LAPORAN TEORI

PENGOLAHAN CITRA DIGITAL

NAMA : Kurniawan Wibisono

NIM : 202331187

KELAS : C

DOSEN : Ir. Darma Rusjdi, M.Kom

NO.PC :

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:** Operasi konvolusi adalah proses matematis di mana sebuah *kernel* (matriks kecil) digeser di atas citra masukan untuk menghasilkan citra keluaran. Setiap piksel keluaran dihitung dari penjumlahan perkalian antara nilai piksel di sekitar area *kernel* dengan nilai-nilai dalam *kernel* itu sendiri.

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

   - **Filter Rata-rata:** Mengganti nilai piksel pusat dengan rata-rata nilai piksel di sekitarnya. Efektif untuk mengurangi *noise* Gaussian (distribusi normal), namun dapat menyebabkan citra menjadi blur.
   - **Filter Median:** Mengganti nilai piksel pusat dengan nilai median dari piksel-piksel di sekitarnya. Sangat efektif untuk mengurangi *noise* salt-and-pepper (berupa bintik hitam dan putih), dan lebih baik dalam mempertahankan detail tepi citra dibandingkan filter rata-rata.

3. **Langkah-langkah Proses Konvolusi untuk Satu Piksel Output:**

   - Pilih posisi piksel pada citra input untuk dihitung.
   - Tempatkan *kernel* di atas piksel tersebut, sehingga pusat *kernel* sejajar dengan piksel yang akan dihitung.
   - Kalikan setiap nilai piksel di bawah *kernel* dengan nilai *kernel* yang bersesuaian.
   - Jumlahkan semua hasil perkalian tersebut.
   - Nilai hasil penjumlahan ini adalah nilai piksel output untuk posisi tersebut.

4. **Signifikansi dan Kegunaan Operasi Konvolusi dalam Pengolahan Citra dan Deep Learning (CNNs):** Operasi konvolusi dianggap fundamental karena kemampuannya untuk mengekstraksi fitur dari citra. Dalam pengolahan citra tradisional, digunakan untuk *blurring*, *sharpening*, deteksi tepi, dll. Dalam Deep Learning (khususnya CNNs), operasi konvolusi memungkinkan model secara otomatis mempelajari hierarki fitur dari data, mulai dari fitur tingkat rendah (tepi, sudut) hingga fitur tingkat tinggi (objek, pola kompleks), yang sangat penting untuk tugas-tugas seperti klasifikasi gambar dan deteksi objek.

5. **Aplikasi Utama Operasi Konvolusi dalam Pengolahan Citra Digital:**

   - Peningkatan Citra: Mempertajam citra, mengurangi *noise*. Contoh: Meningkatkan kualitas foto yang buram atau berbintik.
   - Deteksi Tepi: Mengidentifikasi batas-batas objek dalam citra. Contoh: Digunakan dalam sistem pengenalan wajah atau analisis citra medis untuk mengidentifikasi tumor.
   - Ekstraksi Fitur: Mengidentifikasi pola atau karakteristik penting dalam citra. Contoh: Dalam bidang keamanan, digunakan untuk mendeteksi anomali pada rekaman CCTV.
   - Pengenalan Pola/Objek: Bagian integral dari sistem pengenalan objek otomatis. Contoh: Mobil tanpa pengemudi menggunakan konvolusi untuk mengidentifikasi rambu lalu lintas, pejalan kaki, dan kendaraan lain.

LAPORAN PRAKTIKUM

PENGOLAHAN CITRA DIGITAL

NAMA : Kurniawan Wibisono

NIM : 202331187

KELAS : C

DOSEN : Ir. Darma Rusjdi, M.Kom

NO.PC :

ASISTEN : 1. Abdur Rasyid Ridho

2. Rizqy Amanda

3. Kashrina Masyid Azka

4. Izzat Islami Kagapi

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2025

TAHAPAN PROGRAM

```
#Kurniawan Wibisono_202331187
import cv2
import numpy as np
import matplotlib.pyplot as plt

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

✓ 0.5s

Baris ini mengimpor tiga *library* Python yang esensial untuk pengolahan citra dan visualisasi:

- cv2: Ini adalah *library* OpenCV (Open Source Computer Vision Library), yang menyediakan berbagai fungsi untuk pengolahan citra dan visi komputer.
- numpy: Ini adalah *library* fundamental untuk komputasi numerik di Python. Ia sangat penting untuk bekerja dengan array (matriks) yang digunakan untuk merepresentasikan citra.
- matplotlib.pyplot: Ini adalah modul dari *library* Matplotlib yang digunakan untuk membuat plot dan visualisasi data, termasuk menampilkan gambar.

