

PEMBUATAN APLIKASI PERBAIKAN KUALITAS CITRA PADA DOMAIN SPASIAL MEGGUNAKAN PYTHON

LAPORAN TUGAS

Disusun untuk memenuhi persyaratan Mata Kuliah Pengolahan Citra sebagai Pengerjaan
Quiz Program Studi S1 Teknologi Informasi

Oleh:

CINDY KEVINA	212310021
EMANUEL CHAREL ALESSANDO SOGE	212310041
RIZKI RAMDANI	212310059



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS INFORMATIKA DAN PARIWISATA
INSTITUT BISNIS DAN INFORMATIKA KESATUAN**

BOGOR

2024

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	3
PENDAHULUAN	3
1.1. Latar Belakang	3
1.2. Tujuan.....	3
1.3. Ruang Lingkup	4
1.4. Manfaat.....	4
BAB II.....	6
ALAT DAN BAHAN	6
2.1. Alat	6
2.2. Bahan.....	6
BAB III	7
METODE PEGEMBANGAN	7
3.1. Desain Aplikasi	7
3.2. Langkah – Langkah Pembuatan Aplikasi.....	7
BAB IV	9
HASIL DAN PEMBAHASAN.....	9
4.1. Antarmuka Pengguna	9
4.2. Hasil Pemrosesan Gambar.....	9
4.3. Performa Aplikasi.....	10
BAB V.....	11
KESIMPULAN DAN SARAN.....	11
DAFTAR PUSTAKA	13

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pengolahan citra digital telah menjadi salah satu bidang yang penting dalam berbagai aplikasi teknologi, mulai dari fotografi, pengawasan keamanan, hingga teknologi medis. Citra digital sering kali mengalami degradasi kualitas karena faktor-faktor seperti gangguan lingkungan atau kondisi teknis, yang menyebabkan munculnya noise atau distorsi. Salah satu metode yang digunakan untuk memperbaiki kualitas citra yang mengalami gangguan tersebut adalah dengan menerapkan filter dalam domain spasial, seperti filter median dan filter rata-rata.

Dalam konteks pengolahan citra, noise adalah salah satu masalah utama yang mengganggu kualitas gambar dan mengurangi tingkat keterbacaan informasi yang terkandung di dalamnya. Noise dapat muncul karena berbagai sebab, seperti kesalahan sensor kamera, kondisi pencahayaan yang buruk, atau gangguan pada saat transmisi citra. Filter median dan filter rata-rata adalah dua metode yang sering digunakan untuk mengurangi noise. Filter median cenderung efektif dalam mengurangi noise tipe salt-and-pepper, sementara filter rata-rata lebih cocok untuk mereduksi noise Gaussian. Aplikasi yang dikembangkan dalam proyek ini bertujuan untuk memudahkan pengguna, baik dari kalangan akademisi maupun praktisi, dalam memproses citra dan memperbaiki kualitasnya dengan menerapkan filter median dan rata-rata. Aplikasi ini dirancang berbasis Python dengan antarmuka grafis yang interaktif, sehingga pengguna dapat dengan mudah membuka gambar, memilih filter, mengatur parameter kernel, melihat hasil perbaikan secara langsung, dan menyimpan gambar hasil pemrosesan.

1.2. Tujuan

Tujuan utama dari aplikasi ini adalah untuk memberikan alat bantu yang user-friendly bagi pengguna dalam melakukan proses perbaikan kualitas citra yang mengalami noise. Secara lebih spesifik, tujuan yang ingin dicapai melalui pengembangan aplikasi ini meliputi:

1. Menyediakan antarmuka yang intuitif bagi pengguna untuk membuka dan menampilkan gambar.

2. Mengimplementasikan filter median dan filter rata-rata dengan opsi pengaturan ukuran kernel.
3. Menampilkan hasil perbaikan citra secara real-time, memungkinkan pengguna untuk membandingkan gambar asli dan hasil yang telah diproses.
4. Menyediakan fitur penyimpanan gambar hasil proses ke dalam perangkat lokal.
5. Memvisualisasikan perubahan distribusi piksel dalam bentuk histogram sebelum dan sesudah pemrosesan untuk memberikan gambaran visual dampak dari penerapan filter.

1.3. Ruang Lingkup

Pengembangan aplikasi ini difokuskan pada penerapan dua jenis filter untuk memperbaiki citra yang mengalami noise, yaitu:

1. **Filter Median:** Menggunakan ukuran kernel yang dapat diatur oleh pengguna. Filter ini bekerja dengan mengganti setiap piksel dengan nilai median dari piksel-piksel di sekitarnya dalam area kernel yang telah ditentukan. Filter ini sangat efektif untuk menghilangkan noise salt-and-pepper.
2. **Filter Rata-Rata (Average Filter):** Filter ini bekerja dengan mengganti nilai setiap piksel dengan rata-rata dari nilai piksel di sekitarnya dalam area kernel yang dipilih. Filter ini membantu mereduksi noise tipe Gaussian, meskipun juga dapat menyebabkan sedikit efek blur.

