



Perbandingan Algoritma untuk Mereduksi Noise pada Citra Digital

Ginanjar Setyo Nugroho¹, Gulam Hazmin²

^{1,2}Digital Business, Universitas PGRI Yogyakarta, Yogyakarta, Indonesia

Email: ¹ginanjar@upy.ac.id, ²gulam@upy.ac.id

Abstract

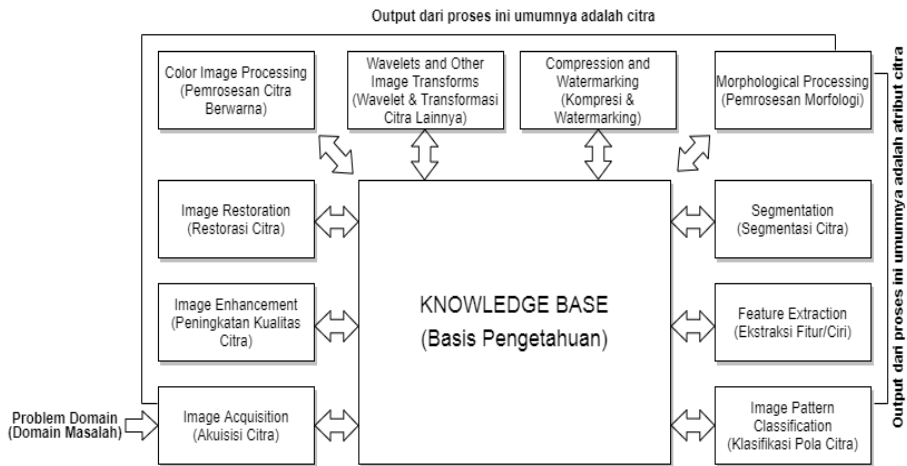
Image restoration is one of the stages in the field of Digital Image Processing. Image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation. The mathematical algorithm to reduce noise in digital images in this study uses 8 filtering algorithm methods. The purpose of this study is to compare 8 filtering algorithm and conclude which algorithm is the best for reducing noise in digital images. The method for generating noise uses Rayleigh Noise and Erlang (Gamma) Noise. The algorithm for reducing noise is Arithmetic Mean Filter, Geometric Mean Filter, Harmonic Mean Filter, Conharmonic Mean Filter, Geometric Mean Filter, Harmonic Mean Filter, Conharmonic Mean Filter, Median Filter, Maximum Filter, Minimum Filter, and Midpoint Filter. The measurement to determine which algorithm is the best using Root Mean Square Error (RMSE). Tests were carried out on 15 digital images by testing 1200 times. The conclusion of this study is that the best algorithm for noise reduction is Median Filter by resulting the smallest RMSE value of 6.0860942.

Keywords: Digital Image Processing, Image Restoration, Noise Reduction, Filtering Algorithm

1. PENDAHULUAN

Penelitian ini berfokus pada bidang *digital image processing* (pemrosesan citra digital). Bidang pengolahan citra digital mengacu pada pemrosesan citra digital yang mana menggunakan sebuah komputer digital [1]. Pemrosesan citra digital adalah merupakan sub bidang *digital signal processing* (pemrosesan sinyal digital) [2]. Citra/gambar digital terdiri dari sejumlah elemen, yang mana elemen ini memiliki lokasi dan nilai tertentu [1]. Elemen-elemen ini disebut *picture element*, *image elements*, *pels*, dan *pixels* [1]. Meski mempunyai sebutan yang banyak, *pixel* adalah istilah yang paling banyak digunakan untuk menunjukkan elemen citra digital [2]. Secara garis besar bidang pemrosesan citra digital langkah-langkahnya seperti gambar dibawah [1], [2]:





