

**LAPORAN RESTORASI CITRA
CITRA DIGITAL**



Ditulis oleh:

**Renaldi Septian
NIM. 226201029**

**D3 TEKNIK KOMPUTER
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI SAMARINDA
2024**

DAFTAR ISI

DAFTAR ISI	i
BAB I PENDAHULUAN	1
1.1 PROGRAM 1	1
1.1.1 Blur.....	1
1.1.2 Gaussian	1
1.1.3 Bilateral	1
1.1.4 Median.....	2
1.2 PROGRAM 2	2
1.2.1 Median.....	2
1.2.2 Maksimum	2
1.2.3 Minimum.....	3
BAB II SOURCE CODE	4
2.1 Tujuan penerapan Source Code.....	4
2.1.1 Program 1	4
2.1.2 Program 2	4
2.2 Source Code	4
2.2.1 Program 1	4
2.2.2 Program 2	6
2.3 Hasil Source Code	8
2.3.1 Program 1	8
2.3.2 Program 2	10
2.4 Histogram	10
2.4.1 Program 1	10

2.4.2	Program 2	10
2.5	Perbandingan	11
BAB III PENUTUP		12
3.1	Kesimpulan.....	12
3.2	Opini Penulis	12
DAFTAR PUSTAKA		13

BAB I

PENDAHULUAN

1.1 PROGRAM 1

1.1.1 Blur

Blur dalam pemrosesan citra merujuk pada pengurangan ketajaman atau kejelasan dalam sebuah gambar. Ini adalah teknik yang melibatkan penyebaran piksel dalam gambar untuk mengurangi detail dan noise. Teknik ini berguna untuk berbagai tugas pemrosesan citra, seperti menghaluskan gambar, mengurangi visibilitas ketidaksempurnaan dan cacat, serta mengurangi noise dalam gambar

Blur: Teknik yang mengurangi ketajaman atau kejelasan gambar dengan menyebarkan piksel untuk mengurangi detail dan noise. Berguna untuk menghaluskan gambar dan mengurangi ketidaksempurnaan.

1.1.2 Gaussian

Gaussian Blur adalah teknik penghalusan gambar untuk mengurangi noise dan variasi acak dalam kecerahan atau warna piksel. Metode ini menggunakan dua fungsi matematika untuk sumbu x dan y, membentuk fungsi konvolusi yang menghasilkan distribusi normal dari nilai piksel. Penghalusan yang dihasilkan bergantung pada ukuran radius blur yang dipilih. Setiap piksel mendapatkan nilai baru sebagai rata-rata tertimbang dari piksel di sekitarnya, dengan bobot lebih tinggi untuk piksel yang lebih dekat. Hasilnya, gambar tampak lebih kabur dan halus.

Gaussian Blur: Menggunakan fungsi konvolusi untuk menghasilkan distribusi normal dari nilai piksel. Setiap piksel baru merupakan rata-rata tertimbang dari piksel di sekitarnya. Efektif untuk menghaluskan gambar dan mengurangi variasi acak dalam kecerahan atau warna.

1.1.3 Bilateral

Bilateral adalah teknik penghalusan gambar yang mempertahankan tepi-tepi objek sambil mengurangi noise. Teknik ini menghitung rata-rata tertimbang dari piksel-piksel di sekitarnya dengan mempertimbangkan perbedaan intensitas dan posisi. Hal ini memungkinkan filter untuk menghaluskan area yang homogen dalam gambar sambil menjaga tepi tetap tajam.

Bilateral Filter: Teknik yang mempertahankan tepi objek sambil mengurangi noise. Menghitung rata-rata tertimbang dari piksel-piksel di sekitarnya,

mempertimbangkan perbedaan intensitas dan posisi. Menghaluskan area homogen sambil menjaga tepi tetap tajam.