Aplikasi ini juga dilengkapi dengan fitur visualisasi histogram untuk memberikan wawasan lebih mendalam mengenai distribusi intensitas piksel pada citra asli dan citra yang telah diproses. Selain itu, aplikasi ini dirancang agar mudah digunakan oleh siapa saja tanpa memerlukan pengetahuan teknis yang mendalam mengenai pengolahan citra.

1.4. Manfaat

Beberapa manfaat yang diharapkan dari aplikasi ini adalah:

1. **Kemudahan Penggunaan:** Aplikasi ini menawarkan antarmuka grafis yang sederhana namun fungsional, memungkinkan pengguna untuk memproses gambar tanpa memerlukan keahlian teknis dalam pemrograman atau pengolahan citra.

2. **Penerapan Algoritma yang Efektif:** Algoritma filter median dan filter rata – rata yang diimplementasikan pada aplikasi ini dapat secara efektif mengurangi noise pada citra, membantu meningkatkan kualitas visual gambar.
3. **Peningkatan Kualitas Citra:** Aplikasi ini bermanfaat bagi kalangan profesional maupun mahasiswa yang membutuhkan alat untuk memperbaiki gambar yang rusak atau terdistorsi akibat noise.
4. **Pembelajaran dan Eksperimen:** Aplikasi ini juga bisa digunakan sebagai alat bantu pembelajaran di bidang pengolahan citra, memungkinkan pengguna untuk bereksperimen dengan berbagai parameter kernel untuk melihat pengaruhnya terhadap hasil gambar.

BAB II

ALAT DAN BAHAN

Untuk mengembangkan aplikasi ini, beberapa alat dan bahan yang digunakan adalah sebagai berikut:

2.1. Alat

- **Bahasa pemrograman:** Python 3.x
- **Library:**
 1. OpenCV: Untuk pemrosesan gambar.
 2. Pillow: Untuk konversi citra menjadi format yang bisa ditampilkan
 3. Matplotlib: Untuk visualisasi histogram.
 4. CustomTkinter: Untuk pembuatan antarmuka grafis (GUI).
- **Sistem Operasi:** Windows/Linux
- **IDE:** Visual Studio Code atau PyCharm
- **Perangkat Keras:** Komputer atau laptop dengan RAM minimal 4GB dan prosesor 2GHz ke atas.

2.2. Bahan

Dataset Gambar: Gambar berformat JPEG atau PNG yang akan diproses oleh aplikasi.

BAB III

METODE PEGEMBANGAN

3.1. Desain Aplikasi

Aplikasi ini dirancang dengan menggunakan arsitektur berbasis GUI (Graphical User Interface), yang diimplementasikan dengan menggunakan library `customtkinter`. Desain aplikasi mengutamakan kemudahan penggunaan, sehingga pengguna dapat dengan mudah memilih gambar, menerapkan filter, dan menyimpan hasilnya.

3.2. Langkah – Langkah Pembuatan Aplikasi


Berikut adalah langkah – langkah dalam pembuatan aplikasi ini:

Langkah 1: Persiapan Lingkungan pengembangan

1. **Install Python 3.x** jika belum terpasang.
2. **Install library yang dibutuhkan** menggunakan pip:

```
bash

pip install opencv-python
pip install Pillow
pip install matplotlib
pip install customtkinter
```

 Copy code

Langkah 2: Membuat Antarmuka Aplikasi

Antarmuka aplikasi dibuat menggunakan `customtkinter`. Aplikasi ini terdiri dari beberapa komponen utama, yaitu:

1. **Menu Bar:** Untuk membuka dan menyimpan gambar.
2. **Frame untuk Menampilkan Gambar:** Terdapat dua area untuk menampilkan gambar asli dan gambar yang telah diproses.
3. **Kontrol Filter:** Slider dan tombol untuk mengatur ukuran kernel filter dan menerapkan filter median atau rata – rata.

Langkah 3: Implementasi Fungsi Pembukaan Gambar

Menggunakan dialog file `filedialog.askopenfilename()` untuk memungkinkan pengguna memilih gambar yang akan diproses. Setelah gambar dipilih, gambar tersebut ditampilkan di dalam aplikasi menggunakan `Pillow` dan `OpenCV`.

