

# Restorasi Citra Digital Menggunakan Iterative Denoising dan Backward Projections with CNN

Irpan Adiputra Pardosi<sup>1</sup>, Pahala Sirait<sup>2</sup>, Williams Chandra<sup>3</sup>, William Makmur<sup>4</sup>

STMIK Mikroskil, Jl. Thamrin No. 112, 124, 140, Telp. (061) 4573767, Fax. (061) 4567789

Program Studi Teknik Informatika, STMIK Mikroskil, Medan

<sup>1</sup>irpan@mikroskil.ac.id, <sup>2</sup>pahala@mikroskil.ac.id, <sup>3</sup>williams.chandra96@gmail.com,

<sup>4</sup>williammakmur@gmail.com

## Abstrak

Citra yang memiliki kualitas buruk ataupun mengalami penurunan mutu umumnya diakibatkan gambar yang kurang tajam, kabur dan munculnya derau atau noise yang disebabkan karena ketidaksempurnaan dalam proses pengambilan gambar dan juga dikarenakan kotoran-kotoran yang terjadi pada sebuah citra. Noise Salt and Pepper merupakan satu dari antara beberapa noise berupa titik-titik hitam atau putih yang tersebar pada sebuah citra yang mempengaruhi kejelasan dari gambar. Banyak metode yang dapat digunakan untuk mengurangi noise pada citra, dan pada penelitian ini metode yang digunakan dalam mengurangi salt and pepper dengan Iterative Denoising dan Backward Projection. Iterative yang merupakan sebuah metode yang mengambil nilai pixel dari pixel sebelumnya dan diiterasi terus sampai kriteria berhenti dicapai yang menandakan bahwa citra sudah bersih dari noise. Hasil pengujian terhadap 7 buah gambar JPG dengan persentase noise mulai 5% hingga kelipatan 5 sampai batas ukuran nilai PSNR diatas 30 dB dan didapatkan hasil pengujian denoising yang memenuhi hingga persentase noise 51% dengan PSNR 30.32 dB menggunakan threshold 192 yang digunakan untuk mengecek citra sudah bersih dari noise atau belum. Threshold yang ideal dibutuhkan untuk mendapatkan iterasi yang tidak berlebihan atau kekurangan yang menyebabkan noise masih bersisa pada citra.

**Kata kunci**—Restorasi Citra, Iterative Denoising, Backward Projection, Salt and Pepper, Noise

## Abstract

Images that have poor quality or decreasing in quality are generally caused by images that are sharpless, blurred and the appearance of noise or noise caused by imperfections in the process of taking pictures and also due to the dirt that occurs in an image. Salt and Pepper noise is one of several noise in the form of scattered black or white dots on an image that affects the clarity of the image. Many methods can be used to reduce noise in images, and in this study the method used in reducing salt and pepper with Iterative Denoising and Backward Projection. Iterative which is a method that takes the pixel value from the previous pixel and continues iterates until the halt criteria are reached which indicates that the image is clear of noises. Results of testing for 7 JPG images with a percentage of noise from 5% to increase of 5 to the size limit of PSNR values above 30 dB and obtained denoising test results that meet up to 51% noise percentage with PSNR 30.32 dB using threshold 192 which is used to check the image is clean from noise or not. The ideal threshold is needed to get an iteration that is not excessive or lacking that causes noise to remain in the image

**Keywords**—Image Restoration, Iterative Denoising, Backward Projection, Salt and Pepper, Noise

## 1. PENDAHULUAN

Citra yang memiliki kualitas buruk ataupun mengalami penurunan mutu umumnya diakibatkan gambar yang kurang tajam, kabur dan munculnya derau atau noise [1]. Noise (derau) pada citra digital dapat terjadi karena banyak faktor, seperti kurangnya pencahayaan saat pengambilan gambar dan keterbatasan resolusi *pixel* dari kamera yang digunakan. Hal ini mengakibatkan informasi yang diterima sering mengalami gangguan sehingga hasilnya tidak sesuai dengan yang diharapkan. Noise juga

menyebabkan sebuah nilai intensitas piksel tidak mencerminkan nilai intensitas piksel yang sebenarnya [2]. Jenis noise *Salt and Pepper* merupakan salah satu jenis *noise* yang sering terdapat pada citra digital yang mengganggu karena mengurangi kualitas citra pada saat akan diproses ke tahap berikutnya dalam pengenalan objek atau saat pencetakan, sehingga citra yang mengalami gangguan (*noise*) perlu diperbaiki agar objek dalam citra dapat terdeteksi [3]. Proses reduksi noise sangat berdampak pada kualitas citra sehingga perlu mempertimbangkan besarnya persentasi noise berdasarkan kualitas citra setidaknya diatas 30 db [4].

