

LAPORAN TEORI

PENGOLAHAN CITRA DIGITAL



NAMA : ROMI FIRMANSYAH

NIM : 202331214

KELAS : D

DOSEN : Yessy Fitriani, ST., M.Kom

NO.PC : 39

ASISTEN : 1. Abdur Rasyid Ridho

2. Rizqy Amanda

3. Kashrina Masyid Azka

4. Izzat Islami Kagapi

INSTITUT TEKNOLOGI PLN

SISTEM INFORMASI

2025

1. Apa itu operasi konvolusi dalam konteks pengolahan citra digital?

Konvolusi operasi dalam pemrosesan citra digital adalah proses matematika untuk mengubah nilai-nilai piksel dari suatu citra (gambar) dengan menggunakan kernel (atau filter). Filter ini adalah matriks kecil yang digeser atas citra. Pada setiap posisi filter dihasilkan nilai output baru berdasarkan hasil perkalian antara elemen filter dan elemen piksel yang bersesuaian yang kemudian dijumlahkan. **Tujuan utama** dari konvolusi adalah untuk mengekstraksi fitur seperti tepi, tekstur, atau pola, serta melakukan transformasi seperti blur, sharpen, edge detection, dsb.

2. Perbedaan mendasar antara mean filter & median filter, serta kapan waktu penggunaannya

202331214_ROMI FIRMANSYAH		
Aspek	Mean Filter (Rata-rata)	Median Filter (Median)
Cara Kerja	Menghitung nilai Mean dari piksel	Mengambil nilai Median Dari piksel
Sensitivitas	Rentan terhadap Noise ekstrem	Lebih tahan terhadap Noise ekstrem
Efek Pada gambar	Cenderung mengaburkan tepi	Mempertahankan tepi lebih baik
Speed	Lebih cepat secara komputasi	Lebih lambat karena perlu sortir nilai
WAKTU PENGGUNAAN		
Mean filter	digunakan saat ingin menghaluskan citra dan noise tidak terlalu tajam	
Median filter	menghilangkan noise yang menyebabkan piksel ekstrem	

3. Proses konvolusi satu piksel [Filter 3 x 3]

```

Citra 3x3:      Kernel 3x3 (Blur):
[10  20  30]    [1  1  1]
[40  50  60]    [1  1  1]
[70  80  90]    [1  1  1]

→ Hasil = (10 + 20 + 30 + 40 + 50 + 60 + 70 + 80 + 90) = 450
→ Output = 450 / 9 = 50
  
```

4. Mengapa konvolusi penting dalam pengolahan citra dan CNN (Deep Learning)?

Konvolusi dianggap fundamental karena :

- **Ekstraksi fitur otomatis:** konvolusi dapat ditemukan Tanpa memerlukan proses
- **Parameter sharing:** kernel yang sama digunakan untuk semua wilayah gambar
- **Efisiensi komputasi:** bisa mendeteksi lokal dengan jumlah parameter lebih sedikit
- **Invarian terhadap translasi:** Membantu mengenali objek meskipun berpindah posisi di gambar.
- **Signifikansi:** Tanpa konvolusi, CNN tidak akan mampu mengenali sebuah objek secara efisien.

5. Aplikasi utama konvolusi dalam pengolahan citra + contoh nyata

Aplikasi utama:

- Pendeteksian tepi (edge detection)
- Blur / smoothing gambar
- Sharpening / peningkatan detail
- Pengenalan pola (pattern recognition)
- Segmentasi citra
- Pembersihan noise

Contoh konkret:

- Camera HP : Otomatis mengoptimalkan kualitas gambar, menghilangkan noise, mendeteksi wajah.
- Google Lens : Menggunakan konvolusi untuk mengenali tulisan pada gambar.
- Diagnostik medis : CNN dengan konvolusi digunakan untuk mendeteksi tumor dari MRI atau CT scan.