1.1.4 Median

Median Blur adalah teknik penghalusan gambar yang mengambil nilai median dari semua piksel dalam area kernel, menggantikan elemen tengah dengan nilai median tersebut. Sangat efektif untuk mengatasi noise salt-and-pepper. Dalam median blurring, elemen tengah selalu digantikan oleh nilai piksel yang sudah ada dalam gambar, sehingga lebih efektif dalam mengurangi noise. Ukuran kernel harus bilangan ganjil positif.

Median Blur: Mengambil nilai median dari semua piksel dalam area kernel, menggantikan elemen tengah dengan nilai median. Efektif untuk menghilangkan noise salt-and-pepper. Ukuran kernel harus bilangan ganjil positif.

1.2 PROGRAM 2

1.2.1 Median

Filter median adalah jenis filter statistik urutan yang digunakan dalam pemrosesan citra untuk menggantikan piksel tengah dari suatu area dengan nilai median dari jendela tersebut. Filter ini efektif menghilangkan jenis-jenis noise tertentu seperti noise Gaussian, acak, dan salt-and-pepper dari gambar.

Filter Median: Jenis filter statistik yang menggantikan piksel tengah dari suatu area dengan nilai median dari jendela tersebut. Efektif menghilangkan noise Gaussian, acak, dan salt-and-pepper.

1.2.2 Maksimum

Filter maksimum digunakan untuk memperluas batas objek dalam citra. Setiap piksel diperbarui berdasarkan perbandingan dengan piksel di sekitarnya dalam jendela yang berjalan. Piksel tengah digantikan dengan piksel terterang dalam jendela tersebut. Filter ini berguna untuk memperjelas batas-batas objek dalam citra yang kabur.

Filter Maksimum (Dilasi): Memperluas batas objek dalam citra dengan menggantikan piksel tengah dengan piksel terterang dalam jendela yang berjalan. Berguna untuk memperjelas batas-batas objek.

1.2.3 Minimum

Filter minimum digunakan untuk mengurangi bentuk-bentuk pada citra. Setiap piksel diperbarui berdasarkan perbandingan dengan piksel di sekitarnya dalam jendela yang berjalan. Piksel tengah digantikan dengan piksel terendah dalam jendela tersebut. Filter ini berguna untuk menghilangkan noise atau detail-detail kecil yang tidak diinginkan dalam citra.

Filter Minimum (Erosi): Mengurangi bentuk-bentuk pada citra dengan menggantikan piksel tengah dengan piksel terendah dalam jendela yang berjalan. Berguna untuk menghilangkan noise atau detail kecil yang tidak diinginkan.

BAB II

SOURCE CODE

2.1 Tujuan penerapan Source Code

2.1.1 Program 1

Pada Program 1 tujuan kode tersebut untuk menerapkan dan membandingkan efek dari berbagai teknik pengaburan pada gambar grayscale dan untuk memberikan pemahaman visual tentang bagaimana setiap teknik mempengaruhi gambar dan histogram intensitas pikselnya. Ini berguna dalam pengolahan citra untuk mengurangi noise, mempersiapkan gambar untuk analisis lebih lanjut, atau hanya untuk tujuan visualisasi.

2.1.2 Program 2

Pada Program 2 tujuan kode tersebut, secara keseluruhan untuk mempelajari dan membandingkan efek dari berbagai teknik pemfilteran pada citra grayscale, serta untuk memvisualisasikan hasilnya. Ini penting dalam pengolahan citra untuk memahami bagaimana setiap teknik dapat mengubah kualitas gambar dan membantu dalam mengurangi noise atau meningkatkan detail.

2.2 Source Code

2.2.1 Program 1

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Membaca gambar asli berwarna
img_color = cv2.imread('images2.jpeg', 1)

# Membaca gambar yang sama dalam skala abu-abu
img_gray = cv2.imread('images2.jpeg', 0)

# Menerapkan teknik blur yang berbeda
blur1 = cv2.blur(img_gray, (3, 3))          # Blur Rata-rata
blur2 = cv2.GaussianBlur(img_gray, (3, 3), 0) # Gaussian Blur
```

```

median = cv2.medianBlur(img_gray, 3)          # Median Blur
blur3 = cv2.bilateralFilter(img_gray, 9, 75, 75) # Bilateral Blur