Gambar 1. Langkah-langkah pemrosesan citra digital

Pada penelitian ini, pemrosesan citra digital berfokus di *image restoration*. Pemilihan ini menurut penulis dilandasi karena hal dasar pengolahan citra digital ada di langkah *image restoration*. *Image acquisition* dan *image enhancement* bukanlah langkah awal untuk mengolah citra. Istilah *image acquisition* adalah langkah pertama dimana citra mentah awal yang hanya memberi petunjuk asal usul citra digital dan berisi tahap *preprocessing* seperti *scaling*, bahkan sederhana adalah kita diberi gambar yang sudah dalam bentuk digital [1]. Istilah *image enhancement* adalah langkah kedua dimana tidak ada teori umum tentang *image enhancement*, dan pemirsa adalah penilai tertinggi dari seberapa baik gambar yang dilihatnya [1]. Dengan kata lain *image enhancement* bersifat subjektif, dan hal-hal yang biasa dilakukan di langkah ini adalah pengaturan seperti *brightness*, *contrast*, *edge enhancement*, *pseudo coloring*, *sharpening* dan *magnifying* [1], [2]. Ide dasar dari *image enhancement* adalah memperbesar *contrast* antara area terang dan gelap [3]. *Image restoration* adalah langkah ketiga dimana langkah ini berbeda dari *image enhancement* yang bersifat subjektif, *image restoration* bersifat objektif [1]. Sifat objektif disini maksudnya adalah bahwa teknik di *image restoration* cenderung didasarkan pada model matematis atau probabilistik dari citra yang terdegradasi/terdapat penurunan kualitas citra [1]. Penurunan kualitas citra yang terjadi pada penelitian ini disebut sebagai *noise* (derau). Ketika *noise* pada citra bisa kita perbaiki dengan meningkatkan kualitasnya menggunakan algoritma yang ada di *image restoration*, bukankah hal ini bisa membantu banyak orang di bidang apapun. Selain itu, membandingkan algoritma-algoritma yang ada dan menyimpulkan algoritma mana yang terbaik adalah tujuan dari penelitian ini. Sehingga manfaat dari penelitian ini

adalah memberikan sumbangsih ilmu algoritma manakah yang terbaik yang dapat diaplikasikan untuk *image restoration* di bidang apapun di masa depan.

Secara khusus topik yang penulis pilih adalah *image noise reduction* (pengurangan derau pada citra). *Image noise reduction* adalah bidang paling penting dan menjadi salah satu konten penelitian paling dasar di bidang pemrosesan citra digital dan bahkan para peneliti di dalam dan luar negeri telah melakukan penelitian mendalam tentang teknologi pengurangan derau pada citra [4]. Saat ini, metode pengurangan derau pada citra secara kasar dapat dibagi menjadi dua kategori, yaitu metode pengurangan derau domain transformasi dan metode pengurangan derau domain spasial [4]. Penelitian ini berfokus pada metode pengurangan derau domain spasial.

Guna mendukung tujuan dari penelitian ini, penulis melakukan tinjauan pustaka dari penelitian sebelumnya guna menemukan kebaruan, keunikan dan menjawab *gap* dari penelitian-penelitian sebelumnya. Penulis menggunakan 8 kajian dari penelitian-penelitian sebelumnya. Kajian pertama menghasilkan kesimpulan bahwa metode *Gaussian*, *Mean* dan *Median* akan menghasilkan kecerahan dan kualitas gambar yang lebih baik dari citra digital aslinya [5]. Kesimpulan lainnya adalah metode *filtering Mean* merupakan metode terbaik dibandingkan dengan 2 metode lainnya dalam melakukan *noise reduction* [5]. Kekurangan dari penelitian ini adalah penyimpulan yang dibuat masih menggunakan grafik histogram serta tidak disebutkan metode apa yang ada di urutan nomor 2 dan 3.

Kajian kedua menghasilkan kesimpulan bahwa metode *Harmonic Mean Filter* dan *Canny* paling baik digunakan untuk mereduksi jenis *Salt and Pepper Noise* [6]. Kesimpulan lainnya yaitu bahwa *Harmonic Mean Filter* lebih baik digunakan untuk menghilangkan *Gaussian Noise* dan membuat garis tepi dari pada *Arithmetic Mean Filter*. Kekurangan dari penelitian ini adalah penyimpulan yang dibuat belum objektif atau belum menggunakan suatu rumus/cara pengukuran matematis. Kajian ketiga menghasilkan kesimpulan bahwa penerapan konvolusi di metode *filtering arithmetic* yang ukuran matriknya 3x3 lebih baik daripada tidak menggunakan konvolusi [7]. Kekurangan dari penelitian ini adalah penyimpulan yang dibuat merupakan teori mengenai *arithmetic mean filtering* itu sendiri. Kekurangan yang lainnya adalah hanya menggunakan 1 metode saja untuk menerapkan *noise reduction* dan karena hanya menggunakan 1 metode maka penyimpulannya juga bersifat subjektif atau dengan kata lain belum menggunakan suatu rumus/cara pengukuran matematis.