LAPORAN PRAKTIKUM

PENGOLAHAN CITRA DIGITAL



NAMA : ROMI FIRMANSYAH

NIM : 202331214

KELAS : D

DOSEN : Yessy Fitriani, ST., M.Kom

NO.PC : 39

ASISTEN : 1. Abdur Rasyid Ridho

2. Rizqy Amanda

3. Kashrina Masyid Azka

4. Izzat Islami Kagapi

INSTITUT TEKNOLOGI PLN

SISTEM INFORMASI

2025

Codingan di Jupyter Notebook

Jupyter 202331214_PCD_Konvolusi Last Checkpoint: 13 hours ago



File Edit View Run Kernel Settings Help

Trusted

Code

JupyterLab Python 3 (ipykernel)

202331214_ROMI FIRMANSYAH

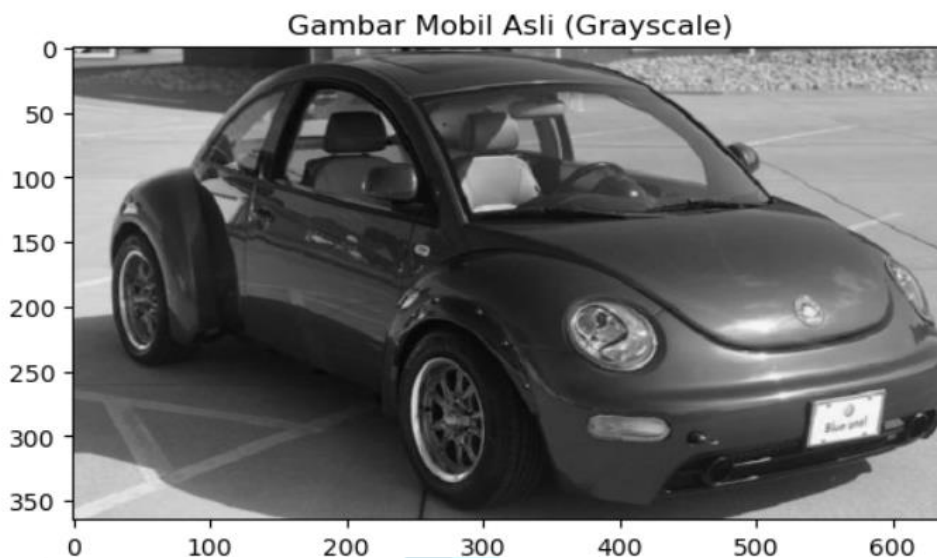
```
[2]: import cv2
import numpy as np
import matplotlib.pyplot as plt
```

Penjelasan :

- cv2: library OpenCV untuk memproses gambar.
- numpy: digunakan untuk operasi numerik dan array.
- matplotlib.pyplot: untuk menampilkan gambar atau grafik.

Bagian 1 { Gambar Mobil } Membaca dan Menampilkan Gambar Asli

```
[7]: # 202331214_ROMI FIRMANSYAH
img_mobil = cv2.imread('kendaraan.jpg')
citra_mobil_gray = cv2.cvtColor(img_mobil, cv2.COLOR_BGR2GRAY)
plt.imshow(citra_mobil_gray, cmap='gray')
plt.title('Gambar Mobil Asli (Grayscale)')
plt.show()
```

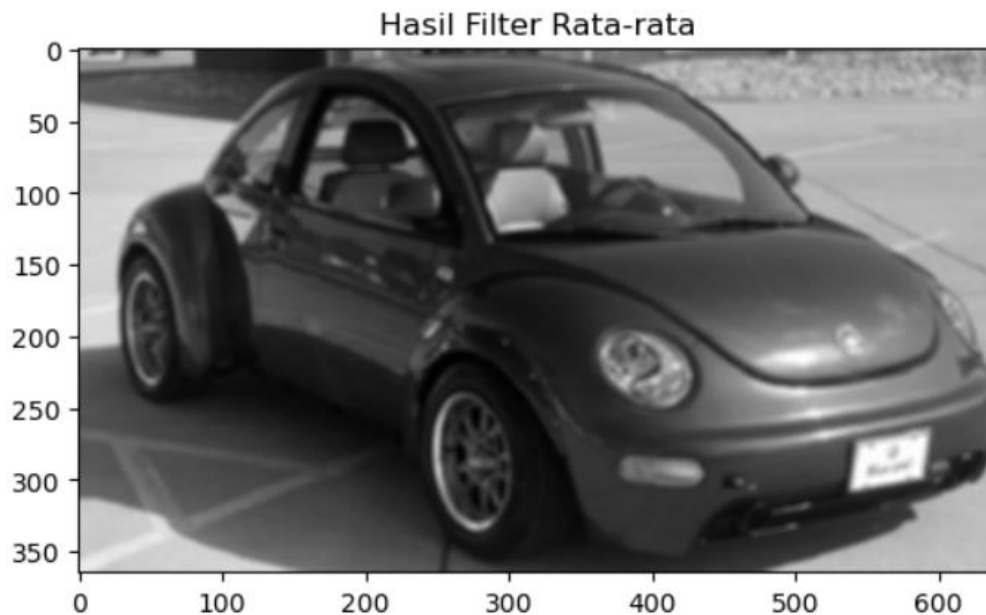


Penjelasan :

