

LAPORAN TEORI

PENGOLAHAN CITRA DIGITAL



NAMA : Alvin Krisna Fratama

NIM : 202331238

KELAS : C

DOSEN : Ir. Darma Rusjdi, M.Kom

NO.PC : 18

ASISTEN : 1. Abdur Rasyid Ridho

2. Rizqy Amanda

3. Kashrina Masyid Azka

4. Izzat Islami Kagapi

INSTITUT TEKNOLOGI PLN

SISTEM INFORMASI

2025

1. Operasi Konvolusi dalam Pengolahan Citra Digital

Operasi konvolusi adalah sebuah teknik matematis fundamental dalam pengolahan citra digital untuk memodifikasi atau mengekstrak informasi dari sebuah citra. Secara sederhana, konvolusi bekerja dengan "menggeser" sebuah matriks kecil yang disebut kernel (atau filter) ke seluruh bagian citra input. Pada setiap posisi, nilai piksel di bawah kernel akan dihitung ulang berdasarkan nilai-nilai dalam kernel tersebut untuk menghasilkan satu nilai piksel baru pada citra output. Proses ini memungkinkan berbagai macam operasi, seperti penajaman (sharpening), penghalusan (smoothing/blurring), deteksi tepi (edge detection), dan banyak lagi, tergantung pada nilai-nilai yang ada di dalam kernel yang digunakan.

2. Perbedaan Filter Rata-rata (Mean) dan Filter Median

Meskipun keduanya digunakan untuk penghalusan citra dan reduksi noise, filter rata-rata dan filter median memiliki perbedaan mendasar dalam cara kerja dan efektivitasnya.

Fitur	Filter Rata-rata (Mean Filter)	Filter Median (Median Filter)
Prinsip Kerja	Mengganti nilai piksel pusat dengan nilai rata-rata (average) dari semua piksel yang berada di bawah kernel.	Mengganti nilai piksel pusat dengan nilai tengah (median) dari semua piksel yang berada di bawah kernel setelah diurutkan.
Efek pada Citra	Menghasilkan efek blur yang lebih halus dan seragam.	Cenderung mempertahankan ketajaman tepi lebih baik daripada filter rata-rata.
Sensitivitas	Sangat rentan terhadap noise ekstrem seperti noise garam & merica (salt-and-pepper) karena nilai ekstrem (hitam/putih) akan sangat memengaruhi hasil rata-rata.	Sangat efektif untuk menghilangkan noise garam & merica karena nilai ekstrem tersebut akan terabaikan saat nilai median diambil.

3. Langkah-langkah Proses Konvolusi pada Satu Piksel

Berikut adalah rincian langkah demi langkah proses konvolusi untuk menghitung nilai satu piksel output: Pilih Piksel Target, Posisikan Kernel, Operasi Perkalian Elemen-demi-Elemen, Jumlahkan Hasil Perkalian, Hasil Akhir (Nilai Piksel Output). Proses ini diulangi untuk setiap piksel pada citra input untuk menghasilkan citra output yang telah terkonvolusi sepenuhnya.

4. Signifikansi Konvolusi dalam Pengolahan Citra dan Deep Learning (CNN)

Dalam Pengolahan Citra: Dengan merancang kernel yang berbeda, kita dapat secara spesifik menargetkan dan memanipulasi fitur-fitur citra seperti tepi, sudut, tekstur, atau bahkan menghilangkan noise. Ini adalah dasar dari hampir semua teknik analisis dan filter citra tingkat rendah.

Di dalam *Convolutional Neural Networks* (CNN), **kernel filter tidak dirancang manual**, melainkan **dipelajari secara otomatis** oleh jaringan selama proses *training*. Jaringan akan menciptakan filternya sendiri untuk mendeteksi fitur paling penting, mulai dari yang sederhana (garis, warna) hingga kompleks (bentuk, tekstur).

5. Aplikasi dan Tujuan Utama Operasi Konvolusi

Tujuan utama konvolusi adalah untuk **memfilter** dan **mengekstrak fitur** dari citra.