Kajian keempat menghasilkan kesimpulan bahwa metode *Harmonic Mean Filter* dapat meminimalisir *Gaussian Noise* pada citra rontgen [8]. Kekurangan dari

penelitian ini telah disebutkan juga di bagian saran yaitu: di penelitian selanjutnya dapat menerapkan beberapa metode sebagai perbandingan agar lebih akurat, dan melakukan perbandingan tipe harmonic dan tipe noise yang lain untuk mendapatkan tingkat akurasi yang lebih tinggi [8]. Kekurangan yang lain adalah penyimpulan yang dibuat belum objektif atau belum menggunakan suatu rumus/cara pengukuran matematis. Kajian kelima menghasilkan kesimpulan bahwa metode *Geometric Mean Filter* merupakan reduksi noise yang terbaik yang memiliki nilai rata-rata MSE terkecil [9]. Kesimpulan sebelumnya rancu karena pada penelitian ini hanya menggunakan 1 metode saja sehingga tidak dibandingkan dan mau berapapun nilai MSE, itu tidak bisa disebut terkecil ataupun terbesar. Kekurangan dari penelitian ini telah disebutkan di bagian saran yaitu: diharapkan penelitian selanjutnya dapat menerapkan algoritma yang lain seperti Harmonic Mean Filter dan Contraharmonic Mean Filter [9].

Kajian keenam menghasilkan kesimpulan bahwa algoritma *Arithmetic Mean Filter* memiliki hasil yang lebih baik dibanding algoritma *Contraharmonic Mean Filter* dalam mereduksi *Exponential Noise* yang dapat dilihat berdasarkan nilai MSE dan PSNR yang didapatkan [10]. Kesimpulan lainnya yaitu algoritma *Arithmetic Mean Filter* dalam mereduksi *Exponential Noise* menghasilkan nilai lebih kecil dan nilai PSNR lebih besar dibanding dengan menggunakan algoritma *Contraharmonic Mean Filter* [10]. Kekurangan dari penelitian ini telah disebutkan yaitu: diharapkan penelitian selanjutnya dapat menerapkan jenis *noise* yang lain selain *exponential noise* dan diharapkan penelitian selanjutnya dapat menggunakan jenis citra RGB (*true color*) sebagai objek citra yang diuji di sistem [10].

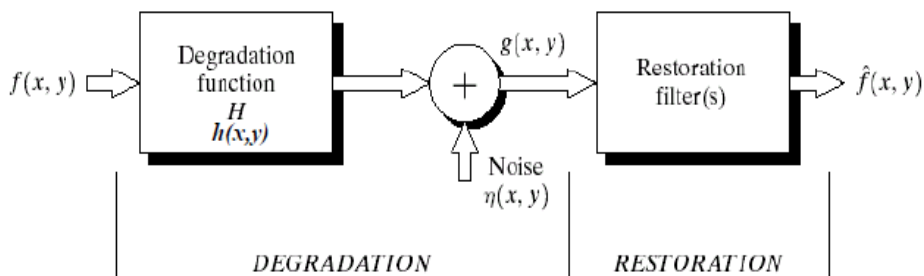
Kajian ketujuh menghasilkan kesimpulan bahwa algoritma *Arithmetic Mean Filter* dapat diterapkan pada citra, teknik konvolusi juga memudahkan proses reduksi dan hasil reduksinya juga baik [11]. Kekurangan dari penelitian ini adalah hanya menggunakan 1 metode saja untuk menerapkan *noise reduction*. Kekurangan yang lain adalah studi pustaka dari penelitian sebelumnya tidak dibahas pada penelitian ini. Kekurangan yang lain masih sama dengan beberapa kajian yang ada di penelitian ini, yaitu belum menggunakan suatu rumus/cara pengukuran matematis jadi sifatnya subjektif belum objektif. Kajian kedelapan menghasilkan kesimpulan bahwa reduksi citra bernoise salt and pepper sangat tepat menggunakan metode contra harmonic mean filter, proses reduksi dipengaruhi oleh nilai Q yang memudahkan untuk proses reduksi [12]. Kekurangan dari penelitian ini adalah hanya menggunakan 1 metode saja untuk menerapkan *noise reduction* dan belum menggunakan suatu rumus/cara pengukuran matematis jadi sifatnya subjektif belum objektif atau dengan kata lain kesuksesannya hanya dilihat dari penglihatan mata, apakah citra hasil reduksi lebih baik dari citra yang mempunyai *noise*.

Berdasarkan kekurangan dan saran yang telah disebutkan pada 8 kajian di atas, maka penelitian ini mencoba menjawab kekurangan dan *gap* yang ada sehingga menjadi penelitian yang baru dan unik. Perbedaan dan kebaruan pada penelitian ini adalah menggunakan 8 algoritma untuk *noise reduction*, menggunakan 2 jenis *noise*, menggunakan citra RGB (*true color*), menggunakan konvolusi matrik 3x3 dan menggunakan suatu pengukuran secara matematis untuk perbandingan algoritma yang digunakan.