- **img_mobil = cv2.imread('kendaraan.jpeg')**: Membaca file gambar kendaraan.jpeg dari folder dan menyimpannya ke dalam variabel bernama img_mobil.
- **citra_mobil_gray = cv2.cvtColor**: Mengubah gambar dari berwarna (format BGR) menjadi hitam-putih (GRAY). Ini dilakukan karena operasi filter lebih mudah dan umum diterapkan pada gambar satu warna.
- **plt.imshow**: Perintah untuk menampilkan gambar. **cmap='gray'** dalam skala abu-abu.
- **plt.title**: judul pada jendela gambar yang akan muncul.
- **plt.show()**:jendela gambar tersebut di layar.

Mean Filter

```
•[10]: # 202331214_ROMI_FIRMANSYAH
kernel_mean = np.ones((5, 5), np.float32) / 25
citra_mobil_mean = cv2.filter2D(citra_mobil_gray, -1, kernel_mean)
plt.imshow(citra_mobil_mean, cmap='gray')
plt.title('Hasil Filter Rata-rata')
plt.show()
```



Penjelasan :

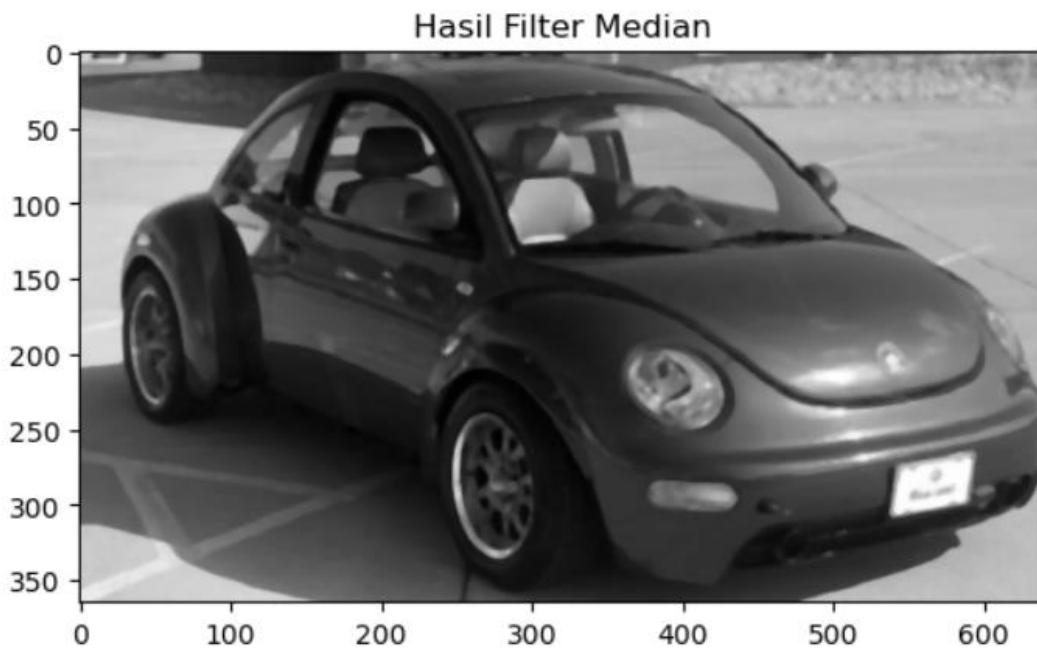
- A. **kernel_mean = np.ones((5, 5), np.float32) / 25:**
Key untuk membuat filter & matriks berukuran 5x5 piksel yang semua isinya adalah angka 1 & Inilah yang disebut **kernel**.
- B. **citra_mobil_mean = cv2.filter2D:**
Menerapkan filter ke gambar. Fungsi ini "menggeser" kernel yang kita buat ke seluruh bagian gambar, lalu menghitung nilai rata-rata piksel di bawahnya untuk menghasilkan gambar baru.
- C. **plt.imshow, plt.title, plt.show():**
Menampilkan gambar yang sudah diberi efek filter.

