



Penerapan Image Processing Menggunakan OpenCV Dan Python Untuk Mengurangi Noise Pada Gambar Digital Dengan Metode Median Filter

Ayyash Umar Kholid^{1*}, Muhammad Farhan², Satria Agung Wibowo³, Perani Rosyani⁴

^{1,2,3,4}Fakultas Ilmu Komputer, Program Studi Teknik Informatika, Universitas Pamulang, Tangerang Selatan, Indonesia

Email: ^{1*}ayyashumarko@gmail.com, ²frhnapet@gmail.com, ³aryotamvan@gmail.com,

⁴dosen00837@unpam.ac.id

(* : coresponding author)

Abstrak – Penelitian ini bertujuan untuk mengurangi noise pada gambar digital menggunakan metode median filter dengan bantuan pustaka OpenCV dan bahasa pemrograman Python. Noise pada gambar digital dapat mempengaruhi kualitas visual dan mengganggu analisis citra. Metode median filter dipilih karena kemampuannya dalam mempertahankan tepi gambar sambil menghilangkan noise. Hasil eksperimen menunjukkan bahwa penerapan median filter secara signifikan meningkatkan kualitas gambar dengan mengurangi noise tanpa mengorbankan detail penting.

Kata Kunci: Pemrosesan Citra; OpenCV; Python; Median Filter; Noise Reduction; Gambar Digital

Abstract – This research aims to reduce noise in digital images using the median filter method with the help of the OpenCV library and the Python programming language. Noise in digital images can affect visual quality and interfere with image analysis. The median filter method was chosen because of its ability to preserve image edges while removing noise. Experimental results show that applying a median filter significantly improves image quality by reducing noise without sacrificing important details.

Keywords: Image Processing; OpenCV; Python; Median Filter; Noise Reduction; Digital Images

1. PENDAHULUAN

Dalam era digital saat ini, gambar digital telah menjadi bagian integral dari kehidupan sehari-hari. Namun, proses pengambilan gambar digital sering kali menghasilkan gambar yang terpengaruh oleh noise, yang dapat mengurangi kualitas gambar dan mengganggu proses analisis lebih lanjut. Oleh karena itu, pengembangan metode untuk mengurangi noise pada gambar digital merupakan topik yang penting dalam bidang pengolahan citra. Menurut (Fadillah, N., & Gunawan, C. R., 2019). Untuk meningkatkan kualitas citra dan mengurangi noise agar informasi yang ada pada citra dapat dipahami dengan baik dibutuhkan metode filtering.

2. METODE

Pada penelitian ini, kami menggunakan bahasa pemrograman Python dan library OpenCV untuk mengimplementasikan Median Filter.

Langkah-langkah yang dilakukan adalah sebagai berikut:

2.1 Membaca Gambar Digital

Pertama-tama, gambar digital dibaca menggunakan fungsi `cv2.imread()` dari library OpenCV. Gambar dibaca dalam format BGR (Blue, Green, Red).

Berikut adalah implementasinya :



```
import cv2
# Membaca gambar digital
image = cv2.imread('Hutan.jpg')
# Menampilkan gambar asli
cv2.imshow('Original Image',
image)
cv2.waitKey(0)
# Menutup semua jendela
cv2.destroyAllWindows()
```

Penjelasan:

- import cv2: Mengimpor library OpenCV yang digunakan untuk pemrosesan citra.
- cv2.imread('path_to_image.jpg'): Membaca gambar dari file yang ditentukan. Fungsi cv2.imread() mengembalikan gambar dalam format BGR (Blue, Green, Red). Variabel image menyimpan data gambar yang telah dibaca.
- Pembacaan Gambar:** Gambar dibaca dalam format BGR yang merupakan format default OpenCV. Format ini berbeda dengan format RGB yang biasa digunakan dalam banyak aplikasi.

2.2 Menerapkan Median Filter

Setelah gambar dibaca, Median Filter diterapkan pada gambar menggunakan fungsi `cv2.medianBlur()` dari OpenCV. Median Filter bekerja dengan mengambil nilai median dari piksel-piksel dalam jendela kernel, dan nilai median tersebut diterapkan pada piksel tengah. Proses ini dilakukan untuk setiap piksel dalam gambar.