2. METODOLOGI PENELITIAN

Pada penelitian ini, model penelitian yang dipakai adalah penelitian kuantitatif. Penelitian ini memiliki tujuan membandingkan algoritma-algoritma yang ada pada *image restoration* dan menyimpulkan algoritma mana yang terbaik untuk *noise reduction*. Secara garis besar penelitian ini menggunakan citra digital yang awal (dikatakan sebagai citra yang tanpa *noise*), lalu diberikan tambahan 2 *noise*, lalu dilakukan *image restoration* menggunakan 8 algoritma, lalu hasil *image restoration* ini dibandingkan dengan citra digital awal, setelah itu hasil dari nilai perbandingan ini digunakan untuk menyimpulkan algoritma mana yang terbaik untuk *noise reduction* berdasarkan 2 jenis *noise* yang ada.

Gambar dibawah merupakan gambar dari tahapan besar yang ada pada penelitian ini [1]. Citra digital awal adalah fungsi $f(x,y)$. Kemudian menggunakan fungsi degarasi dan penambahan *noise*, sehingga menghasilkan fungsi $g(x,y)$. Fungsi tersebut kemudian dilakukan proses *restoration* dan memperoleh fungsi fungsi $\hat{f}(x,y)$. Pada tahap ini telah ada 3 citra digital, yaitu: citra awal, citra dengan *noise*, dan citra hasil *image restoration*. Perhitungan untuk menyimpulkan algoritma mana yang terbaik adalah dengan menghitung perbedaan *pixel* dari citra digital awal dan citra hasil *image restoration*.



Gambar 2..Proses *image restoration*

Penelitian ini menggunakan 8 algoritma, yaitu: *Arithmetic Mean Filter*, *Geometric Mean Filter*, *Harmonic Mean Filter*, *Contraharmonic Mean Filter*, *Filter Median*, *Filter Maksimum*, *Filter Minimum*, dan *Midpoint Filter* (Filter Titik Tengah). Selain itu menggunakan 2 *noise*, yaitu: *Rayleigh Noise* dan *Erlang (Gamma) Noise*. Pengukurannya menggunakan *Root Mean Square Error* (RMSE).

2.1 Arithmetic Mean Filter

Arithmetic Mean Filter adalah metode yang paling sederhana dari *mean filter* [1]. *Arithmetic Mean Filter* mengkomputasi nilai rata-rata dari *corrupted image* $g(x,y)$ di area yang didefinisikan S_{xy} . Misalkan S_{xy} merupakan himpunan koordinat dalam sebuah kernel/window yang punya ukuran $m \times n$, berpusat di titik (x,y) . Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s, t) \quad (1)$$

2.2 Geometric Mean Filter

Geometric Mean Filter setiap *pixel* yang diperbaiki dimasukkan nilai hasil kali masing-masing *pixel* dalam *subimage* kernel/window, kemudian dipangkatkan dengan $1/mn$ [1]. Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \left[\prod_{(s,t) \in S_{xy}} g(s, t) \right]^{\frac{1}{mn}} \quad (2)$$

2.3 Harmonic Mean Filter

Harmonic Mean Filter bekerja dengan baik untuk *noise salt*, tapi gagal untuk *noise pepper* [1]. Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \frac{mn}{\sum_{(s,t) \in S_{xy}} \frac{1}{g(s,t)}} \quad (3)$$

2.4 Contraharmonic Mean Filter

Contraharmonic Mean Filter bekerja dengan baik untuk *noise salt and pepper*, hal ini dikarenakan algoritma ini memiliki variabel Q [1]. Q positif mengeliminasi *noise pepper* dan Q negatif mengeleminiasi *salt pepper* [1]. Eliminasi ini tidak dapat dilakukan untuk keduanya secara bersamaan [1].

Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \frac{\sum_{(s,t) \in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t) \in S_{xy}} g(s,t)^Q} \quad (4)$$

2.5 Filter Median

Filter Median memberikan hasil yang sangat bagus untuk citra yang terkena *noise impulse*, *bipolar* dan *unipolar* [1]. Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \text{median}_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (5)$$

2.6 Filter Maksimum

Filter Maksimum berguna untuk menemukan titik paling terang dalam gambar atau untuk mengikis daerah gelap yang berdekatan dengan area terang [1]. Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \max_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (6)$$

2.7 Filter Minimum

Filter Minimum berguna untuk menemukan titik paling gelap dalam gambar atau untuk mengikis daerah terang yang berdekatan dengan area gelap [1]. Selain itu juga dapat mengurangi *salt noise* sebagai akibat dari operasi min. Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \min_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (7)$$

2.8 Midpoint Filter (Filter Titik Tengah)

Midpoint Filter/Filter Titik Tengah menghitung titik tengah antara nilai maksimum dan minimum di area yang dicakup oleh filter [1]. Filter ini menggabungkan *order statistics* dan *averaging* [1]. Filter ini bekerja dengan sangat baik untuk *randomly distributed noise* seperti: *gaussian* atau *uniform noise* [1]. Berikut adalah persamaannya [1]:

$$\hat{f}(x, y) = \frac{1}{2} \left[\max_{(s,t) \in S_{xy}} \{g(s, t)\} + \min_{(s,t) \in S_{xy}} \{g(s, t)\} \right] \quad (8)$$