Hasil dan Analisis:

Efek ini terjadi karena detail-detail tajam pada gambar "dirata-ratakan" dengan piksel di sekelilingnya, sehingga perbedaannya menjadi lebih landai.

Median Filter

```
•[13]: # 202331214_ROMI_FIRMANSYAH
citra_mobil_median = cv2.medianBlur(citra_mobil_gray, 5)
plt.imshow(citra_mobil_median, cmap='gray')
plt.title('Hasil Filter Median')
plt.show()
```



Penjelasan

A. `citra_mobil_median = cv2.medianBlur(citra_mobil_gray, 5)` :

Menggunakan fungsi bawaan `medianBlur` dari OpenCV.

- **`citra_mobil_gray`** :
Gambar yang ingin kita filter.
- **`5`** : Ukuran area piksel tetangga yang dipertimbangkan (area 5x5), dan akan mencari nilai tengah (median) dari 25 piksel tersebut untuk menjadi nilai piksel baru.

B. `plt.imshow, plt.title, plt.show()` :

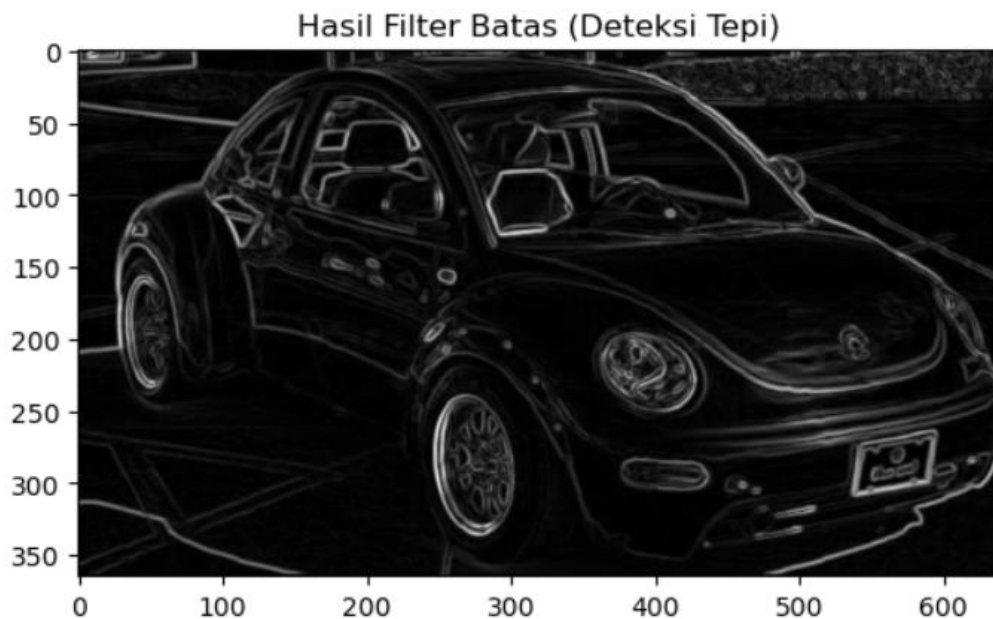
Menampilkan gambar hasil filter median.

Hasil Analisis Saya:

Saya mendapatkan sebuah pengertian bahwa gambar yang dihasilkan cukup halus, tapi jika dibandingkan dengan filter rata-rata, tepi objek (garis jendela atau lekuk bodi mobil) terlihat **lebih tajam dan tidak terlalu buram**. Ini karena filter median lebih ampuh dalam mempertahankan garis batas objek.

Edge Filter

```
[16]: #202331214_ROMI_FIRMANSYAH
sobel_x = cv2.Sobel(citra_mobil_gray, cv2.CV_64F, 1, 0, ksize=5)
sobel_y = cv2.Sobel(citra_mobil_gray, cv2.CV_64F, 0, 1, ksize=5)
citra_mobil_tepi = np.sqrt(sobel_x**2 + sobel_y**2)
plt.imshow(citra_mobil_tepi, cmap='gray')
plt.title('Hasil Filter Batas (Deteksi Tepi)')
plt.show()
```



Penjelasan

- **sobel_x = cv2.Sobel(..., 1, 0, ...)** :Menerapkan filter Sobel untuk mencari tepi **vertikal** (garis dari atas ke bawah).
- **sobel_y = cv2.Sobel(..., 0, 1, ...)** :Menerapkan filter Sobel untuk mencari tepi **horizontal** (garis dari kiri ke kanan).
- **citra_mobil_tepi = np.sqrt** :Menggabungkan hasil dari kedua deteksi tepi (vertikal dan horizontal) menjadi satu gambar utuh yang menunjukkan semua tepi.
- **plt.imshow, plt.title, plt.show()** :Menampilkan hasil akhir deteksi tepi.