Beberapa penelitian sejenis yang pernah dilakukan sebelumnya dengan metode *denoising* pada citra digital seperti *Salt and Pepper Noise Removal* dengan *Spatial Median Filter (SMF)* dan *Adaptive Noise Reduction (ANR)* dengan pengujian yang dilakukan terhadap persentasi *noise* yang berbeda pada masing-masing gambar mulai dari 5% sampai 75% dengan range 5%, dengan kesimpulan kualitas citra dengan metode SMF lebih baik pada persentasi noise diatas 20% [1]. Namun, proses *Median Filter* terhadap semua *pixel* justru menyamakan citra asli yang sebenarnya sudah benar dan tidak perlu diperbaiki dan justru menyebabkan turunnya kualitas citra hasil perbaikan [5]. Penelitian lainnya menggunakan metode *Adaptive Fuzzy Filter (AFF)* dan *Spatial Median Filter (SMF)* dengan persentasi noise 45% mampu menghasilkan kualitas citra hingga 28.10 dB namun masih dibawah standar kualitas citra yang baik [4].

Untuk menyelesaikan permasalahan dalam mereduksi noise dengan mendapatkan kualitas citra diatas 30dB, maka dapat diterapkan metode *Iterative Denoising and Backward Projections* untuk proses restorasi citra digital [6]. Metode ini mampu melakukan proses rekonstruksi citra dari versi degradasinya yang memiliki *noise*, *blur*, dan *downsampled*. Metode ini menggunakan konsep yang sama seperti metode *Plug and Play (P&P)*, yang menyediakan sebuah cara elegan untuk memisahkan model pengukuran dan gambar, sedemikian rupa sehingga gambar ditangani semata-mata oleh *denoising* operasi. Metode P&P telah diterapkan dalam berbagai aplikasi seperti *bright field electron tomography*, *poisson denoising*, dan *postprocessing* dari citra terkompresi [7]. Metode *Iterative Denoising and Backward Projections* merupakan sebuah metode iteratif sederhana untuk menyelesaikan problema *invers linier* dengan menggunakan algoritma *denoising*, yang menyediakan sebuah alternatif untuk P&P. Metode ini memerlukan parameter yang lebih sedikit dan jumlah iterasi yang lebih sedikit daripada P&P [6]. Penerapan metode *Iterative Denoising and Backward Projections* dapat menggunakan bantuan algoritma *Convolutional Neural Network (CNN)*. Algoritma *Convolutional Neural Network (CNN)* ini digunakan sebagai operator *denoising* pada metode *Iterative Denoising and Backward Projections*, sehingga algoritma ini sering disebut sebagai *Iterative Denoising and Backward Projections-Convolutional Neural Network (IDBP-CNN)*.

## 2. METODE PENELITIAN

### 2.1 Algoritma Iterative Denoising and Backward Projections (IDBP)

Metode *Iterative Denoising and Backward Projections* merupakan sebuah metode iteratif sederhana untuk menyelesaikan problema *invers linier* dengan menggunakan algoritma *denoising*, yang menyediakan sebuah alternatif untuk P&P. Metode ini memerlukan parameter yang lebih sedikit dan jumlah iterasi yang lebih sedikit daripada P&P. *Pseudocode* dari metode *Iterative Denoising and Backward Projections* dapat dirincikan sebagai berikut: [6]

**Input:**  $H, y, \sigma_e$ , operator *denoising*  $D(.,\sigma)$ , kriteria berhenti,  $y = Hx + e$ , sedemikian sehingga  $e \sim N(0, \sigma_e I_m)$  dan  $x$  adalah sebuah sinyal tidak dikenal dimana modelnya dispesifikasikan oleh  $D(.,\sigma)$

**Output:**  $\chi$  adalah estimasi dari  $x$

**Inisialisasi:**  $y_0$  = inisialisasi awal,  $k = 0$ , persyaratan  $\delta$ .

**while** {kriteria berhenti belum tercapai} **do**

$k = k + 1$ ;

$\chi_k = D(y_{k-1}; \sigma_e + \delta)$ ;

$$y_k = H^T y + (I_n - H^T H) \chi_k;$$

end

$$\chi = \chi_k;$$

Keterangan:

D = fungsi *operator denoising*.