2.9 Root Mean Square Error (RMSE)

Root Mean Square Error (RMSE) adalah alat ukur kuantitatif untuk mengetahui perbedaan nilai dari citra digital awal dengan citra hasil *image restoration* [1]. Semakin kecil nilai yang dihasilkan, maka dapat dikatakan algoritma *image restoration* yang dipilih adalah tepat/akurat/cocok. Diperlukan operasi akar pada perhitungan ini agar hasil nilainya mendekati dengan nilai sebenarnya, hal terjadi karena pada tahap sebelumnya dilakukan operasi kuadrat. Jadi dengan kata lain, untuk mengetahui *error* atau selisih nilai prediksi dan nilai sebenarnya, dibutuhkan 2 citra digital, yaitu: citra digital awal dan citra digital hasil *image restoration*. Berikut adalah persamaannya [1]:

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x, y) - f(x, y)]^2 \right]^{1/2} \quad (9)$$

2.10 Probability Density Function (PDF)

Probability Density Function (PDF) digunakan untuk membangkitkan noise [1]. Noise yang umum ditemukan di aplikasi pengolahan citra digital adalah: *Gaussian Noise*, *Rayleigh Noise*, *Erlang (Gamma) Noise*, *Exponential Noise*, *Uniform Noise*, dan *Impulse (salt-and-pepper) Noise* [1]. Pada persamaan pembangkit noise terdapat nilai **a** dan **b**. Nilai-nilai ini dapat dicari menggunakan *mean* populasi dan variansi populasi [1]. Persamaan (10) merupakan PDF dari *Rayleigh Noise*, persamaan (11) merupakan *mean* dan persamaan (12) merupakan *variance* dari densitas *Rayleigh Noise* [1].

$$p(z) = \begin{cases} \frac{2}{b} (z - a) e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \quad (10)$$

$$\mu = a + \sqrt{\pi b/4} \quad (11)$$

$$\sigma^2 = \frac{b(4-\pi)}{4} \quad (12)$$

Persamaan (13) merupakan PDF dari *Erlang (Gamma) Noise*, persamaan (14) merupakan *mean* dan persamaan (15) merupakan *variance* dari densitas *Erlang Noise* [1]. Tanda ! artinya adalah faktorial.

$$f(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \geq 0 \\ 0 & \text{for } z < 0 \end{cases} \quad (13)$$

$$\mu = \frac{b}{a} \quad (14)$$






$$\sigma^2 = \frac{b}{a^2} \quad (15)$$











3. HASIL DAN PEMBAHASAN

3.1 Sampel Citra Digital

Bab ini dimulai dengan menunjukkan 15 sampel citra digital yang digunakan pada penelitian ini. Adapun 15 sampel citra digital ini ditunjukkan pada tabel dibawah:

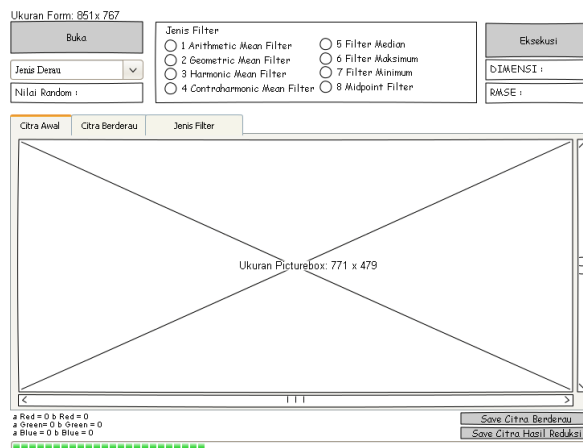
Tabel 1. Sampel citra digital yang di uji

No	Sampel citra digital	Dimensi gambar (lebar x tinggi)
1		535 pixel x 393 pixel
2		960 pixel x 854 pixel
3		640 pixel x 390 pixel
4		700 pixel x 466 pixel
5		852 pixel x 480 pixel

6		1040 pixel x 624 pixel
7		742 pixel x 487 pixel
8		400 pixel x 300 pixel
9		1280 pixel x 1024 pixel
10		1040 pixel x 768 pixel
11		600 pixel x 480 pixel
12		960 pixel x 600 pixel
13		800 pixel x 532 pixel
14		640 pixel x 413 pixel
15		600 pixel x 357 pixel