Berikut adalah implementasinya :

```
import cv2
# Membaca gambar digital
image =
cv2.imread('Hutan.jpg')
# Menerapkan Median
Filter
median_filtered_image =
cv2.medianBlur(image,
5) # Menggunakan kernel
berukuran 5x5
# Menampilkan gambar
hasil filtering
cv2.imshow('Filtered
Image',
median_filtered_image)
cv2.waitKey(0)
# Menutup semua jendela
```



Penjelasan:

- `cv2.medianBlur(image, ksize=5)`: Menerapkan median filter pada gambar. Fungsi `cv2.medianBlur()` membutuhkan dua parameter:
- `image`: Gambar yang akan difilter.
- `ksize`: Ukuran kernel (jendela) yang digunakan dalam filter. Ukuran ini harus berupa angka ganjil (misalnya, 3, 5, 7). Pada contoh ini, ukuran kernel adalah 5.
- Fungsi ini bekerja dengan menggantikan setiap piksel dengan nilai median dari piksel-piksel dalam jendela kernel yang mengelilinginya. Ini membantu menghilangkan noise, terutama noise tipe salt-and-pepper, sambil mempertahankan tepi gambar.
- Penyimpanan Gambar**: Gambar yang telah difilter disimpan ke file sehingga hasil filtering dapat digunakan untuk keperluan lebih lanjut atau dibandingkan dengan gambar asli untuk analisis efektivitas metode.

2.3 Menyimpan Gambar Hasil Filtering

Setelah proses filtering selesai, gambar hasil filtering disimpan menggunakan fungsi `cv2.imwrite()` dari OpenCV. Gambar disimpan dalam format yang ditentukan (misalnya, format JPEG atau PNG) dengan nama file yang sesuai.

Metode ini memungkinkan untuk mengurangi noise pada gambar digital tanpa mengorbankan detail penting. Median Filter dipilih karena efektivitasnya dalam mereduksi noise, terutama noise salt and pepper, sambil mempertahankan detail gambar yang signifikan. Implementasi menggunakan OpenCV mempermudah proses pengolahan gambar dengan dukungan yang luas dan efisiensi komputasi yang tinggi.

Berikut adalah implementasinya :

```
import cv2
# Membaca gambar digital
image = cv2.imread('Hutan.jpg')
# Menerapkan Median Filter
median_filtered_image = cv2.medianBlur(image, 5) #
Menggunakan kernel berukuran 5x5
# Menyimpan gambar hasil filtering
cv2.imwrite('Hutan_output.jpg',
median_filtered_image)
print("Gambar hasil filtering telah disimpan sebagai
'gambar_output.jpg'.")
```

Penjelasan:

- `cv2.imwrite('filtered_image.jpg', filtered_image)`: Menyimpan gambar hasil filtering ke dalam file dengan nama `filtered_image.jpg`. Fungsi `cv2.imwrite()` membutuhkan dua parameter:
- `filename`: Nama file tempat gambar akan disimpan.
- `image`: Gambar yang akan disimpan. Pada contoh ini, gambar yang disimpan adalah `filtered_image`, yaitu gambar yang telah difilter menggunakan median filter.
- Penyimpanan Gambar**: Gambar yang telah difilter disimpan ke file sehingga hasil filtering dapat digunakan untuk keperluan lebih lanjut atau dibandingkan dengan gambar asli untuk analisis efektivitas metode.



3. ANALISA DAN PEMBAHASAN

3.1 Noise

Noise (Derau) adalah gambar atau *pixel* yang mengganggu kualitas citra. Derau dapat disebabkan oleh gangguan fisik(optik) pada alat akuisisi maupun secara disengaja akibat proses pengolahan yang tidak sesuai. Contohnya adalah bintik hitam atau putih yang muncul secara acak yang tidak diinginkan di dalam citra. bintik acak ini disebut dengan derau salt & pepper (Sari, S. , 2019).

