3564	7.9828	-11.1132
	0.7	-11.3902	7.9828	-11.1513
	0.8	-11.4186	7.9828	-11.1835
	0.9	-11.4428	7.9828	-11.2111
	1	-11.4635	7.9828	-11.2347
PSNR	0.5	12.7585	32.0591	13.0076
	0.6	12.7199	32.0591	12.9631
	0.7	12.6861	32.0591	12.925
	0.8	12.6577	32.0591	12.8928
	0.9	12.6335	32.0591	12.8652
	1	12.6128	32.0591	12.8416

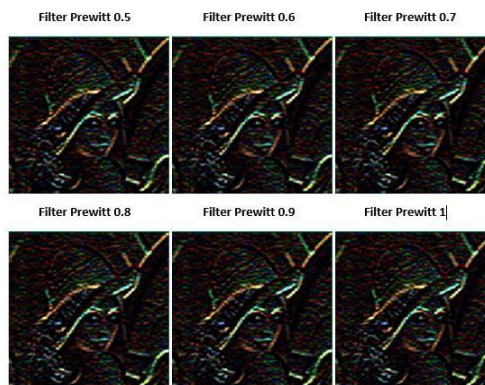


Gambar 18. Perbandingan Citra pada Pengujian Skenario 3 untuk *Noise Speckle* dan *Filter Laplacian*

- d) Berikut ini adalah hasil pengujian setelah citra diberi *noise Speckle* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Prewitt dengan nilai parameter sigma dari 0.5 sampai 1.

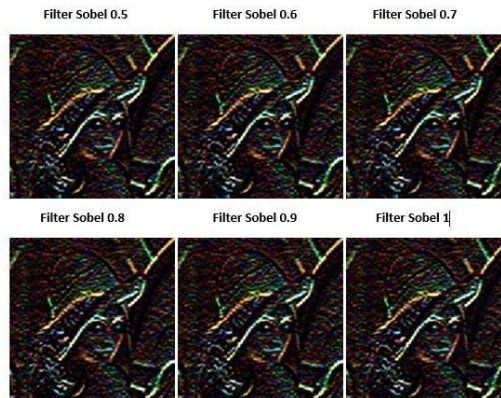
Tabel 14. Hasil Pengujian Citra pada Skenario 3 untuk *Noise Speckle* dan *Filter Prewitt*

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3329.5915	40.473063	3199.9624
	0.6	3329.5915	40.473063	3199.9624
	0.7	3329.5915	40.473063	3199.9624
	0.8	3329.5915	40.473063	3199.9624
	0.9	3329.5915	40.473063	3199.9624
	1	3329.5915	40.473063	3199.9624
SNR	0.5	-11.1694	7.9828	-10.9969
	0.6	-11.1694	7.9828	-10.9969
	0.7	-11.1694	7.9828	-10.9969
	0.8	-11.1694	7.9828	-10.9969
	0.9	-11.1694	7.9828	-10.9969
	1	-11.1694	7.9828	-10.9969
PSNR	0.5	12.9069	32.0591	13.0794
	0.6	12.9069	32.0591	13.0794
	0.7	12.9069	32.0591	13.0794
	0.8	12.9069	32.0591	13.0794
	0.9	12.9069	32.0591	13.0794
	1	12.9069	32.0591	13.0794



Gambar 20. Perbandingan Citra pada Pengujian Skenario 3 untuk *Noise Speckle* dan *Filter Prewitt*

- e) Berikut ini adalah hasil pengujian setelah citra diberi *noise Speckle* dengan kepadatan 0,01. *Filter* yang digunakan adalah *filter* HPF Sobel dengan nilai parameter sigma dari 0.5 sampai 1.



Gambar 21. Perbandingan Citra pada Pengujian Skenario 3 untuk *Noise Speckle* dan *Filter* Sobel

Tabel 15. Hasil Pengujian Citra pada Skenario 3 untuk *Noise Speckle* dan *Filter* Sobel

Nilai	Sigma	Nilai sesudah di-filter		
		Red	Green	Blue
MSE	0.5	3266.5528	40.473063	3056.6777
	0.6	3266.5528	40.473063	3056.6777
	0.7	3266.5528	40.473063	3199.9624
	0.8	3266.5528	40.473063	3199.9624
	0.9	3266.5528	40.473063	3199.9624
	1	3266.5528	40.473063	3199.9624
SNR	0.5	-11.0864	7.9828	-10.798
	0.6	-11.0864	7.9828	-10.798
	0.7	-11.0864	7.9828	-10.798
	0.8	-11.0864	7.9828	-10.798
	0.9	-11.0864	7.9828	-10.798
	1	-11.0864	7.9828	-10.798
PSNR	0.5	12.9899	32.0591	13.2783
	0.6	12.9899	32.0591	13.2783
	0.7	12.9899	32.0591	13.2783
	0.8	12.9899	32.0591	13.2783
	0.9	12.9899	32.0591	13.2783
	1	12.9899	32.0591	13.2783

Analisis Hasil Pengujian

Setelah dilakukan beberapa skenario pengujian dan melakukan analisa terhadap hasilnya, dapat ditarik kesimpulan sebagai berikut:

1. Pengenaan dari berbagai intensitas noise dapat diatasi dengan menggunakan filter Gaussian. Semakin tinggi intensitas filter, perbaikan citra akan memberikan hasil yang lebih baik.
2. Hal ini dapat dilihat secara kuantitatif dengan melihat hasil tabel pada filter Gaussian, dimana nilai PSNR hampir mendekati 30 dBm.
3. Perubahan nilai sigma untuk hasil MSE, SNR, dan PSNR akan tetap sama pada filter Average, Prewitt, dan Sobel.
4. Filter HPF akan menghasilkan PSNR yang rendah, dan berhasil untuk memunculkan edge tepian pada setiap gambar, karena HPF merupakan suatu proses pada gambar/citra dari bentuk filter yang mengambil komponen frekuensi tinggi dan menurunkan komponen frekuensi rendah.
5. Untuk perubahan nilai PSNR pada warna green tidak berubah, karena sigma tidak mempengaruhi nilai green.

KESIMPULAN

Kesimpulan yang dapat diambil dari pengolahan citra untuk melakukan proses *filter* menggunakan *filter neighborhood processing* dan menghilangkan *noise* menggunakan *noise reduction* adalah sebagai berikut:

1. Untuk mengurangi noise Gaussian, noise salt and pepper dan noise sepckle, filter terbaik filter Gaussian karena gambar yang dihasilkan yang digunakan adalah lebih halus dibandingkan dengan menggunakan filter-filter yang lain. Semakin tinggi intensitas filter, semakin baik hasilnya.
2. Hasil dari filter HPF yaitu mengambil komponen frekuensi tinggi dan mempertajam edge pada suatu citra.
3. Pemisahan warna citra RGB menjadi R, G, dan B cukup signifikan untuk dapat mengetahui intensitas masing-masing warna yang terdapat dalam gambar.
4. Filter LPF menghasilkan gambar yang lebih bagus dibandingkan HPF. Hasil gambar filter LPF lebih terlihat jelas dibandingkan dengan filter HPF, karena kedua filter ini memiliki fungsi yang berbeda.

DAFTAR PUSTAKA

- Anbarsanti, N. 2017. *Neighborhood Processing*. Jakarta: Diktat Perkuliahan.
- Aditya. 2017. *Pencitraan Korelasi dan Konvolusi pada Matlab*. [Online: <https://wiidhiet22.wordpress.com/2012/02/28/penggunaan-korelasi-dan-konvolusi-pada-matlab>, diakses tanggal 2 Mei 2017]
- Cloud, S. 2017. *Pengenalan LPF, HPF, BPF Pada Matlab*. [Online: http://jagocoding.com/tutorial/497/Pengenalan_Low_Pass_Filter_High_Pass_Filter_dan_Band_Pass_Filter_Pada_Matlab, diakses tanggal 2 Mei 2017.]
- Dyrad, P. 2017. *Apa itu Noise pada Kamera?*.
- Dzil, R. 2017. *Cara Menghitung SNR dan PSNR Pada Pengolahan Citra MATLAB*. [Online: <http://blog.pointopoin.com/2017/05/cara-menghitung-snr-dan-psnr-pada.html>, diakses tanggal 17 Mei 2017]
- Elia, T. 2017. *Aplikasi Peningkatan Kualitas Citra Menggunakan Metode Histogram Equalization* [Online: [https://Judul%20Baru%20Banget/citra%20digital%20\(Penting%202\).pdf](https://Judul%20Baru%20Banget/citra%20digital%20(Penting%202).pdf), diakses tanggal 20 April 2017.
- Image Gradient*. 2017. [Online: https://en.wikipedia.org/wiki/Image_gradient diakses tanggal 5 Mei 2017]
- Iqbal, M. 2017. *Dasar Pengolahan Citra Menggunakan Matlab*. [Online: <https://rakata89.files.wordpress.com/2012/01/13574039-basic-image-processing-matlab.pdf> diakses tanggal 27 April 2017.]
- Ketutrare. 2017. *Contoh Program Perhitungan PSNR dan MSE dengan Matlab*. [Online: tersedia di <http://ketutrare.blogspot.co.id/2014/11/contoh-program-perhitungan-mse-dan-psnr.html>, diakses tanggal 20 Mei 2017]
- Raizal. 2017. *Cara menghitung SNR dan PSNR pada Pengolahan Citra Matlab*. [Online: <http://www.pointopoin.com/2017/05/cara-menghitung-snr-dan-psnr-pada.html> diakses tanggal 20 Mei 2017]
- Sabrina. 2017. *Menambahkan Noise pada Citra Image & Menampilkannya dengan Menggunakan Matlab*. [Online: <http://hwsmartssolution.com/blog/2016/01/28/menambahkan-noise-pada-citraimage-menampilkannya-dengan-menggunakan-matlab/>, diakses tanggal 2 Mei 2017]

- Susilawati, I. 2017. *Teknik Pengolahan Citra*. [Online: http://ebook.repo.mercubuanyogya.ac.id/Kuliah/materi_20141_doc/6.%20Citra%20-%20TM5%20-%20EL%20-%20Neighborhood%20Processing.pdf diakses tanggal 1 Mei 2017]
- Tn. 2017. *Noise pada Foto Digital*. [Online: <http://belfot.com/noise-foto-kamera-digital>, diakses tanggal 17 Mei 2017]