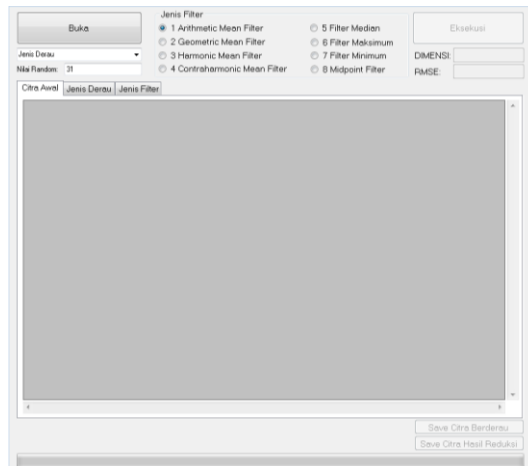
3.2 Implementasi Sistem

Implementasi sistem merupakan hasil dari rancangan antarmuka sistem. Rancangan antarmuka sistem ditampilkan pada gambar dibawah. Gambar dibawah menjelaskan ukuran form adalah 851 x 767, ukuran *picturebox* 771 x 479, terdapat 3 *picturebox*, *user* memilih satu dari 2 jenis *noise*/derau pada *combobox*, *user* memasukkan nilai di *textbox* nilai random dan *user* memilih satu dari 8 jenis filter pada *groupbox*. Selain itu karena citra yang kita *reduction* adalah citra RGB, maka ditampilkan informasi variabel a & b untuk RGB.



Gambar 3. Rancangan antarmuka sistem

Implementasi sistem terdiri dari 2 gambar. Gambar pertama merupakan implementasi sistem sebelum dilakukan proses *noise reduction*. Gambar pertama menunjukkan bahwa sistem terdiri dari 4 *button* yaitu *button* buka, *button* eksekusi yang belum aktif, *button* save citra berderau (citra dengan *noise*) yang belum aktif, *button* save citra hasil reduksi yang belum aktif, 1 *combobox* yang mempunyai teks jenis derau, 3 *textbox* yaitu nilai random dengan nilai *default* 31, dimensi, dan rmse, 1 *groupbox* jenis filter, dan 3 *picturebox* yaitu citra awal, jenis derau, dan jenis filter. Gambar dibawah adalah gambar pertama.



Gambar 4. Implementasi sistem sebelum dilakukan *noise reduction*

Gambar kedua merupakan implementasi sistem setelah dilakukan proses *noise reduction*. Gambar kedua menunjukkan bahwa sistem terdiri dari 4 *button* yaitu *button* buka, *button* eksekusi yang dinonaktifkan setelah *user* mengklik *button* eksekusi, *button* save citra berderau yang telah aktif, *button* save citra hasil reduksi yang telah aktif, dan 1 *combobox* yang teks telah dipilih, 3 *textbox* yaitu nilai random dengan nilai *default* 31, dimensi yang telah mempunyai nilai, dan rmse yang telah mempunyai nilai, 1 *groupbox* jenis filter yang telah mempunyai nilai, dan 3 *picturebox* yaitu citra awal, jenis derau yang telah dipilih, jenis filter yang telah dipilih, dan terdapat label informasi variabel a, b, mean, dan variansi untuk masing-masing RGB. Gambar dibawah adalah gambar kedua.



Gambar 5. Implementasi sistem setelah dilakukan *noise reduction*

3.3 Pengujian Sistem

Sistem diuji menggunakan 15 sampel citra digital. Setiap citra memiliki *noise* yang dibangkitkan menggunakan *Rayleigh Noise* dan *Erlang Noise*. Setiap citra yang memiliki *noise* dilakukan *noise reduction* menggunakan 8 algoritma, yaitu: *Arithmetic Mean Filter*, *Geometric Mean Filter*, *Harmonic Mean Filter*, *Contraharmonic Mean Filter*, Filter Median, Filter Maksimum, Filter Minimum dan Filter Titik Tengah. Guna membuat *noise*, maka diperlukan variabel random. Variabel ini bertindak sebagai pengatur apakah suatu pixel diberi *noise* atau tidak. Semakin besar nilai di variabel ini maka akan semakin kecil nilai RMSE yang dihasilkan. Nilai default dari variabel ini adalah 31. Nilai 31 diambil dari penjelasan bahwa *initial seed* (di penelitian ini disebut variabel random) seharusnya bernilai prima yang lebih dari 30 [13]. Karena hal ini, maka penelitian ini menggunakan nilai 31 karena nilai prima terkecil setelah 30 adalah 31.