H = matriks degradasi  $m \times n$ .

$H^\dagger$  = *pseudo invers* dari matriks H.

$\chi$  = warna piksel citra hasil restorasi.

$\chi_k$  = warna piksel citra hasil restorasi untuk iterasi ke-k.

y = warna piksel citra yang sedang diamati.

$y_k$  = warna piksel citra untuk pengamatan berikutnya.

$\delta$  = parameter persyaratan.

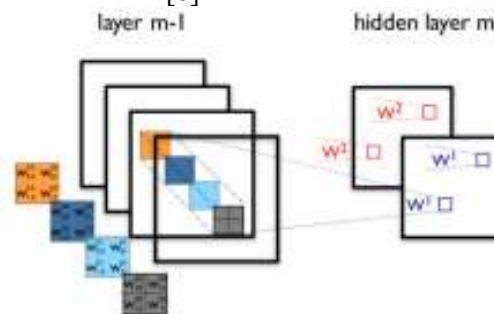
$\sigma_e$  = standar deviasi.

$I_n$  = matriks dengan ukuran  $n \times n$ .

Algoritma *Iterative Denoising and Backward Projections* memerlukan *input* berupa *operator denoising* dan kriteria berhenti. *Output* dari algoritma ini adalah nilai estimasi untuk setiap piksel x pada citra digital yang dimasukkan. Proses kerja dari algoritma ini akan menghitung estimasi setiap nilai piksel berdasarkan nilai piksel sebelumnya. Proses ini akan diulangi hingga kriteria berhenti tercapai [6].

## 2.2 Convolution Neural Network

*Convolutional Neural Network* (CNN) termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Cara kerja CNN memiliki kesamaan pada *Multilayer Perceptron* (MLP), namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi [8]. Data yang dipropagasikan pada jaringan CNN adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi, Karena sifat proses konvolusi, maka CNN hanya dapat digunakan pada data yang memiliki struktur dua dimensi seperti citra dan suara [8].



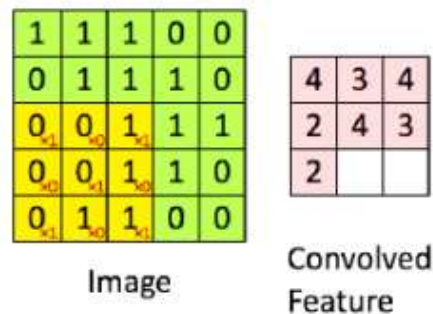
Gambar 1 Proses Konvolusi pada CNN [8]

Sebuah CNN terdiri dari beberapa *layer*. Berdasarkan arsitektur LeNet5, terdapat empat macam *layer* utama pada sebuah CNN namun yang diterapkan pada TA ini hanya tiga macam lapisan antara lain [8]:

### 1. Convolution Layer

*Convolution Layer* melakukan operasi konvolusi pada output dari *layer* sebelumnya. *Layer* tersebut adalah proses utama yang mendasari sebuah CNN dengan tujuan untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial

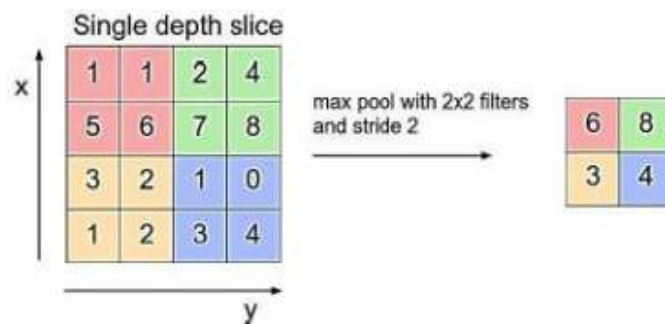
pada data. Bobot pada *layer* tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN. [8]



Gambar 2 Operasi Konvolusi [8]