LAPORAN PRAKTIKUM

PENGOLAHAN CITRA DIGITAL



NAMA : Alvin Krisna Fratama

NIM : 202331238

KELAS : C

DOSEN : Ir. Darma Rusjdi, M.Kom

NO.PC : 18

ASISTEN : 1. Abdur Rasyid Ridho

2. Rizqy Amanda

3. Kashrina Masyid Azka

4. Izzat Islami Kagapi

INSTITUT TEKNOLOGI PLN

SISTEM INFORMASI

2025

```
#20233123_Alvin Krisna Fratama
import cv2
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['figure.figsize'] = [15, 10]
```

- `import cv2`: Mengimpor library OpenCV, yang menyediakan semua fungsi utama untuk pemrosesan citra.
- `import numpy as np`: Mengimpor library NumPy dan memberinya alias `np`. NumPy sangat penting untuk operasi numerik, terutama untuk membuat matriks kernel filter.
- `import matplotlib.pyplot as plt`: Mengimpor modul pyplot dari library Matplotlib dengan alias `plt`. Modul ini digunakan untuk membuat visualisasi data, termasuk menampilkan gambar di notebook.
- `plt.rcParams['figure.figsize'] = [15, 10]`: Baris ini bersifat opsional. Fungsinya adalah untuk mengatur ukuran default dari setiap gambar (figure) yang akan ditampilkan oleh Matplotlib agar lebih besar dan mudah dilihat.

```
#202331238_Alvin Krisna Fratama

# 1. Memuat gambar
img_makanan = cv2.imread('Makanan_Sulawesi.jpg')
img_makanan_rgb = cv2.cvtColor(img_makanan, cv2.COLOR_BGR2RGB)

# 2. Aplikasi Filter Rata-rata (Mean Filter)
kernel_rata_rata = np.ones((5, 5), np.float32) / 25
makanan_rata_rata = cv2.filter2D(img_makanan_rgb, -1, kernel_rata_rata)

# 3. Aplikasi Filter Median
makanan_median = cv2.medianBlur(img_makanan_rgb, 5)

# 4. Aplikasi Filter Batas (Edge Detection) menggunakan Laplacian
kernel_batas = np.array([[0, -1, 0],
                          [-1, 4, -1],
                          [0, -1, 0]])

# -----

# 5. Aplikasikan filter ke gambar grayscale
img_makanan_gray = cv2.cvtColor(img_makanan, cv2.COLOR_BGR2GRAY)
makanan_batas = cv2.filter2D(img_makanan_gray, -1, kernel_batas)
```

- `img_makanan = cv2.imread(...)`: Membaca file gambar `Makanan_Sulawesi.webp` dari disk dan menyimpannya sebagai matriks NumPy ke dalam variabel `img_makanan`. Penting: OpenCV memuat gambar dengan urutan channel warna **BGR** (Blue, Green, Red).
- `img_makanan_rgb = cv2.cvtColor(...)`: Mengubah urutan channel warna dari BGR ke **RGB** (Red, Green, Blue). Ini langkah krusial agar gambar dapat ditampilkan dengan warna yang benar oleh Matplotlib.
- `kernel_rata_rata = np.ones((5, 5), np.float32) / 25`: Membuat kernel untuk filter rata-rata. `np.ones((5, 5), ...)` menciptakan matriks 5x5 yang semua elemennya bernilai 1. Kemudian, setiap elemen dibagi 25 (yaitu 5x5) agar total nilai semua elemen kernel adalah 1. Ini menghasilkan efek *blur* dengan merata-ratakan piksel.
- `makanan_rata_rata = cv2.filter2D(...)`: Ini adalah fungsi konvolusi utama. Fungsi ini menerapkan `kernel_rata_rata` ke seluruh `img_makanan_rgb` untuk menghasilkan citra yang sudah di-blur.
- `makanan_median = cv2.medianBlur(...)`: Menerapkan filter median. Berbeda dari `filter2D`, fungsi ini secara spesifik mencari nilai tengah (median) dari piksel-piksel di sekitarnya (dalam area 5x5). Sangat efektif untuk menghilangkan noise tipe *salt-and-pepper*.
- `kernel_batas = np.array(...)`: Membuat kernel Laplacian secara manual. Kernel ini dirancang untuk mendeteksi tepi atau garis pada citra.