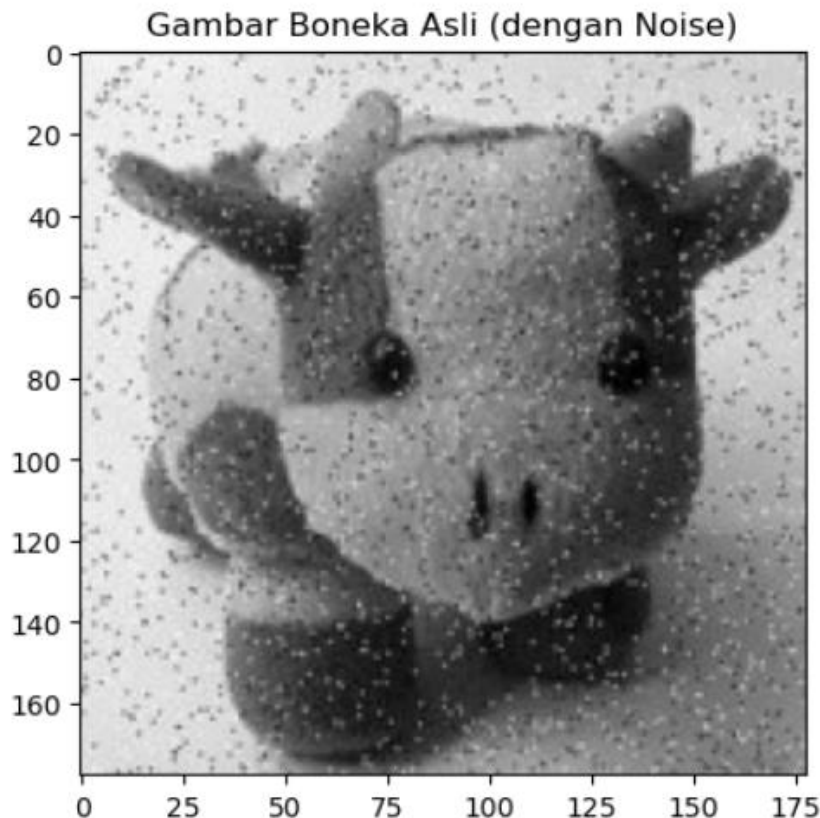
Hasil Analisis Saya :

Disini juga saya cukup mengamati dan mendapat kesimpulan bahwa hasil output disini sangat berbeda. Yaitu warna gambar menjadi hitam pekat, dan yang terlihat hanyalah **garis-garis putih yang membentuk kontur mobil**. Ini dikarena filter hanya menyorot area di mana ada perubahan warna yang drastis, yaitu di bagian tepi objek.

Bagian 2 {Gambar Boneka} ¶

Membaca & Menampilkan Gambar Boneka

```
•[21]: # 202331214_ROMI_FIRMANSYAH
citra_boneka = cv2.imread('boneka.jpg', cv2.IMREAD_GRAYSCALE)
plt.imshow(citra_boneka, cmap='gray')
plt.title('Gambar Boneka Asli (dengan Noise)')
plt.show()
```

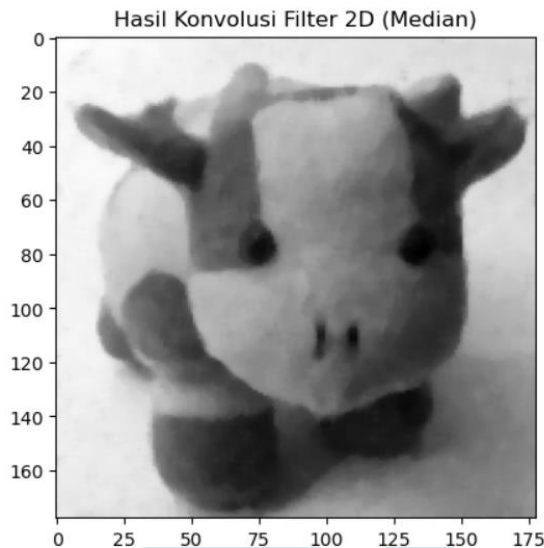


Penjelasan :