# Fungsi untuk menampilkan gambar dan histogramnya
def plot_image_and_histogram(image, title, is_color=False):
    plt.figure(figsize=(12, 5)) # Membuat figure dengan ukuran yang ditentukan
    plt.subplot(1, 2, 1) # Menambahkan subplot dalam grid 1x2 di posisi 1
    if is_color:
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB) # Mengubah BGR
        ke RGB jika gambar berwarna
        plt.imshow(image) # Menampilkan gambar berwarna
    else:
        plt.imshow(image, cmap='gray') # Menampilkan gambar skala abu-abu
        plt.title(f'{title} Image', fontsize=14) # Menentukan judul subplot gambar
        plt.axis('off') # Mematikan sumbu

    plt.subplot(1, 2, 2) # Menambahkan subplot dalam grid 1x2 di posisi 2
    plt.hist(image.ravel(), bins=256, color='blue', alpha=0.7) # Menampilkan
    histogram gambar
    plt.title(f'Histogram of {title}', fontsize=14) # Menentukan judul subplot
    histogram
    plt.xlabel('Pixel Intensity') # Label sumbu x
    plt.ylabel('Frequency') # Label sumbu y
    plt.grid(True, linestyle='--', alpha=0.5) # Menambahkan grid ke histogram
    plt.xlim([0, 256]) # Menentukan batas sumbu x
    plt.tight_layout() # Menyesuaikan subplot agar sesuai dengan area figure
    plt.show() # Menampilkan plot

# Menampilkan gambar asli berwarna dan histogramnya
plot_image_and_histogram(img_color, "Original", is_color=True)

# Menampilkan gambar yang diproses dan histogramnya

```



```

plot_image_and_histogram(blur1, "Averaging Blur")
plot_image_and_histogram(blur2, "Gaussian Blur")
plot_image_and_histogram(blur3, "Bilateral Blur")
plot_image_and_histogram(median, "Median Blur")

```

```

cv2.waitKey(0)
cv2.destroyAllWindows()

```

2.2.2 Program 2

```

import cv2
import numpy as np
from matplotlib import pyplot as plt

# Membaca gambar asli berwarna
img_color = cv2.imread('images2.jpeg', 1)

# Membaca gambar yang sama dalam skala abu-abu
img_gray = cv2.imread('images2.jpeg', 0)

# Menghitung histogram dari gambar asli dalam skala abu-abu
histr_img = cv2.calcHist([img_gray], [0], None, [256], [0, 256])

# Menerapkan berbagai filter pada gambar skala abu-abu
filter_maximum = cv2.dilate(img_gray, np.ones((3, 3), np.uint8)) # Filter
Maksimum
filter_minimum = cv2.erode(img_gray, np.ones((3, 3), np.uint8)) # Filter
Minimum
filter_median = cv2.medianBlur(img_gray, 3) # Filter Median
filter_gaussian = cv2.GaussianBlur(img_gray, (3, 3), 0) # Filter Gaussian

# Menghitung histogram dari gambar hasil filter
histr_maximum = cv2.calcHist([filter_maximum], [0], None, [256], [0, 256])
histr_minimum = cv2.calcHist([filter_minimum], [0], None, [256], [0, 256])

```

```

histr_median = cv2.calcHist([filter_median], [0], None, [256], [0, 256])
histr_gaussian = cv2.calcHist([filter_gaussian], [0], None, [256], [0, 256])

# Judul untuk masing-masing gambar dan histogram
titles = ['Original Image', 'Maximum Filter', 'Minimum Filter', 'Median Filter',
'Gaussian Filter']
images = [img_color, filter_maximum, filter_minimum, filter_median,
filter_gaussian]
histograms = [histr_img, histr_maximum, histr_minimum, histr_median,
histr_gaussian]