- `img_makanan_gray = cv2.cvtColor(...)`: Mengubah citra asli menjadi grayscale (skala keabuan). Deteksi tepi seringkali lebih efektif dilakukan pada citra grayscale.
- `makanan_batas = cv2.filter2D(...)`: Menerapkan kernel batas (Laplacian) pada citra grayscale untuk menghasilkan citra baru yang hanya menonjolkan bagian tepinya.

```
# 202331238_Alvin Krisna Fratama

# Menyiapkan subplot untuk menampilkan semua gambar
fig, axs = plt.subplots(2, 2, figsize=(12, 12))

axs[0, 0].imshow(img_makanan_rgb)
axs[0, 0].set_title('1. Citra Asli (Makanan Sulawesi)')
axs[0, 0].axis('off') # Menghilangkan sumbu x dan y

axs[0, 1].imshow(makanan_rata_rata)
axs[0, 1].set_title('2. Hasil Filter Rata-rata (Blur)')
axs[0, 1].axis('off')

axs[1, 0].imshow(makanan_median)
axs[1, 0].set_title('3. Hasil Filter Median (Penghilang Noise)')
axs[1, 0].axis('off')

axs[1, 1].imshow(makanan_batas, cmap='gray')
axs[1, 1].set_title('4. Hasil Filter Batas (Deteksi Tepi)')
axs[1, 1].axis('off')

# Menampilkan plot
plt.tight_layout()
plt.show()
```

- `fig, axs = plt.subplots(...)`: Membuat sebuah "figure" (kanvas) dan sebuah grid "subplot" (area gambar) berukuran 2 baris dan 2 kolom untuk menampung 4 gambar.
- `axs[0, 0].imshow(...)`: Menampilkan citra `img_makanan_rgb` pada subplot pertama (posisi baris 0, kolom 0).
- `axs[0, 0].set_title(...)`: Memberikan judul pada subplot tersebut.
- `axs[0, 0].axis('off')`: Menghilangkan penanda sumbu (sumbu x dan y) agar tampilan lebih bersih.
- Baris-baris berikutnya melakukan hal yang sama untuk menampilkan hasil filter rata-rata, median, dan batas pada subplot masing-masing.
- `plt.show()`: Perintah akhir untuk merender dan menampilkan keseluruhan figure beserta semua subplotnya.

1. Citra Asli (Makanan Sulawesi)



2. Hasil Filter Rata-rata (Blur)



3. Hasil Filter Median (Penghilang Noise)



4. Hasil Filter Batas (Deteksi Tepi)



```
# 202331238_Alvin krisna Fratama
img_boneka = cv2.imread('boneka.jpg')
img_boneka_rgb = cv2.cvtColor(img_boneka, cv2.COLOR_BGR2RGB)
kernel_sharpen = np.array([[ -1, -1, -1],
                           [ -1,  9, -1],
                           [ -1, -1, -1]])

# Mengaplikasikan konvolusi dengan kernel 2D
boneka_sharpened = cv2.filter2D(img_boneka_rgb, -1, kernel_sharpen)
```

- `kernel_sharpen = np.array(...)`: Di sini, kita mendefinisikan kernel penajaman (*sharpening*) secara manual. Nilai 9 yang besar di tengah dikelilingi oleh nilai -1 akan memperkuat perbedaan antara piksel pusat dengan piksel tetangganya, sehingga menghasilkan efek visual yang lebih tajam.
- `boneka_sharpened = cv2.filter2D(...)`: Sama seperti sebelumnya, fungsi ini menerapkan `kernel_sharpen` pada citra `boneka` untuk mendapatkan versi yang lebih tajam.

```
# 202331238_Alvin Krisna F

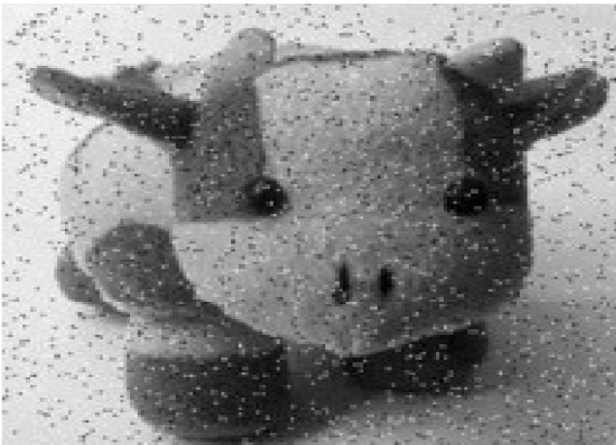
# Menyiapkan subplot untuk menampilkan gambar asli dan hasil
fig, axs = plt.subplots(1, 2, figsize=(12, 6))

axs[0].imshow(img_boneka_rgb)
axs[0].set_title('1. Citra Asli (Boneka Sapi)')
axs[0].axis('off')

axs[1].imshow(boneka_sharpened)
axs[1].set_title('2. Hasil Konvolusi Filter 2D (Sharpen)')
axs[1].axis('off')

plt.tight_layout()
plt.show()
```

1. Citra Asli (Boneka Sapi)



2. Hasil Konvolusi Filter 2D (Sharpen)