- **citra_boneka = cv2.imread(..., cv2.IMREAD_GRAYSCALE)** :Membaca gambar boneka.jpg dan langsung menyimpannya sebagai gambar hitam-putih.
- **citra_boneka_bersih = cv2.medianBlur(citra_boneka, 3)** :Kita kembali menggunakan **filter median**. Menurut saya ini adalah pilihan terbaik karena *noise* pada gambar ini adalah tipe *salt-and-pepper* (bintik-bintik), dan filter median sangat efektif untuk menghilangkannya. **Angka 3** berarti kita menggunakan area **3x3**, cukup untuk membersihkan tanpa membuat gambar terlalu buram.
- **fig, axes = plt.subplots** :Perintah untuk menyiapkan sebuah "bingkai" yang bisa menampung dua gambar sekaligus.
- **axes[0].imshow & axes[0].set_title** :Menampilkan gambar asli (yang kotor) di posisi pertama (kiri).
- **axes[1].imshow & axes[1].set_title** :Menampilkan gambar hasil filter (yang bersih) di posisi kedua (kanan).
- **plt.show()** :Menampilkan bingkai berisi kedua gambar tersebut.

Konvolusi dengan Filter 2D untuk Menghilangkan Noise ya gaes ya!

```
*[24]: # 202331214_ROMI_FIRMANSYAH
citra_boneka_bersih = cv2.medianBlur(citra_boneka, 5)
plt.imshow(citra_boneka_bersih, cmap='gray')
plt.title('Hasil Konvolusi Filter 2D (Median)')
plt.show()
```



Penjelasan :

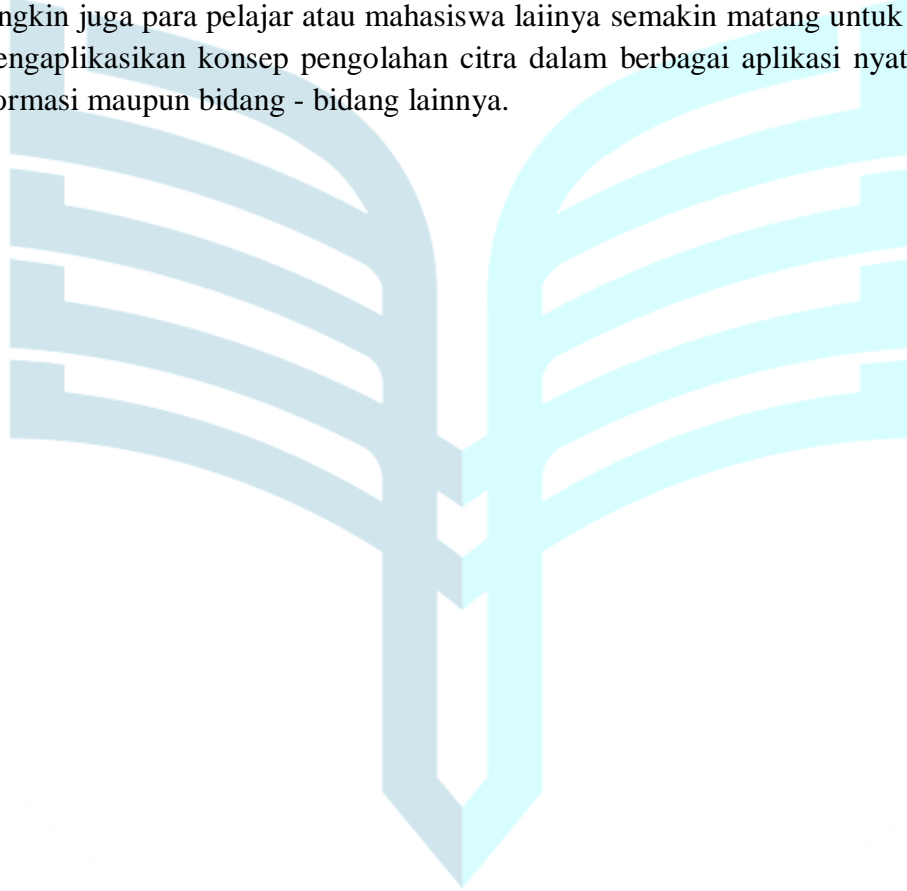
- 1) **citra_boneka_bersih = cv2.medianBlur(citra_boneka, 5):**
 - **cv2.medianBlur:** fungsi dari library OpenCV untuk menerapkan **filter median**.
 - **citra_boneka:** variabel yang berisi gambar asli dari gambar yang masih memiliki *noise*.
 - **5:** Ini adalah ukuran *kernel* atau filter (disebut *aperture size*) yang digunakan untuk matriks 5x5 piksel. Untuk setiap posisi akan mengambil semua nilai piksel di bawahnya (25 piksel), mengurutkannya, dan memilih nilai tengah (median) sebagai nilai piksel baru untuk posisi tersebut di gambar output.
 - **citra_boneka_bersih:** variabel baru yang akan menyimpan hasil gambar yang sudah dibersihkan dari *noise*.
- 2) **plt.imshow(citra_boneka_bersih, cmap='gray'):** adalah fungsi dari Matplotlib untuk menampilkan gambar.
 - **citra_boneka_bersih:** Menampilkan gambar yang sudah diproses.
 - **cmap='gray':** Menampilkan gambar dalam mode *grayscale* (skala keabuan).
- 3) **plt.title('Hasil Konvolusi Filter 2D (Median)'):** Memberikan judul pada plot atau gambar yang akan ditampilkan.
- 4) **plt.show():** Perintah untuk menampilkan jendela yang berisi gambar tersebut.