## 2. *Subsampling Layer*

*Subsampling* adalah proses mereduksi ukuran sebuah data citra dan bisa juga untuk meningkatkan invariansi posisi dari fitur. Dalam sebagian besar CNN, metode *subsampling* yang digunakan adalah *max pooling*. Proses tersebut memastikan fitur yang didapatkan akan sama meskipun objek citra mengalami translasi (pergeseran) dalam hal ini *pooling layer* untuk mereduksi ukuran citra sehingga dapat dengan mudah digantikan dengan *convolution layer* dengan stride yang sama dengan *pooling layer* yang bersangkutan. [8]



Gambar 3 Operasi *Max Pooling* [8]

## 3. *Fully Connected Layer*

*Layer* tersebut adalah *layer* yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap *neuron* pada *convolution layer* perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah *fully connected layer*. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, *fully connected layer* hanya dapat diimplementasikan di akhir jaringan. Dalam sebuah jurnal oleh Lin et al., dijelaskan bahwa *convolution layer* dengan ukuran kernel 1 x 1 melakukan fungsi yang sama dengan sebuah *fully connected layer* namun dengan tetap mempertahankan karakter spasial dari data. Hal tersebut membuat penggunaan *fully connected layer* pada CNN sekarang tidak banyak dipakai.

Fungsi aktivasi adalah fungsi non linear yang memungkinkan sebuah JST untuk dapat mentransformasi data input menjadi dimensi yang lebih tinggi sehingga dapat dilakukan pemotongan hyperlane sederhana yang memungkinkan dilakukan klasifikasi. Dalam CNN terdapat fungsi aktivasi digunakan yaitu fungsi sigmoid. Fungsi *sigmoid* mentransformasi *range* nilai dari input  $x$  menjadi antara 0 dan 1. Fungsi *sigmoid* sekarang sudah tidak banyak digunakan dalam praktek karena memiliki kelemahan utama yaitu *range* nilai *output* dari fungsi *sigmoid* tidak terpusat pada angka nol. Hal tersebut menyebabkan terjadinya proses *backpropagation* yang tidak ideal, selain itu bobot pada JST tidak terdistribusi rata antara nilai positif dan negatif serta nilai bobot akan banyak mendekati ekstrim 0 atau 1. Dikarenakan komputasi nilai propagasi menggunakan perkalian, maka nilai ekstrim tersebut akan

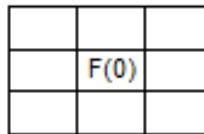
menyebabkan efek *saturating gradients* dimana jika nilai bobot cukup kecil, maka lama kelamaan nilai bobot akan mendekati salah satu ekstrim sehingga memiliki gradien yang mendekati nol. Jika hal tersebut terjadi, maka neuron tersebut tidak akan dapat mengalami update yang signifikan dan akan nonaktif [8].

### 2.3 Mean Absolute Gradient (MAG)

Untuk mengklasifikasikan piksel *corrupted* atau tidak, akan digunakan metode *Mean Absolute Gradient* (MAG). MAG yang kecil mengindikasikan sebuah daerah datar dan MAG yang besar biasanya mengindikasikan daerah kompleks atau daerah impulse noise. MAG dapat didefinisikan sebagai berikut:

$$MAG = \frac{1}{N-1} \sum_{i=0}^{n-1} F(0) - F(i) \quad (1)$$

Dimana  $F(i)$  melambangkan nilai intensitas dari piksel pada sebuah region.  $N$  adalah total piksel pada region.  $F(0)$  adalah intensitas dari piksel pada bagian tengah seperti terlihat pada gambar berikut: [3]



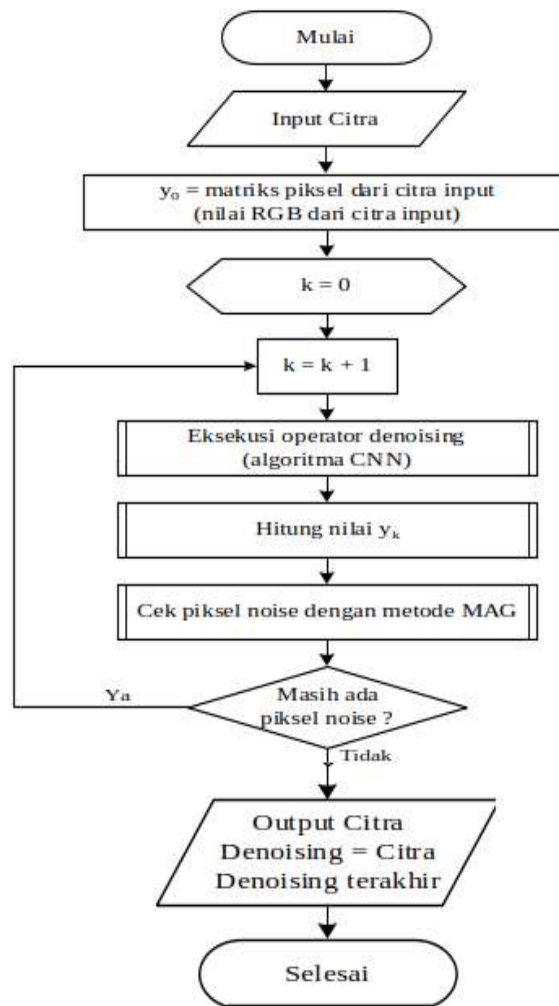
Gambar 4 Window MAG 3 x 3. [3]

