

LAPORAN TEORI

PENGOLAHAN CITRA DIGITAL



NAMA : Rama Dinantiar

NIM : 202331044

KELAS : Pengolahan Citra Digital C

DOSEN : Ir. Darma Rusjdi, M.Kom

NO.PC : 21

ASISTEN : 1. Abdur Rasyid Ridho

2. Rizqy Amanda

3. Kashrina Masyid Azka

4. Izzat Islami Kagapi

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2025

1. Operasi **konvolusi** dalam pengolahan citra digital adalah proses matematika yang dilakukan pada citra dengan cara mengalikan matriks citra dengan matriks kecil yang disebut **kernel** atau **filter**, lalu menjumlahkan hasilnya untuk menghasilkan nilai piksel baru pada citra output.

Operasi ini dilakukan dengan **menggeser kernel** di seluruh area citra input (biasanya baris demi baris, kolom demi kolom), dan pada setiap posisi, dilakukan perhitungan elemen-elemen yang beririsan.

Contoh sederhana operasi konvolusi adalah saat menerapkan filter deteksi tepi, blur, sharpening, atau peredaman noise.

2. Perbedaan Mean dan Median Filter, serta kapan digunakan

ASPEK	MEAN FILTER	MEDIAN FILTER
Prinsip kerja	Menghitung rata-rata dari nilai piksel dalam jendela (kernel)	Mengambil nilai tengah (median) dari nilai piksel dalam jendela
Efek pada Citra	Menghaluskan citra namun bisa membuat tepi menjadi kabur	Menghaluskan citra sambil tetap menjaga tepi objek
Efektif untuk	Mengurangi noise Gaussian (noise acak ringan)	Menghilangkan salt-and-pepper noise (bintik putih/hitam acak)
Kelemahan	Rentan terhadap outlier (nilai ekstrem)	Lebih mahal secara komputasi

3. Langkah – langkah operasi pada satu piksel:

- Pilih kernel/filter, misalnya matriks 3x3.
- Letakkan kernel di atas citra input sehingga pusat kernel tepat berada pada piksel target.
- Kalikan elemen kernel dengan elemen citra yang beririsan (posisi yang sama).
- Jumlahkan hasil perkalian tersebut untuk mendapatkan satu nilai tunggal.
- Simpan nilai tersebut di citra output pada posisi yang sama dengan piksel pusat.
- Geser kernel ke posisi selanjutnya, dan ulangi langkah di atas sampai seluruh piksel diproses.

4. Pentingnya Operasi Konvolusi dalam Pengolahan Citra Digital dan Deep Learning

Operasi konvolusi dianggap **fundamental** karena:

- a. **Meniru cara kerja sistem penglihatan manusia**, yaitu mengenali pola lokal seperti tepi, sudut, dan tekstur.
- b. Efisien untuk **ekstraksi fitur lokal** dari citra (misalnya: garis, bentuk, pola).
- c. Dalam **Deep Learning**, konvolusi menjadi inti dari proses pembelajaran karena:
 - Mengurangi jumlah parameter dibandingkan fully-connected layer.
 - Menjaga hubungan spasial antar piksel.
 - Mendeteksi fitur penting pada berbagai lapisan (low-level hingga high-level).

5. - **Peningkatan Citra (Image Enhancement):** Blur/Smoothing dan Sharpening

- **Deteksi Blur:** Edge Detection dan Corner Detection
- **Segmentasi Citra:** Memisahkan Objek dari latar belakang dengan filter tertentu
- **Pengenalan Wajah Objek:** Digunakan CNN untuk Mengekstrak fitur wajah atau objek
- **Kehidupan sehari-hari:** Kamera Smartphone (Otomatis memperbaiki kualitas gambar menggunakan filter konvolusi), CCTV (Deteksi gerakan dan pengenalan wajah), Mobil otonom (Memproses citra dari kamera untuk mengenali rambu lalu lintas, pejalan kaki, dll).