Langkah 4: Implementasi Filter Median

Filter median diimplementasikan menggunakan fungsi **cv2.medianBlur()** dari **OpenCV**. Pengguna dapat mengatur ukuran kernel melalui slider yang telah disediakan. Gambar hasil filter kemudian ditampilkan di area gambar yang diproses.

Langkah 5: Implementasi Filter Rata-Rata

Filter rata-rata diimplementasikan menggunakan fungsi **cv2.blur()** dari **OpenCV**. Sama seperti filter median, pengguna dapat mengatur ukuran kernel menggunakan slider. Setelah filter diterapkan, gambar yang diproses ditampilkan di layar.

Langkah 6: Visualisasi Histogram

Fungsi histogram diimplementasikan menggunakan **matplotlib**. Histogram ini memberikan gambaran distribusi piksel pada gambar asli dan gambar yang diproses, membantu pengguna memahami perubahan yang terjadi setelah penerapan filter.

Langkah 7: Menyimpan Gambar

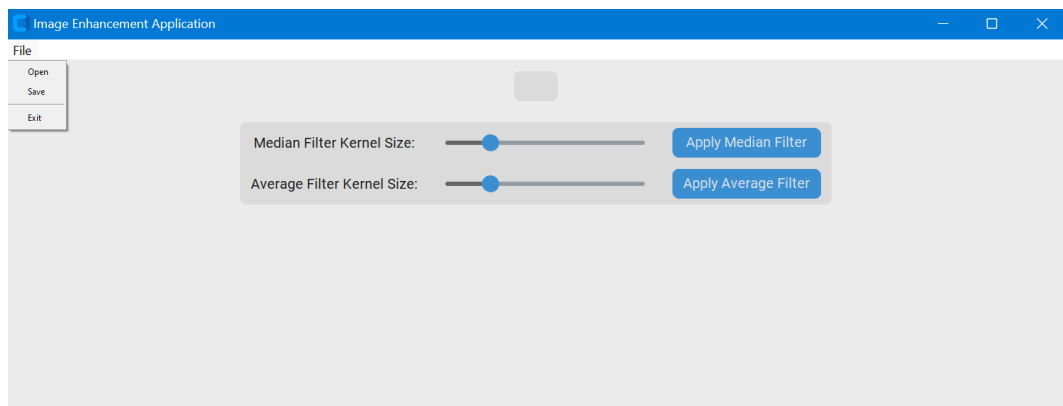
Fungsi untuk menyimpan gambar yang telah diproses menggunakan **filedialog.asksaveasfilename()**. Gambar disimpan dalam format PNG atau JPEG sesuai pilihan pengguna.

BAB IV

HASIL DAN PEMBAHASAN

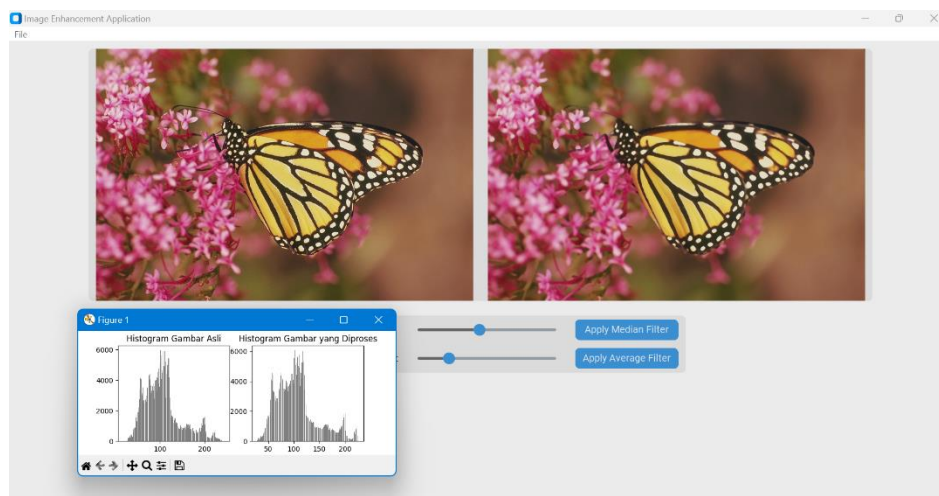
4.1. Antarmuka Pengguna

Antarmuka pengguna yang dibuat menggunakan **customtkinter** memberikan pengalaman yang sederhana dan intuitif. Pengguna dapat dengan mudah membuka gambar, menerapkan filter, melihat histogram, dan menyimpan gambar hasil proses.