3.2 Opencv

OpenCV (*Open Source Computer Vision Library*) adalah sebuah *library* yang digunakan untuk mengolah gambar dan video sehingga pengguna dapat meng-ekstrak informasi dari sebuah citra. (Andrekhya, M. Z. , & Huda, Y. , 2021). Dengan adanya OpenCV mendukung berbagai bahasa pemrograman seperti C, C++, Python, Java, dan PHP. Beberapa aplikasi OpenCV yang bisa diterapkan diantaranya adalah Interaksi Manusia Komputer (HCI), Identifikasi dan Pengenalan Objek, Deteksi [11] dan Pengenalan Wajah [12], Pengenalan Gestur, Tracking Gerakan, Proses pada image, dan Mobile Robotics. (Santoso, B. & Kristianto. R. A. , 2020).

OpenCV dikembangkan dengan menggunakan bahasa pemrograman C++ dan dapat digunakan dengan berbagai bahasa pemrograman lain seperti Python dan Java melalui pengikatan (bindings) yang tersedia (Jonathan, M., Hafidz, M. T., Apriyanti, N. A., Husaini, Z., Rosyani, P. , 2023).

3.3 Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile (Romzi, M. & Kurniawan, B. , 2020).

3.4 Median Filter

Metode *Median Filtering* merupakan filter *non-linear* yang dikembangkan oleh Tukey. Metode tersebut berfungsi untuk mengurangi derau dan menghaluskan citra digital. Dikatakan *non-linear* karena karakterja *window* atau penapis ini tidak termasuk kedalam kategori operasi konvolusi. Operasi *non-linear* dihitung dengan cara mengurutkan nilai sekelompok piksel atau nilai ketetanggaan, kemudian mengganti nilai piksel yang diproses dengan nilai tengah atau median dari seluruh ketetanggaannya (Yasir, A. , Satria, W. , Yuanda, P. , 2023). Median filter mampu mengurangi beberapa type derau (noise) acak dengan mempertimbangkan efek blur yang sedikit disbanding filter linier smoothing (Prastya. D. Z. E., Pamungkas, D. P., Niswatin, R. K. , 2022).

Pada tahap ini, kami melakukan analisis terhadap hasil eksperimen yang dilakukan. Beberapa hasil eksperimen yang signifikan akan dibahas secara rinci.

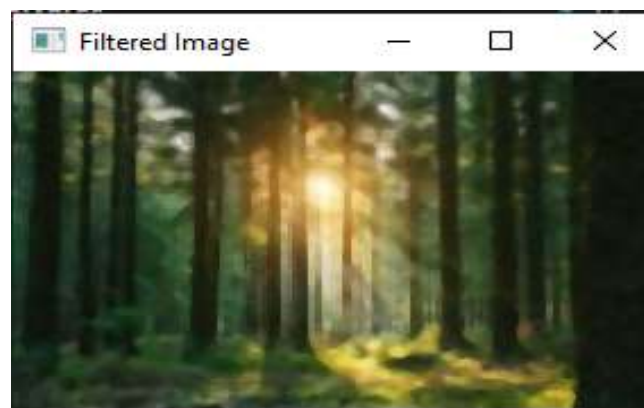
Hasil Eksperimen :

Berdasarkan hasil eksperimen di atas, kami dapat menyimpulkan bahwa metode Median Filter cukup efektif dalam mengurangi noise pada gambar digital. Namun, tingkat keberhasilannya dapat dipengaruhi oleh tingkat noise awal pada gambar dan kompleksitas detail gambar. Selain itu, parameter-parameter seperti ukuran kernel Median Filter juga dapat mempengaruhi kinerja metode ini.



Gambar 1. Gambar Hutan sebelum di Filter

Fungsi: Gambar ini menunjukkan kondisi asli gambar digital sebelum dilakukan proses filtering. Ini bertujuan untuk memberikan gambaran tentang tingkat noise atau gangguan yang ada pada gambar asli sebelum penerapan metode median filter. Gambar ini penting untuk dibandingkan dengan hasil setelah filtering untuk menunjukkan efektivitas metode yang digunakan.