# Membuat figure dengan ukuran yang ditentukan
plt.figure(figsize=(12, 10))
for i in range(5):
    plt.subplot(5, 2, 2*i+1) # Menambahkan subplot untuk gambar
    if i == 0:
        plt.imshow(cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB)) #
Mengonversi dari BGR ke RGB untuk tampilan yang benar
    else:
        plt.imshow(images[i], cmap='gray') # Menampilkan gambar dalam skala abu-
abu
    plt.title(titles[i]) # Menambahkan judul pada gambar
    plt.axis('off') # Mematikan sumbu

    plt.subplot(5, 2, 2*i+2) # Menambahkan subplot untuk histogram
    plt.plot(histograms[i]) # Menampilkan histogram
    plt.xlim([0, 256]) # Menentukan batas sumbu x
    plt.title(f'Histogram of {titles[i]}') # Menambahkan judul pada histogram
    plt.xticks([]) # Mematikan label sumbu x
    plt.yticks([]) # Mematikan label sumbu y

plt.tight_layout() # Menyesuaikan layout agar subplot tidak saling tumpang tindih

```

```
plt.show()
```

```
cv2.waitKey(0)
```

```
cv2.destroyAllWindows()
```

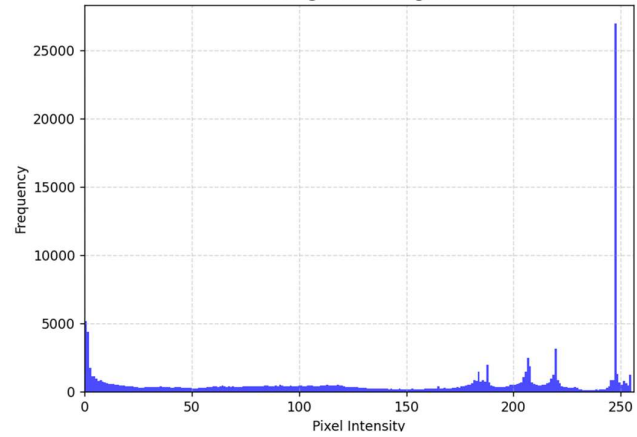
2.3 Hasil Source Code

2.3.1 Program 1

Original Image



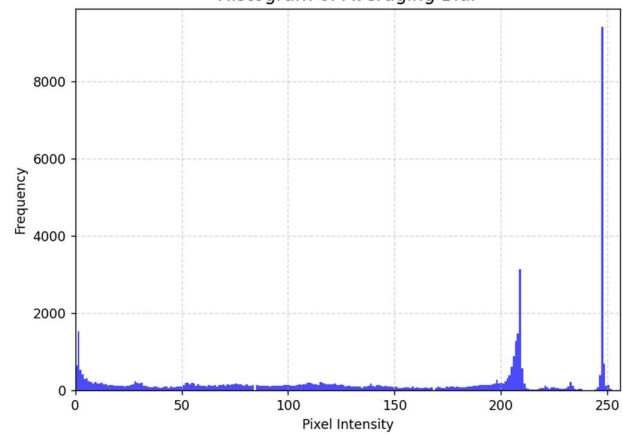
Histogram of Original



Averaging Blur Image



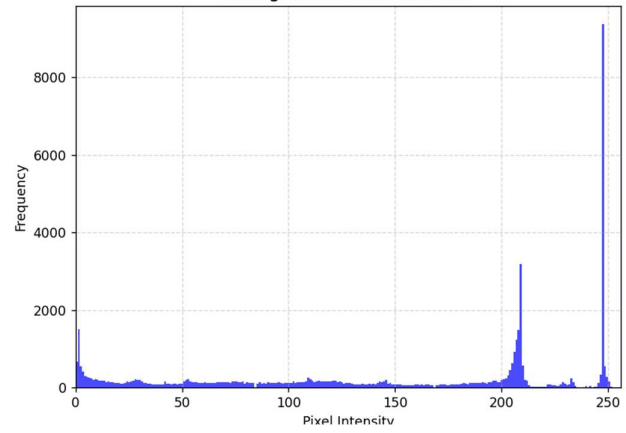
Histogram of Averaging Blur



Gaussian Blur Image



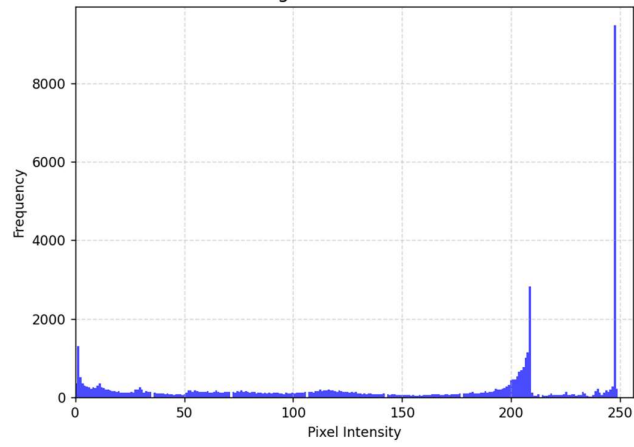
Histogram of Gaussian Blur



Bilateral Blur Image



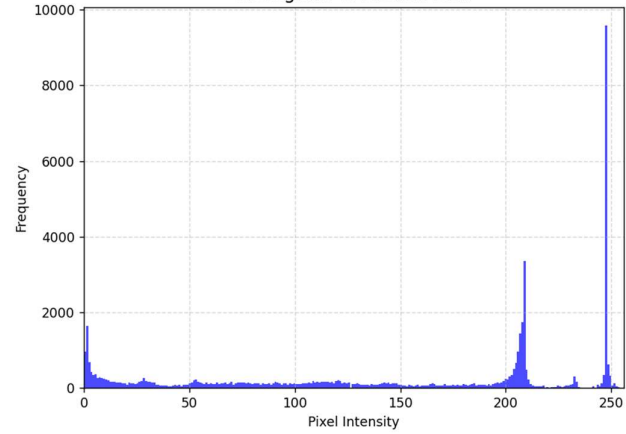
Histogram of Bilateral Blur



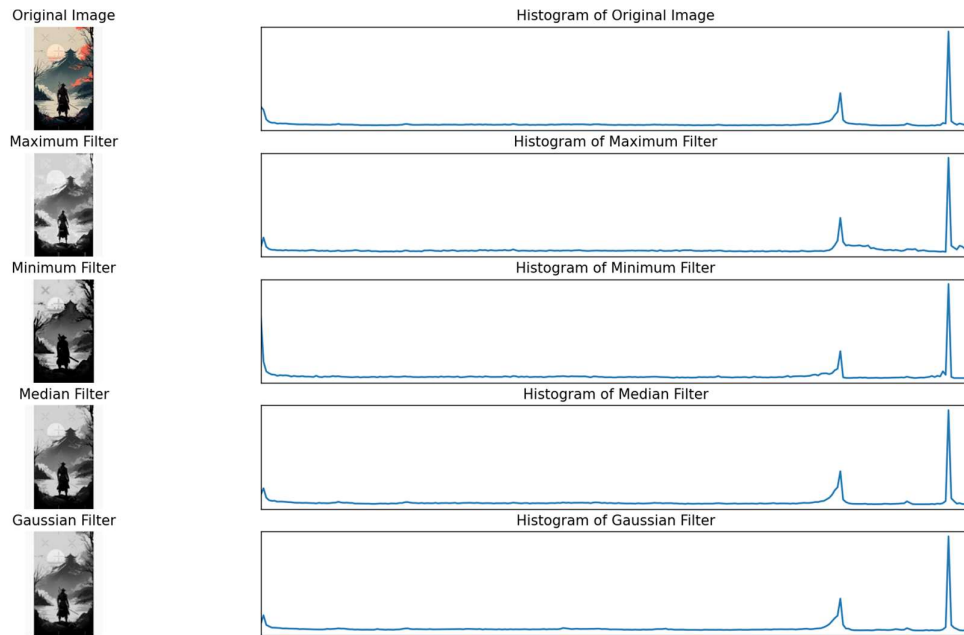
Median Blur Image



Histogram of Median Blur



2.3.2 Program 2



2.4 Histogram

Pengertian Histogram