4.2. Hasil Pemrosesan Gambar

Dengan menerapkan filter median dan filter rata-rata, aplikasi ini mampu memperbaiki citra yang mengalami noise. Hasilnya terlihat pada gambar yang lebih halus dan noise yang berkurang. Filter median sangat efektif dalam mengatasi noise salt-and-pepper, sedangkan filter rata-rata efektif dalam mengurangi noise Gaussian, meskipun menyebabkan gambar sedikit blur.



4.3. Performa Aplikasi

Aplikasi berjalan dengan baik pada gambar dengan ukuran yang sedang (hingga 1920x1080 piksel). Namun, jika gambar berukuran sangat besar, proses penerapan filter mungkin memakan waktu lebih lama. Pengaturan ukuran kernel yang besar juga dapat memperlambat proses pemrosesan.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Aplikasi yang dikembangkan berhasil memenuhi tujuan utama yaitu memberikan alat bantu bagi pengguna dalam memperbaiki citra yang mengalami noise. Aplikasi ini memiliki fitur yang lengkap, mulai dari pembukaan gambar, penerapan filter, hingga penyimpanan gambar yang diproses. Visualisasi histogram juga memberikan insight yang berharga mengenai perubahan distribusi piksel setelah pemrosesan. Aplikasi ini sangat bermanfaat bagi pengguna yang ingin melakukan perbaikan kualitas citra secara cepat dan mudah, tanpa harus mempelajari algoritma pengolahan citra yang kompleks.

Pendekatan yang digunakan dalam aplikasi perbaikan kualitas citra ini adalah dengan menerapkan dua jenis filter spasial, yaitu **filter median** dan **filter rata-rata (average filter)**. Filter median dipilih karena kemampuannya yang sangat efektif dalam menghilangkan noise salt-and-pepper, di mana nilai piksel ekstrim dapat diabaikan dengan mengganti piksel dengan nilai median dari area tetangganya. Sementara itu, filter rata-rata dipilih karena cocok untuk mereduksi noise Gaussian dengan menggantikan setiap piksel dengan rata-rata nilai tetangga di sekitarnya, meskipun memiliki kelemahan dalam menyebabkan gambar menjadi sedikit kabur.

Hasil dari penerapan filter median menunjukkan perbaikan yang signifikan pada citra yang mengalami noise salt-and-pepper. Noise yang ada pada gambar berhasil dihilangkan tanpa mengurangi detail gambar secara drastis. Di sisi lain, penerapan filter rata-rata juga berhasil mengurangi noise, namun efek blur lebih jelas terlihat, terutama pada gambar dengan banyak detail.

5.2. Saran

Beberapa saran pengembangan yang dapat diterapkan untuk meningkatkan fungsi dan performa aplikasi ini meliputi:

1. Penambahan Filter Lain: Aplikasi dapat dikembangkan lebih lanjut dengan menambahkan jenis filter lain seperti Gaussian blur, sharpen, atau edge detection untuk memperluas fungsi aplikasi.

2. Optimasi Proses Pemrosesan: Meningkatkan performa aplikasi untuk dapat memproses gambar berukuran besar dengan lebih cepat, seperti dengan menggunakan metode pemrograman paralel atau optimasi algoritma.
3. Peningkatan Antarmuka Pengguna: Menambahkan fitur zoom atau tampilan layar penuh untuk memperbaiki pengalaman pengguna saat bekerja dengan gambar berukuran besar.
4. Dukungan Format Gambar Tambahan: Memperluas dukungan format gambar untuk mencakup lebih banyak jenis file gambar seperti TIFF, BMP, atau RAW.
5. Implementasi Teknologi Cloud: Mengintegrasikan aplikasi dengan teknologi cloud untuk pemrosesan citra skala besar atau aplikasi yang membutuhkan penyimpanan dan komputasi lebih lanjut.

Dengan pengembangan lebih lanjut, aplikasi ini dapat digunakan secara lebih luas dalam berbagai skenario pengolahan citra, baik untuk penelitian, pendidikan, maupun aplikasi komersial.

DAFTAR PUSTAKA

1. The OpenCV Foundation. (2023). *OpenCV Documentation*. Retrieved from <https://docs.opencv.org>
2. Lundh, F. (2023). *Pillow (PIL Fork) Documentation*. Python Imaging Library (PIL). Retrieved from <https://python-pillow.org>
3. Matplotlib Developers. (2023). *Matplotlib Documentation*. Retrieved from <https://matplotlib.org>
4. CustomTkinter Team. (2023). *CustomTkinter Documentation*. Retrieved from <https://github.com/TomSchimansky/CustomTkinter>
5. Python Software Foundation. (2023). *Python Documentation*. Retrieved from <https://docs.python.org>