$$z_{ij} = \begin{cases} 1 & MAG > T \\ 1 & x_{ij} = 0 \\ 1 & x_{ij} = 255 \\ 0 & MAG \leq T \end{cases} \quad (2)$$

Sebuah nilai threshold  $T$  dapat ditentukan berdasarkan hasil pengujian. Jika  $Z_{ij} = 1$ , maka piksel  $X_{ij}$  akan ditandai sebagai kandidat noise, jika tidak, berarti  $X_{ij}$  tidak memiliki noise [3]

### 2.4 Analisis Proses

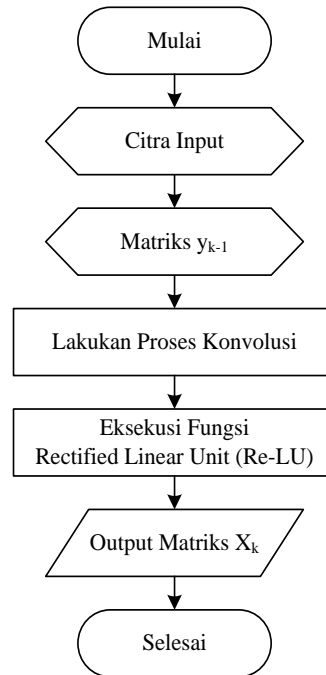
Aplikasi Restorasi Citra dengan *Iterative Denoising and Backward Projections* memiliki proses kerja yang dapat digambarkan sebagai berikut:



Gambar 5 Flowchart dari Algoritma *Iterative Denoising and Backward Projections - Convolutional Neural Network (IDBP-CNN)*.

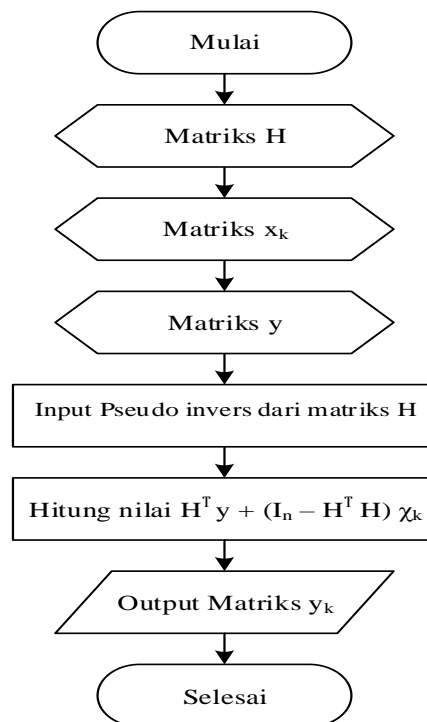
Pada gambar 5 proses yang akan dilakukan dimulai dengan mengambil nilai RGB dari citra input yang akan menjadi  $y_0$ , kemudian akan dilanjutkan dengan *iterasi* dan akan dilakukan proses *Operator Denoising CNN* untuk mendapatkan nilai yang akan digunakan untuk menghitung nilai yang akan digunakan untuk mencari nilai MAG dimana jika nilai MAG yang dihasilkan lebih besar dari MAG yang ditentukan maka citra dianggap masih memiliki *noise* dan akan dilakukan perulangan kembali sampai nilai MAG yang dihasilkan lebih kecil dari yang ditentukan yang menandakan citra sudah bersih dari *noise*.