Penulis menyadari bahwa setiap nilai RMSE selalu berbeda di setiap pengujian, maka penulis melakukan 5 kali pengujian untuk setiap jenis *noise* dan setiap jenis algoritma *noise reduction*. Hasil RMSE dari 5 kali pengujian tersebut dijumlahkan lalu dibagi 5, dan hasil rata-rata ini yang digunakan sebagai nilai RMSE. Total pengujiannya adalah 5 uji x 2 noise x 8 algoritma *noise reduction* x 15 sampel citra digital = 1200 kali uji.

3.4 Hasil Pengujian

Hasil pengujian berupa tabel nilai RMSE dari 15 sampel citra digital.

Tabel 2. Hasil *Noise Reduction* untuk sampel citra digital pertama

Jenis Noise	Algoritma Noise Reduction	RMSE
Rayleigh Noise	Filter Median	18.1020878
	Contraharmonic Mean Filter	21.0039868
	Arithmetic Mean Filter	24.6255236
	Harmonic Mean Filter	33.9942019
	Filter Maksimum	40.1462292
	Filter Titik Tengah	62.4108022
	Geometric Mean Filter	128.3714239
	Filter Minimum	131.0721716

Erlang Noise	Filter Median	18.2318513
	Contraharmonic Mean Filter	21.1135946
	Arithmetic Mean Filter	24.7085767
	Harmonic Mean Filter	34.0528678
	Filter Maksimum	40.1920793
	Filter Titik tengah	62.4024296
	Geometric Mean Filter	128.289758
	Filter Minimum	131.0155854

Tabel 2 merupakan hasil *noise reduction* untuk sampel citra digital pertama. Tabel 2 memperlihatkan bahwa urutan algoritma dengan nilai RMSE terkecil sampai nilai RMSE terbesar adalah sama baik untuk *Rayleigh Noise* dan *Erlang Noise*. Demi memperingkas penelitian ini, maka tabel hasil pengujian untuk citra digital kedua sampai citra digital kelima belas tidak ditampilkan. Setelah dilakukan pengujian, hasil untuk sampel citra digital kedua sampai citra digital kelima belas dari nilai RMSE terkecil sampai nilai RMSE terbesar untuk *Rayleigh Noise* dan *Erlang Noise* memiliki hasil yang sama seperti tabel 2. Dengan kata lain ketika salah satu sampel citra digital di uji, maka akan menghasilkan urutan nilai RMSE yang sama untuk 2 jenis *noise* yang dipakai. Meski demikian, dari 15 sampel yang diuji menghasilkan urutan algoritma dengan nilai RMSE terkecil sampai nilai RMSE terbesar adalah berbeda antara sampel satu dengan yang lainnya.

Tabel dibawah adalah tabel hasil uji untuk citra digital pertama sampai citra digital kelima belas. Keterangan tambahan untuk tabel dibawah adalah A1-A8 merupakan urutan algoritma pertama yang nilai RMSE terkecil sampai urutan algoritma kedelapan yang nilai RMSE terbesar. S1-S15 adalah sampel citra digital pertama sampai citra digital kelima belas. Med artinya algoritma median, con artinya algoritma *contraharmonic*, ari artinya algoritma *arithmetic*, har artinya algoritma *harmonic*, mak artinya algoritma maksimum, mid artinya algoritma *midpoint*, geo artinya algoritma *geometric*, dan min artinya algoritma minimum.

Tabel 3. Urutan algoritma terbaik untuk *noise reduction*

No	A1	A2	A3	A4	A5	A6	A7	A8
S1	med	con	ari	har	mak	mid	geo	min
S2	med	con	ari	har	mak	mid	geo	min
S3	med	con	mak	ari	har	mid	geo	min
S4	med	con	har	mak	ari	mid	geo	min
S5	med	con	ari	har	mak	mid	geo	min
S6	med	ari	con	har	mid	mak	geo	min

S7	med	ari	con	har	mid	mak	geo	min
S8	med	con	ari	har	mak	mid	geo	min
S9	med	ari	con	har	mid	mak	geo	min
S10	med	ari	con	har	mid	mak	geo	min
S11	med	ari	con	har	mid	mak	geo	min
S12	med	con	ari	har	mak	mid	geo	min
S13	med	ari	con	har	mid	mak	geo	min
S14	med	con	ari	har	mak	mid	geo	min
S15	med	ari	con	har	mak	mid	geo	min

Berdasarkan tabel 3, didapatkan hasil algoritma terbaik untuk *noise reduction* di *Rayleigh Noise* dan *Erlang Noise* pada 15 sampel citra digital adalah Filter Median. Selain itu, algoritma terburuk untuk *noise reduction* di *Rayleigh Noise* dan *Erlang Noise* pada 15 sampel citra digital adalah Filter Minimum.