adalah grafik yang menunjukkan frekuensi kemunculan setiap nilai intensitas (biasanya dalam rentang dari 0 hingga 255 untuk citra grayscale) dan memberikan informasi penting tentang kecerahan (brightness) dan kontras (contrast) citra tersebut.

2.4.1 Program 1

Hasil Histogram pada program 1 menunjukkan bagaimana metode blur mempengaruhi citra dan intensitas pikselnya. Untuk blur gaussian ini menghasilkan efek blur lebih alami dibandingkan blur rata rata, pada bilateral mempertahankan detail penting sekaligus mengurangi noise pada area yang lebih seragam.

2.4.2 Program 2

Hasil Histogram pada program 2 menunjukkan bagaimana berbagai filter mempengaruhi citra dan intensitas pikselnya. Untuk filter maksimum menunjukkan peningkatan nilai intensitas piksel, menyebabkan gambar tampak lebih terang, pada filter minimum terjadi penurunan intensitas piksel yang menyebabkan gambar tampak lebih gelap, pada filter median menunjukkan pengurangan noise dan

mempertahankan tepi objek dalam gambar, histogrammnya lebih halus dibandingkan gambar asli.

2.5 Perbandingan

Program 1 lebih menekankan pada teknik blurring untuk penghalusan citra, sedangkan Program 2 mengeksplorasi teknik morfologi dan blurring sambil memberikan analisis histogram yang lebih mendalam. Keduanya memiliki tujuan yang berbeda namun saling melengkapi dalam pemrosesan citra.

BAB III

PENUTUP

3.1 Kesimpulan

Setelah mempelajari dari program 1 dan program 2, dapat disimpulkan bahwa kedua program tersebut, menunjukkan dari segi Teknik pemrosesan citra dapat digunakan untuk meningkatkan kualitas visual dan analisis gambar. Metode Teknik pada kedua program tersebut diantaranya dapat memainkan peran penting dalam mengurangi noise dan memperjelas fitur dalam citra.