LAPORAN PRAKTIKUM

PENGOLAHAN CITRA DIGITAL



NAMA : Rama Dinantiar

NIM : 202331044

KELAS : Pengolahan Citra Digital

DOSEN : Ir. Darma Rusjdi, M.Kom

NO.PC : 21

ASISTEN : 1. Abdur Rasyid Ridho

2. Rizqy Amanda

3. Kashrina Masyid Azka

4. Izzat Islami Kagapi

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2025

Penjelasan Praktikum

Convolution

```
[48]: import cv2 as cv
import matplotlib.pyplot as plt
import numpy as np
# Rama Dinantiar (202331044)
```

Pada cell ini saya mengimpor tiga library penting:

- cv2 → digunakan untuk pengolahan citra seperti membaca gambar, mengubah warna, dan filtering.
- matplotlib.pyplot → dipakai untuk menampilkan gambar dan grafik.
- numpy → digunakan untuk membuat dan memanipulasi array, terutama untuk pembuatan kernel filter.

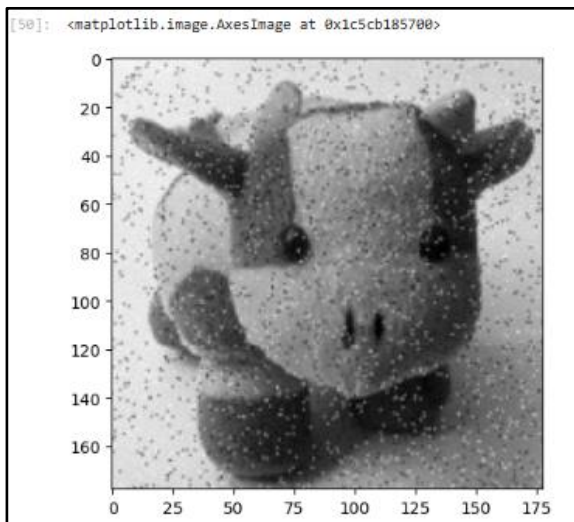
```
[50]: # Membaca Gambar dan Propertiny
img = cv.imread("boneka.jpg")
img_gray = cv.cvtColor(img, cv.COLOR_BGR2GRAY)

plt.imshow(img, cmap="gray")
plt.imshow(img_gray, cmap="gray")
# Rama Dinantiar (202331044)
```

Penjelasan:

- Gambar boneka.jpg dibaca dan dikonversi ke citra grayscale (img_gray) karena filter konvolusi lebih umum diterapkan pada citra hitam putih.
- Kemudian gambar asli dan hasil konversi grayscale ditampilkan (meskipun kedua imshow() akan menampilkan satu sama lain jika tidak dipisah subplotnya).

Output:



Membuat Kernel Deteksi Tepi (Filter Batas)

```
[53]: kernel = np.array(
      [
        [-1,0,-1],
        [0,4,0],
        [-1,0,-1]
      ]
    )

    img_output = cv.filter2D(img_gray, -1, kernel)

    fig, axs = plt.subplots(1,3, figsize=(20,5))
    ax = axs.ravel()

    ax[0].imshow(img_gray, cmap='gray')
    ax[0].set_title("Input")
    ax[0].axis('off')

    ax[1].imshow(img_output, cmap='gray')
    ax[1].set_title("Input")
    ax[1].axis('off')

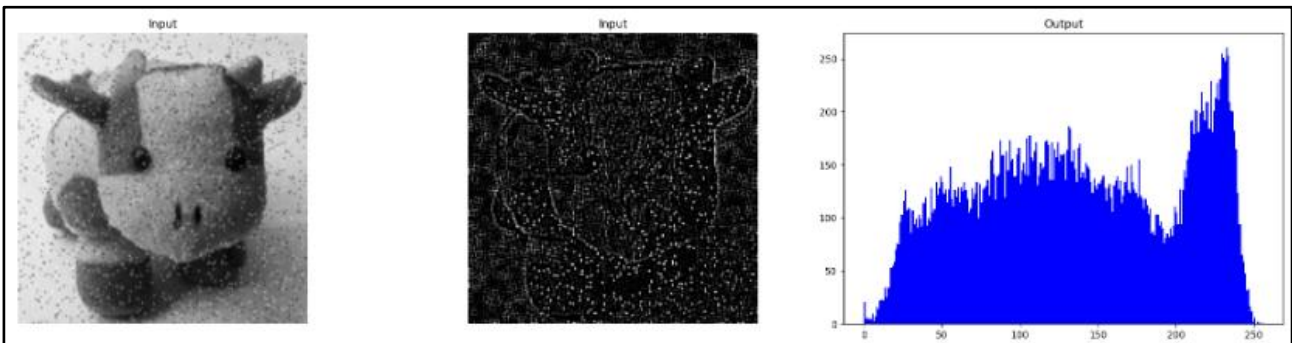
    ax[2].hist(img_gray.ravel(), bins=256, range=(0,256), color='blue')
    ax[2].set_title("Output")

    plt.tight_layout()
    plt.show()
    # Rama Dinantiar (202331044)
```