4. KESIMPULAN DAN SARAN

4.1 Kesimpulan

Berdasarkan hasil & pembahasan pada bab sebelumnya, penulis memberi kesimpulan bahwa algoritma yang terbaik untuk *noise reduction* di *Rayleigh Noise* dan *Erlang Noise* melalui RMSE adalah Filter Median. Selain itu, penulis menyimpulkan bahwa urutan algoritma *noise reduction* di *Rayleigh Noise* dan *Erlang Noise* dimulai dari yang terbaik ke yang terburuk adalah Filter Median, *Contraharmonic Mean Filter*, *Arithmetic Mean Filter*, *Harmonic Mean Filter*, Filter Maksimum, Filter Titik Tengah, *Geometric Mean Filter*, dan Filter Minimum. Nilai RMSE terkecil untuk Filter Median adalah 6.0860942 dan Nilai RMSE terbesar untuk Filter Minimum adalah 152.4071194. Dengan demikian, algoritma yang dapat diaplikasikan untuk *image restoration* di bidang apapun di masa depan disarankan menggunakan algoritma Filter Median.

4.2 Saran

Saran untuk penelitian selanjutnya adalah melakukan perbandingan untuk jenis *noise* yang lain, misalnya: *Gaussian Noise*, *Exponential Noise*, *Uniform Noise*, dan *Impulse (salt-and-pepper) Noise* dengan 8 algoritma *noise reduction* yang telah diteliti di penelitian ini.

DAFTAR PUSTAKA

- [1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4th ed. Pearson

- Education Limited, 2018.
- [2] G. Gupta, "Algorithm for Image Processing Using Improved Median Filter and Comparison of Mean, Median and Improved Median Filter," *Int. J. Soft Comput.*, no. 5, pp. 304–311, 2011.
 - [3] S. Zhang, C. Pei, D. Sun, W. Liu, and L. Cao, "Industrial Image Enhancement Method Based on Cloud Edge Fusion," *Wirel. Commun. Mob. Comput.*, vol. 2022, 2022.
 - [4] Y. Jiang and C. Chen, "Partial Differential Equation Noise Reduction Model and Fuzzy Image Processing in Optimal Application of Sports Dance Exercise Training Mode," *Wirel. Commun. Mob. Comput.*, vol. 2022, pp. 1–9, 2022.
 - [5] A. Wedianto, H. L. Sari, and Y. S. H., "Analisa Perbandingan Metode Filter Gaussian, Mean dan Median terhadap Reduksi Noise," *J. Media Infotama*, vol. 12, no. 1, pp. 21–30, 2016.
 - [6] A. Y. Nasution and G. Ginting, "Implementasi Metode Harmonic Mean Filter Dan Canny untuk Mereduksi Noise pada Citra Digital," *J. Pelita Inform.*, vol. 6, no. 1, pp. 72–76, 2017.
 - [7] M. Fitri, "Implementasi Reduksi Noise Pada Citra Ultrasonografi (USG) Menggunakan Metode Mean Filter," *J. Pelita Inform.*, vol. 7, no. 3, pp. 433–435, 2019.
 - [8] D. Agusti and A. A. Nababan, "Penerapan Metode Harmonic Mean Filter Dalam Mereduksi Gaussian Noise Pada Citra Digital," *J. Nas. Komputasi dan Teknol. Inf.*, vol. 5, no. 3, pp. 565–571, 2022.
 - [9] R. R. Fiska and A. Allwine, "Perancangan Aplikasi Perbaikan dengan Reduksi Noise pada Citra dengan Metode Geometric Mean Filter," *J. Bisantara Inform.*, vol. 3, no. 2, pp. 1–12, 2019.
 - [10] M. Furqan, S. Sriani, and Y. K. Siregar, "Perbandingan Algoritma Conharmonic Mean Filter dan Arithmetic Mean Filter untuk Mereduksi Exponential Noise," *JISKA (Jurnal Inform. Sunan Kalijaga)*, vol. 5, no. 2, pp. 107–115, 2020.
 - [11] P. B. N. Simangunsong, "Reduksi Noise Pada Citra Digital Menggunakan Metode Arithmetic Mean Filter," *MEANS (Media Inf. Anal. dan Sist.*, vol. 02, no. 02, pp. 16–18, 2017.
 - [12] P. B. N. Simangunsong, "Reduksi Noise Salt And Pepper Pada Citra Digital Menggunakan Metode Conharmonic Mean Filter," *MEANS (Media Inf. Anal. dan Sist.*, vol. 2, no. 1, pp. 16–18, 2017.
 - [13] Massachusetts Institute of Technology, "Rayleigh Noise Generator." [Online]. Available: <https://lost-contact.mit.edu/afs/inf.ed.ac.uk/group/teaching/matlab-help/R2016b/comm/ref/rayleighnoisegenerator.html>. [Accessed: 24-Jul-2022].