Kesimpulan

Setelah saya mengerjakan, menyelesaikan, dan mengamati semuanya mulai dari yang paling awal hingga paling akhir saya menarik kesimpulan atas hasil teori dan praktikum bahwa :

konvolusi operasi adalah teknologi dasar yang sangat penting dalam pengolahan citra digital. Melalui operasional berbagai macam filter seperti **mean, median, dan Sobel**, Maka konvolusi dapat digunakan untuk **menghaluskan foto, menghilangkan noise**, serta **mendeteksi tepi obyek dengan baik**. Dan setiap masing – masing filter memiliki kelebihanannya masing-masing tergantung dari user atau perencanaan pengolahan citra.

Bukan hanya itu, konvolusi juga menjadi dasar inti di dalam teknologi kecerdasan buatan, di mana model Convolutional Neural Network (CNN) sangat banyak digunakan dalam sektor pengenalan objek, kamera digital, bahkan sampai diagnostik medis. Berbekal pengetahuan proses konvolusi, dan saya atau mungkin juga para pelajar atau mahasiswa lainnya semakin matang untuk mempersiapkan diri untuk mengaplikasikan konsep pengolahan citra dalam berbagai aplikasi nyata dalam bidang teknologi informasi maupun bidang - bidang lainnya.



DAFTAR PUSTAKA

1. **Gonzalez, R. C., & Woods, R. E. (2018).** *Digital Image Processing (4th Edition)*. Pearson. Buku ini merupakan referensi utama dalam bidang pengolahan citra digital dan membahas secara rinci tentang konvolusi dan berbagai filter. <https://www.pearson.com/en-us/subject-catalog/p/digital-image-processing/P200000000627>
2. **OpenCV Documentation – Image Filtering** Dokumentasi resmi OpenCV yang menjelaskan berbagai jenis filter termasuk mean, median, dan Sobel filter. https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html
3. **Raschka, S., & Mirjalili, V. (2022).** *Python Machine Learning (3rd Edition)*. Packt Publishing. Buku ini menjelaskan penerapan CNN dan konvolusi dalam machine learning serta implementasi dengan Python dan pustaka seperti TensorFlow dan OpenCV. <https://www.packtpub.com/product/python-machine-learning-third-edition/9781789955750.com>
4. **GeeksforGeeks – Image Processing with OpenCV** Tutorial praktis dan mudah diikuti untuk mempelajari pengolahan citra dengan Python dan OpenCV, termasuk filter median dan Sobel. <https://www.geeksforgeeks.org/python-image-processing-using-opencv/>
5. **IBM Cloud Education – Convolutional Neural Networks** Artikel yang menjelaskan peran konvolusi dalam deep learning dan pengolahan citra. <https://www.ibm.com/topics/convolutional-neural-networks>