Proses kerja dari operator *denoising* dengan algoritma *Convolutional Neural Network (CNN)* yang dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



Gambar 6 Flowchart Operator Denoising dengan Algoritma Convolutional Neural Network (CNN)

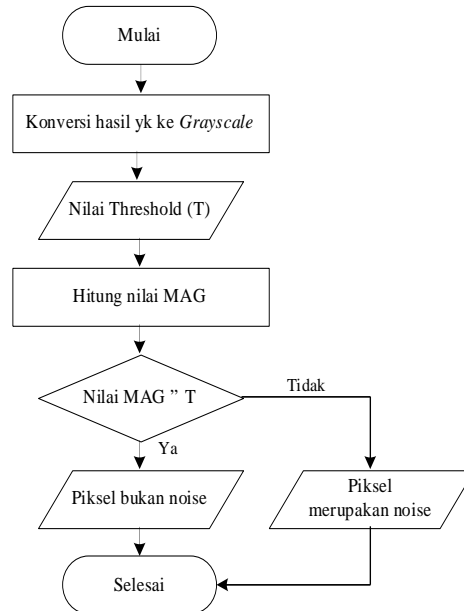
Pada gambar 6 dilakukan proses *Operator Denoising* dengan algoritma *Convolutional Neural Network* (CNN) dengan tahapan Melakukan proses perhitungan metode konvolusi dari nilai RGB citra input. Setelah melakukan proses metode konvolusi maka akan melakukan proses fungsi *Rectified Linear Unit* (Re-LU) yang akan menjadi output  $\chi_k$ . Proses perhitungan nilai  $y_k$  yang dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



Gambar 7 Flowchart dari perhitungan Nilai  $y_k$

Pada gambar 7 dilakukan proses perhitungan untuk mencari Nilai  $y_k$ , yaitu dengan rumus  $H^T y + (I_n - H^T H) \chi_k$  yang akan menjadi output dari matriks  $y_k$

Proses pengecekan *noise* dengan metode *Mean Absolute Gradient* (MAG) yang dapat digambarkan dalam bentuk *flowchart* seperti terlihat pada gambar berikut:



Gambar 8 *Flowchart* dari Cek *Piksel Noise* dengan Metode *Mean Absolute Gradient* (MAG)

Pada Gambar 8 akan mengambil hasil dari  $y_k$  dan akan dikonversi ke Grayscale terlebih dulu kemudian menentukan nilai *Threshold* yang akan digunakan, Setelah itu akan menghitung nilai MAG dari nilai  $y_k$  dan jika nilai MAG yang dihasilkan lebih besar dari *threshold* yang digunakan maka citra dianggap masih memiliki *noise* jika lebih kecil dari *threshold* yang ditentukan maka citra akan dianggap sudah bersih dari *noise*.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil

Proses penambahan *noise* dilakukan pada aplikasi dengan menginput persentase *noise* yang diinginkan dan untuk melakukan proses *denoising* terhadap citra *noise* perlu untuk memberikan nilai T (nilai 192 default) sistem akan menampilkan citra hasil seperti gambar gambar 9 berikut:



Gambar 9 Tampilan Form Restorasi Citra Setelah Proses



Untuk melihat kualitas citra hasil pengujian, dengan mengklik tombol **lanjut** pada *form* restorasi citra (gambar 9) dan tombol **proses** untuk menampilkan hasil citra seperti pada gambar 10 di bawah.



Gambar 10 Tampilan Form Pengujian Setelah Proses





### 3.2 Pembahasan

Beberapa skenario pengujian dilakukan untuk mendapatkan hasil yang valid dan terukur terhadap algoritma yang diimplementasikan, diantaranya:

1. Pengujian *threshold* dengan menguji nilai *threshold* yang paling ideal dari nilai *threshold* antara 10-255 dengan kenaikan 10 per pengujian dan kemudian akan mengambil nilai PSNR yang paling bagus dan akan diuji kembali dengan kenaikan 1 per pengujian untuk mencari *threshold* yang paling ideal.
2. Pengujian citra persentasi noise paling tinggi berdasarkan nilai PSNR diatas 30 dB dengan persentasi mulai dari 5% dan kelipatan 5 (5%, 10%, 15%, 20%, 25%, dst)

Kedua skenario akan diuji coba pada dataset sebanyak 5 citra jpg yang dirincikan sebagai berikut:

Tabel 1 Dataset citra yang digunakan

Nama Citra	Tampilan Citra	Nama Citra	Tampilan Citra
Citra_1.jpg		Citra_5.jpg	
Citra_2.jpg		Citra_6.jpg	

Nama Citra	Tampilan Citra	Nama Citra	Tampilan Citra
Citra_3.jpg		Citra_7.jpg	
Citra_4.jpg			

Berdasarkan pengujian threshold dengan persentasi noise 5%-20% pada semua citra didapatkan nilai threshold terbaik yaitu 192 pada citra\_1.jpg dan citra\_2.jpg seperti terlihat pada rangkuman tabel 2.