```
#Kurniawan Wibisono_202331187

# 1. Memuat gambar
img_pesawat = cv2.imread('pesawat.jpeg')
img_pesawat_rgb = cv2.cvtColor(img_pesawat, cv2.COLOR_BGR2RGB)

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

# 3. Aplikasi Filter Median
pesawat_median = cv2.medianBlur(img_pesawat_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_pesawat_gray = cv2.cvtColor(img_pesawat, cv2.COLOR_BGR2GRAY)
pesawat_batas = cv2.filter2D(img_pesawat_gray, -1, kernel_batas)
```

✓ 0.0s

Penjelasan:

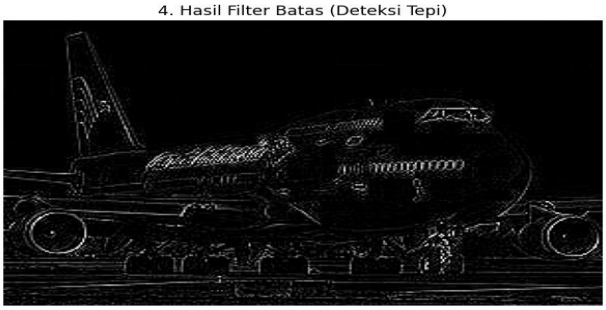
1. **Memuat Citra**:

- cv2.imread('pesawat.jpeg'): Membaca file citra pesawat.jpeg ke dalam variabel img_pesawat. OpenCV membaca citra dalam format BGR (Blue, Green, Red).
- cv2.cvtColor(img_pesawat, cv2.COLOR_BGR2RGB): Mengubah format warna dari BGR ke RGB, karena Matplotlib menggunakan format RGB untuk menampilkan citra. Hasilnya disimpan di img_pesawat_rgb.

2. **Filter Rata-rata (Mean Filter)**:
   - kernel_rata_rata = np.ones((5, 5), np.float32) / 25: Membuat kernel (matriks) berukuran 5x5 yang berisi nilai 1, lalu dibagi 25 untuk menormalkan (rata-rata). Kernel ini digunakan untuk filter rata-rata, yang menghaluskan citra dengan mengganti setiap piksel dengan rata-rata piksel tetangganya.
   - cv2.filter2D(img_pesawat_rgb, -1, kernel_rata_rata): Menerapkan filter rata-rata pada citra RGB menggunakan fungsi filter2D. Parameter -1 menjaga kedalaman warna citra asli. Hasilnya adalah citra yang lebih halus, disimpan di pesawat_rata_rata.
3. **Filter Median**:
   - cv2.medianBlur(img_pesawat_rgb, 5): Menerapkan filter median dengan ukuran kernel 5x5 pada citra RGB. Filter ini mengganti setiap piksel dengan nilai median dari piksel tetangganya, efektif untuk menghilangkan noise (seperti noise salt-and-pepper) tanpa mengaburkan tepi terlalu banyak. Hasilnya disimpan di pesawat_median.
4. **Filter Batas (Edge Detection) dengan Laplacian**:
   - kernel_batas = np.array([[0, -1, 0], [-1, 4, -1], [0, -1, 0]]): Mendefinisikan kernel Laplacian berukuran 3x3 untuk mendeteksi tepi (edge detection). Kernel ini menonjolkan perubahan intensitas piksel, yang menunjukkan tepi dalam citra.
   - Catatan: Kode tampaknya terpotong, tetapi kernel ini biasanya digunakan untuk mendeteksi tepi dengan menonjolkan perbedaan antara piksel pusat dan tetangganya.
5. **Menerapkan Filter pada Citra Grayscale**:
   - cv2.cvtColor(img_pesawat, cv2.COLOR_BGR2GRAY): Mengubah citra BGR menjadi grayscale (skala abu-abu) untuk mempermudah deteksi tepi, karena filter Laplacian biasanya diterapkan pada citra grayscale. Hasilnya disimpan di img_pesawat_gray.
   - cv2.filter2D(img_pesawat_gray, -1, kernel_batas): Menerapkan kernel Laplacian pada citra grayscale untuk mendeteksi tepi. Hasilnya adalah citra yang menonjolkan tepi, disimpan di pesawat_batas.