Gambar 2. Gambar Hutan setelah difilter

Fungsi: Gambar ini menunjukkan hasil gambar setelah diterapkan metode median filter menggunakan OpenCV dan Python. Tujuannya adalah untuk memperlihatkan perubahan yang terjadi pada gambar setelah noise dihilangkan. Perbandingan antara gambar ini dan gambar sebelumnya membantu pembaca memahami seberapa baik metode median filter dalam mengurangi noise tanpa mengorbankan detail penting pada gambar.



Gambar 3. Gambar Hutan Yang Disimpan Setelah Difilter



Fungsi: Gambar ini merupakan hasil akhir yang telah disimpan setelah proses filtering dilakukan. Ini menunjukkan bahwa gambar hasil filtering dapat disimpan dan diakses untuk keperluan selanjutnya, seperti analisis lebih lanjut atau penggunaan dalam aplikasi lain. Menyimpan gambar hasil filtering adalah langkah penting untuk menunjukkan bahwa metode yang diterapkan tidak hanya berfungsi dalam lingkungan pengolahan citra tetapi juga menghasilkan output yang dapat digunakan secara praktis.

Pembahasan

Berdasarkan hasil eksperimen di atas, kami dapat menyimpulkan bahwa metode Median Filter cukup efektif dalam mengurangi noise pada gambar digital. Namun, tingkat keberhasilannya dapat dipengaruhi oleh tingkat noise awal pada gambar dan kompleksitas detail gambar. Selain itu, parameter-parameter seperti ukuran kernel Median Filter juga dapat mempengaruhi kinerja metode ini.

4. Kesimpulan

Dalam penelitian ini, kami berhasil menerapkan metode Median Filter untuk mengurangi noise pada gambar digital dengan menggunakan OpenCV dan Python. Hasil eksperimen menunjukkan bahwa metode ini efektif dalam mengurangi noise pada berbagai tingkat noise awal gambar. Meskipun demikian, perlu diingat bahwa ada beberapa kompromi yang mungkin perlu dilakukan antara pengurangan noise dan mempertahankan detail gambar. Sebagai kesimpulan, metode Median Filter dapat menjadi pilihan yang baik untuk mengurangi noise pada gambar digital dengan tingkat keberhasilan yang baik.

REFERENCES

- Fadillah, N., & Gunawan, C. R. (2019). Mendeteksi Keakuratan Metode Noise Salt and Pepper dengan Median Filter. *Jurnal Informatika*, 6(1), 91-95.
- Sari, S. (2019). Penerapan Metode Median Filter untuk Mereduksi Noise Speckle dan Salt & Pepper pada Citra Ortokromatik. *Building of Informatics, Technology and Science (BITS)*, 1(1), 34-41.
- Andrekhya, M. Z., & Huda, Y. (2021). Deteksi Warna Manggis Menggunakan Pengolahan Citra dengan Opencv Python. *Jurnal Vocational Teknik Elektronika dan Informatika*, 9(4).
- Santoso, B., & Kristianto, R. P. (2020). Implementasi Penggunaan OpenCV pada Face Recognition untuk Sistem Presensi Perkuliahan Mahasiswa. *Universitas AMIKOM Yogyakarta*.
- Jonathan, M., Hafidz, M. T., Apriyanti, N. A., Husaini, Z., & Rosyani, P. (2023). Mendeteksi Plat Nomor Kendaraan dengan Metode YOLO (You Only Look Once) dan Single Shot Detector (SSD). *Jurnal AI dan SPK: Jurnal Artificial Intelligent dan Sistem Penunjang Keputusan*, 1(1), 105-111.
- Romzi, M., & Kurniawan, B. (2020). Pembelajaran Pemrograman Python dengan Pendekatan Logika Algoritma. *JTIM: Jurnal Teknik Informatika Mahakarya*, 3(2), 37-44.
- Yasir, A., Satria, W., & Yuanda, P. (2023). Digital Image Processing Metode Median Filtering dan Morfologi Opening dalam Reduksi Noise Citra. *Volume 17, Nomor 4*, 1687-1701.
- Prastya, D. Z. E., Pamungkas, D. P., & Niswatin, R. K. (2022). Implementasi Metode Gaussian Filter dan Median Filter untuk Penghalusan Gambar. *Seminar Nasional Inovasi Teknologi*, 178.