Tabel 2 Kumpulan Nilai Threshold Terbaik untuk Citra Pengujian

No	Nama Citra	Persen Noise	Nilai <i>Threshold</i>	Nilai MSE	Nilai PSNR
1	Citra_1.jpg	5	192	60.78	39.21
2	Citra_1.jpg	10	192	63.86	39.10
3	Citra_1.jpg	15	192	74.43	38.77
4	Citra_1.jpg	20	192	68.1	38.90
5	Citra_2.jpg	5	192	82.39	38.55
6	Citra_2.jpg	10	192	90.80	38.35
7	Citra_2.jpg	15	192	99.21	38.15
8	Citra_2.jpg	20	192	114.19	37.84

Setelah pengujian terhadap semua citra mulai dari noise 5% maka didapatkan citra hasil dengan denoising paling buruk berdasarkan nilai seperti terlihat pada tabel 3 di bawah.

Tabel 3 Nilai PSNR untuk Hasil Denoising Terburuk

No	Nama Citra	Persen Noise	Nilai <i>Threshold</i>	Nilai MSE	Nilai PSNR
1	Citra_1.jpg	55	192	4364.32	29.93
2	Citra_2.jpg	55	192	4242.69	29.99
3	Citra_3.jpg	57	192	4387.18	29.92
4	Citra_4.jpg	52	192	4288.92	29.97
5	Citra_5.jpg	55	192	4239.1	29.99
6	Citra_6.jpg	58	192	4405.19	29.91
7	Citra_7.jpg	56	192	4321.12	29.98

Dengan proses yang sama untuk mendapatkan hasil pada tabel 3, juga didapatkan hasil *denoising* terbaik seperti terlihat pada tabel 4 di bawah.

Tabel 4 Nilai PSNR untuk Hasil Denosing Terbaik

No	Persen <i>noise</i>	Citra 'Uji 1'	Citra 'Uji 2'	Citra 'Uji 3'	Citra 'Uji 4'	Citra 'Uji 5'	Citra 'Uji 6'	Citra 'Uji 7'
1	5	39.21	38.49	36.37	37.55	37.88	37.26	36.97
2	10	39.07	38.32	36.06	37.19	37.56	37.07	36.59
3	15	38.68	38.12	35.81	36.83	37.25	36.74	36.4

Hasil pengujian terhadap citra *denoising noise* dengan variasi persentasi noise dapat ditampilkan dalam *screenshot* gambar sebagai berikut:

Tabel 5 Hasil *denoising noise* pada beberapa citra uji

Nama Citra (UkuranNoise)	Citra Noise	Citra Hasil
Citra_1.jpg (10%)		
Citra_4.jpg (15%)		
Citra_5.jpg (20%)		
Citra_6.jpg (25%)		

Berdasarkan hasil pengujian yang telah dilakukan, maka diperoleh beberapa informasi hasil pengujian seperti berikut ini:

1. Parameter yang digunakan (*threshold*) yang terlalu tinggi (201-255) akan menyebabkan *noise* masih bersisa di citra sedangkan jika terlalu kecil (10-160) maka akan melakukan iterasi terus menerus walaupun sudah tidak diperlukan dan malah memperburuk nilai PSNR
2. Perulangan sudah otomatis didapatkan dari hasil perbandingan MAG yang didapatkan jika hasil perhitungan MAG lebih besar dari *threshold* yang ditentukan maka akan mengalami perulangan jika sudah lebih kecil maka citra sudah dianggap bersih dari *noise* dan tidak akan diulang lagi
3. Proses *denoising* dengan metode *iterative denoising and backward projection* yang nilai PSNR lebih dari 30dB dari hasil pengujian hanya sampai 56% dengan *threshold* 192.
4. Proses *denoising* dengan metode *iterative denoising and backward projection* yang nilai PSNR paling bagus adalah di noise 5% dengan PSNR 39.21dB pada gambar\_1.jpg
5. Proses *denoising* dengan metode *iterative denoising and backward projection* yang nilai PSNR paling buruk adalah di noise 58% dengan PSNR 29.91dB pada gambar\_6.jpg.
6. Nilai parameter yang mempengaruhi citra hanya *threshold* sehingga akan lebih sedikit mempengaruhi kualitas citra hasil.