```python
#Kurniawan Wibisono_202331187

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

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

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

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

axs[1, 1].imshow(pesawat_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()
```

✓ 0.3s



```
#Kurniawan Wibisono_202331187

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)
```

✓ 0.0s

Penjelasan Kernel Penajaman: Kernel seperti di atas menonjolkan piksel pusat (bobot 5) dan mengurangi kontribusi tetangga (bobot -1). Total bobot kernel adalah 1 (0 - 1 - 1 + 5 - 1 - 1 + 0 = 1), yang menjaga kecerahan citra sambil meningkatkan kontras tepi. Ini membuat detail seperti tekstur atau garis pada boneka sapi lebih tajam.

```
#Kurniawan Wibisono_202331187

# 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()
```

✓ 0.0s

Sel ini bertujuan untuk menampilkan dua citra: citra asli boneka sapi dan citra yang telah ditajamkan (sharpened). Namun, ada masalah karena variabel img_boneka_rgb dan boneka_sharpened tidak didefinisikan dalam kode yang diberikan. Berikut adalah penjelasan langkah-langkah dan asumsi berdasarkan konteks:

1. **Membuat Subplot**:
    - fig, axs = plt.subplots(1, 2, figsize=(12, 6)): Membuat figure dengan dua subplot dalam tata letak 1 baris dan 2 kolom. Ukuran figure adalah 12 inci lebar dan 6 inci tinggi, lebih kecil dari pengaturan default di Sel 1 (15x10), menunjukkan bahwa ukuran ini diatur khusus untuk visualisasi ini.

- fig adalah objek figure Matplotlib, dan axs adalah array yang berisi dua objek subplot (axs[0] dan axs[1]).

2. **Menampilkan Citra Asli**:

   - axs[0].imshow(img_boneka_rgb): Menampilkan citra boneka sapi dalam format RGB pada subplot pertama. Variabel img_boneka_rgb diasumsikan sebagai citra yang telah dimuat (mungkin dengan cv2.imread dan dikonversi ke RGB dengan cv2.cvtColor), serupa dengan img_pesawat_rgb di Sel 2.

   - axs[0].set_title('1. Citra Asli (Boneka Sapi)'): Menambahkan judul "1. Citra Asli (Boneka Sapi)" pada subplot pertama.

   - axs[0].axis('off'): Menyembunyikan sumbu (axis) pada subplot pertama, sehingga hanya citra yang terlihat tanpa angka sumbu atau garis koordinat.

3. **Menampilkan Citra yang Ditajamkan**:

   - axs[1].imshow(boneka_sharpened): Menampilkan citra boneka sapi yang telah ditajamkan pada subplot kedua. Variabel boneka_sharpened diasumsikan sebagai hasil penerapan filter penajaman menggunakan cv2.filter2D dengan kernel penajaman.

   - axs[1].set_title('2. Hasil Konvolusi Filter 2D (Sharpen)'): Menambahkan judul "2. Hasil Konvolusi Filter 2D (Sharpen)" pada subplot kedua.

   - axs[1].axis('off'): Menyembunyikan sumbu pada subplot kedua.

4. **Mengatur Tata Letak dan Menampilkan**:

   - plt.tight_layout(): Mengatur tata letak subplot secara otomatis untuk mencegah tumpang tindih antara subplot atau dengan judul.

   - plt.show(): Menampilkan figure dengan kedua subplot ke layar.