Penjelasan:

- Di sini kamu membuat **kernel deteksi tepi** (edge detection kernel).
- Fungsi filter2D menerapkan kernel ke gambar grayscale untuk menghasilkan img_output, yaitu citra dengan penekanan pada bagian-bagian yang memiliki perbedaan intensitas tinggi (biasanya pinggir objek).
- Disediakan juga visualisasi hasil konvolusi serta histogram dari citra awal.

Output:



Menambahkan Noise (Salt & Pepper, Gaussian, Speckle) dan Menampilkannya

```
[56]: from skimage.util import random_noise

#baca properti gambar
img2 = cv.imread("boneka.jpg")

#fungsi skimage untuk random noise
noise_img_snp = random_noise(img2, mode='s&p', amount=0.2)
noise_img_gaussian = random_noise(img2, mode='gaussian', mean=0, var=0.01)
noise_img_speckle = random_noise(img2, mode='speckle')

noise_img_snp = np.array(255*noise_img_snp, dtype='uint8')
noise_img_gaussian = np.array(255*noise_img_gaussian, dtype='uint8')
noise_img_speckle = np.array(255*noise_img_speckle, dtype='uint8')

#config plot dan axis untuk tampilan, menampilkan dalam versi plot dengan color map gray
fig, axs = plt.subplots(4,1, figsize=(10,40))
ax = axs.ravel()

ax[0].imshow(img2, cmap='gray')
ax[0].set_title('citra asli')

ax[1].imshow(noise_img_snp, cmap='gray')
ax[1].set_title('noise S&P')

ax[2].imshow(noise_img_gaussian, cmap='gray')
ax[2].set_title('noise gaussian')

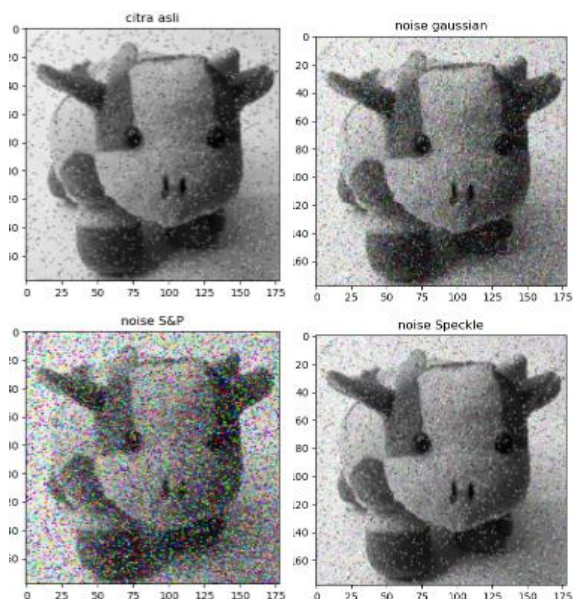
ax[3].imshow(noise_img_speckle, cmap='gray')
ax[3].set_title('noise Speckle')

plt.show()
# Rama Dinantiar (202331044)
```

Penjelasan:

- Menggunakan random_noise dari skimage.util, saya menambahkan tiga jenis noise ke gambar:
 - o Salt & Pepper: bintik hitam-putih acak.
 - o Gaussian: noise berbasis distribusi normal.
 - o Speckle: noise yang umum pada citra radar.
- Hasil noise dikonversi ke format uint8 agar dapat ditampilkan dengan imshow.
- Kemudian, gambar asli + hasil noise ditampilkan secara vertikal.

Output:



Membuat Kernel 3x3 untuk Filter Rata-Rata

```
[59]: #membuat kernel 3*3 yang bernilai 1/9
      # image reduction
      kernel_3_3 = np.ones((3,3), np.float32)/9
      # Rama Dinantiar (202331044)
```

Penjelasan:

- Di sini kamu membuat mean filter (3x3), yaitu filter yang rata-rata semua piksel di sekitarnya.
- Nilainya 1/9 agar hasil konvolusi merupakan nilai rata-rata dari 9 piksel dalam matriks 3x3.