#### 4. KESIMPULAN

Dari hasil pengujian yang sudah dilakukan sebelumnya, maka dapat diambil beberapa kesimpulan yang diantaranya sebagai berikut:

1. Maksimal *noise* yang bisa di *denoising* adalah sampai dengan 51% noise pada citra menggunakan pengujian PSNR dengan nilai minimum 30.32 dB.
2. Proses *denoising* dengan metode *iterative denoising and backward projection* yang nilai PSNR paling bagus adalah di noise 5% dengan PSNR 39.21 pada gambar Uji 1 dan yang paling buruk adalah di noise 58% dengan PSNR 29.91 pada gambar Uji 6.
3. Parameter yang digunakan (*threshold*) yang terlalu tinggi (201-255) akan menyebabkan *noise* masih bersisa di citra sedangkan jika terlalu kecil (10-160) maka akan melakukan iterasi terus menerus walaupun sudah tidak diperlukan dan malah memperburuk nilai PSNR.
4. Penerapan metode *Iterative Denoising and Backward Projections* hanya menggunakan 1 parameter yaitu *Threshold*, sehingga lebih memudahkan untuk mendapatkan hasil citra yang paling optimal.

#### 5. SARAN

Beberapa saran yang dapat dikembangkan untuk meningkatkan kualitas dan untuk penelitian lanjutan diantaranya:

1. Perlu pengujian metode *Iterative Denoising and Backward Projections* dengan *noise* yang lain seperti *Gaussian noise* atau *inpainting* dan *deblurring* dan membandingkan dengan algoritma lain dengan kualitas persentasi noise diatas 51% dengan PSNR diatas 30dB
2. Sistem pengujian nilai *threshold* perlu diuji secara otomatis menggunakan metode terukur sehingga tidak lagi menggunakan parameter yang ditentukan sendiri dan bisa menghasilkan kualitas gambar yang lebih baik

#### DAFTAR PUSTAKA

- [1] I. Irviantina, Syanti; Pardosi, "Salt and Pepper Noise Removal dengan Spatial Median Filter dan Adaptive Noise Reduction," *JSM (Jurnal SIFO Mikroskil)*, vol. 17, no. 2, pp. 127–136, 2016.
- [2] P. Sirait, I. A. Pardosi, and A. Halim, "Implementasi Kombinasi Metode AFF dan FBET Untuk Peningkatan Kualitas Citra," vol. 20, no. 1, pp. 1–10, 2019.
- [3] T. Gebreyohannes and D. Kim, "A DAPTIVE NOISE REDUCTION SCHEME FOR SALT AND PEPPER."
- [4] I. A. Pardosi and A. A. Lubis, "Analisis Kualitas Citra Hasil Reduksi Noise Menggunakan Spatial Median Filter dan Adaptive Fuzzy Filter Terhadap Variasi Kedalaman Citra," *Indones. J. Inf. Syst.*, vol. 1, no. 2, p. 78, 2019, doi: 10.24002/ijis.v1i2.1939.

- [5] H. Sajati, “Analisis kualitas perbaikan citra menggunakan metode median filter dengan penyeleksian nilai Pixel,” *J. Ilm. Bid. Teknol.*, vol. 10, pp. 41–48, 2018.
- [6] T. Tիրer and R. Giryes, “Image Restoration by Iterative Denoising and Backward Projections,” *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1220–1234, 2019, doi: 10.1109/TIP.2018.2875569.
- [7] S. V. Venkatakrisnan, C. A. Bouman, and B. Wohlberg, “Plug-and-Play priors for model based reconstruction,” *2013 IEEE Glob. Conf. Signal Inf. Process. Glob. 2013 - Proc.*, pp. 945–948, 2013, doi: 10.1109/GlobalSIP.2013.6737048.
- [8] W. S. Eka Putra, “Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101,” *J. Tek. ITS*, vol. 5, no. 1, 2016, doi: 10.12962/j23373539.v5i1.15696.