Dalam segi analisis histogram Kedua program menghitung dan menampilkan histogram dari citra untuk memberikan wawasan mengenai distribusi intensitas pixel. Ini penting dalam analisis citra, karena membantu dalam memahami bagaimana berbagai metode pemrosesan mempengaruhi nilai pixel dalam citra.

3.2 Opini Penulis

Program 2 menghitung histogram sebelum dan sesudah penerapan setiap filter, memberikan gambaran yang lebih jelas tentang bagaimana setiap teknik mempengaruhi distribusi intensitas pixel. Ini sangat membantu dalam memahami efek setiap metode secara kuantitatif.

Alasannya Tidak Menggunakan Berbagai Teknik Blurring Program 2 tidak mengeksplorasi berbagai metode blurring secara mendalam seperti yang dilakukan oleh Program 1. Hal ini dapat membatasi pemahaman tentang penghalusan citra.

DAFTAR PUSTAKA

- Blur in Image Processing — An Introductory Guide* | by Henrique Vedoveli |. (n.d.). Retrieved from Medium: <https://medium.com/@henriquevedoveli/blur-in-image-processing-an-introductory-guide-88a9550985e7>
- Median Filter - an overview* | ScienceDirect Topics. (n.d.). Retrieved from Science Direct: <https://www.sciencedirect.com/topics/computer-science/median-filter#:~:text=A%20median%20filter%20is%20a,and%20pepper%20noise%20from%20images>.
- Minimum, Maximum, and Median Filters - Graphics Mill*. (n.d.). Retrieved from Graphics Mill: <https://www.graphicsmill.com/docs/gm/minimum-maximum-median-filters.htm?form=MG0AV3>
- Open CV: Smoothing Images*. (n.d.). Retrieved from OpenCV: https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html
- Using Blur Gaussian In Image Processing*. (n.d.). Retrieved from https://www.adobe.com/id_id/creativecloud/photography/discover/gaussian-blur.html