Membuat Kernel 3x3 untuk Filter Rata-Rata

```
[59]: #membuat kernel 3*3 yang bernilai 1/9
      # image reduction
      kernel_3_3 = np.ones((3,3), np.float32)/9
      # Rama Dinantiar (202331044)
```

Filter Rata-Rata dan Median untuk Noise Salt & Pepper

```
[62]: #melakukan konvolusi untuk mengatasi noise salt and pepper dengan function filter 2D dan medianBlur
      img_snp_avg_filter = cv.filter2D(noise_img_snp, cv.CV_8U, kernel_3_3,
                                      (-1,1), delta=0,
                                      borderType = cv.BORDER_DEFAULT)
      img_snp_median_filter = cv.medianBlur(noise_img_snp, 3)
      # Rama Dinantiar (202331044)
```

Filter Rata-Rata dan Median untuk Noise Gaussian

```
[67]: #melakukan konvolusi untuk mengatasi noise gaussian dengan function filter 2D dan medianBlur
      img_gaussian_avg_filter = cv.filter2D(noise_img_gaussian, cv.CV_8U, kernel_3_3,
                                      (-1,1), delta=0,
                                      borderType = cv.BORDER_DEFAULT)
      img_gaussian_median_filter = cv.medianBlur(noise_img_speckle, 3)
      # Rama Dinantiar (202331044)
```

Filter Rata-Rata dan Median untuk Noise Speckle

```
[70]: #melakukan konvolusi untuk mengatasi noise speckle dengan function filter 2D dan medianBlur
      img_speckle_avg_filter = cv.filter2D(noise_img_speckle, cv.CV_8U, kernel_3_3,
                                      (-1,1), delta=0,
                                      borderType = cv.BORDER_DEFAULT)
      img_speckle_median_filter = cv.medianBlur(noise_img_speckle, 3)
      # Rama Dinantiar (202331044)
```

Penjelasan:

- Kamu menerapkan dua jenis filter untuk tiap jenis noise:
 1. filter2D → untuk mean filter
 2. medianBlur → untuk median filter

- Masing-masing digunakan untuk mengurangi noise:
 - Salt & Pepper cocok ditangani oleh median filter.
 - Gaussian cocok ditangani oleh mean filter.
 - Speckle dapat ditangani dengan keduanya (tergantung kebutuhan).

Menampilkan Hasil Filter pada Berbagai Jenis Noise

```
[73]: # menampilkan perbandingan hasil
fig, axs = plt.subplots(3,3, figsize = (15,15))
ax = axs.ravel()

# noise salt and pepper
ax[0].imshow(noise_img_snp)
ax[0].set_title('Noise S&P')

ax[1].imshow(img_snp_avg_filter)
ax[1].set_title('Average Filter S&P Reduction')

ax[2].imshow(img_snp_median_filter)
ax[2].set_title('Median Filter S&P Reduction')

# noise gaussian
ax[3].imshow(noise_img_gaussian)
ax[3].set_title('Noise Gaussian')

ax[4].imshow(img_gaussian_avg_filter)
ax[4].set_title('Average Filter Gaussian Reduction')

ax[5].imshow(img_gaussian_median_filter)
ax[5].set_title('Median Filter Gaussian Reduction')

# noise speckle
ax[6].imshow(noise_img_speckle)
ax[6].set_title('Noise Speckle')

ax[7].imshow(img_speckle_avg_filter)
ax[7].set_title('Average Filter Speckle Reduction')

ax[8].imshow(img_speckle_median_filter)
ax[8].set_title('Median Filter Speckle Reduction')
# Rama Dinantiar (202331044)
```

Penjelasan:

- Kamu membuat subplot 3x3 untuk menampilkan:
 - Kolom 1: Gambar noise
 - Kolom 2: Hasil filtering dengan average filter
 - Kolom 3: Hasil filtering dengan median filter
- Masing-masing baris menunjukkan satu jenis noise: S&P, Gaussian, dan Speckle.

Hasil ini sangat bagus untuk melihat perbandingan langsung seberapa efektif setiap filter dalam mereduksi jenis noise tertentu.

Output:

