LAPORAN TEORI

Pemrosesan Citra Digital

SOFTWARE ENGINEERING
LABORATORY

NAMA : Miqdad Hilya Hasan

NIM : 202331085

KELAS : A

DOSEN :

NO.PC :

ASISTEN : 1. Sasikirana Ramadhanty Setiawan Putri

2. Davina Najwa Ermawan

3. Viana Salsabila Fairuz Syahla

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2024/2025

1. Jelaskan konsep operasi konvolusi menurut pemahaman kalian!

   Konvolusi adalah operasi matematika yang umum digunakan dalam pemrosesan citra untuk memodifikasi atau mengekstraksi informasi dari gambar. Konvolusi bekerja dengan menggeser kernel (filter) melintasi seluruh piksel gambar dan menghitung nilai baru setiap piksel berdasarkan hasil perkalian antara nilai piksel dan elemen kernel yang bersesuaian.

   Secara sederhana, proses ini seperti "menyapu" gambar dengan filter untuk menghasilkan gambar baru yang menyoroti fitur tertentu seperti tepi, sudut, atau blur. Prosesnya melibatkan:

   - Matriks input (gambar),
   - Matriks kernel (biasanya 3×3 atau 5×5),
   - Operasi perkalian elemen-matriks dan penjumlahan hasilnya untuk setiap posisi kernel.

2. Jelaskan perbedaan mendasar antara filter rata-rata (mean filter), filter median (median filter), dan filter batas dalam operasi konvolusi!

Jenis Filter	Penjelasan	Fungsi Utama	Kelebihan	Kekurangan
Mean	Mengganti nilai piksel dengan rata-rata dari tetangganya.	Mereduksi noise Gaussian.	Sederhana, efektif untuk noise kecil	Mengaburkan tepi (blurring)
Median	Mengganti nilai piksel dengan nilai median dari tetangganya.	Menghilangkan noise impuls.	Menjaga tepi, efektif untuk salt & pepper noise	Lebih kompleks dari mean filterv
Batas	Digunakan untuk mengekstrak tepi dalam gambar dengan menyorot perubahan intensitas.	Deteksi tepi gambar.	Membantu segmentasi dan analisis objek	Rentan terhadap noise, memerlukan preprocessing

3. Jelaskan cara kerja dan fungsi kernel pada codingan yang sudah diajarkan sebelumnya!

   Kernel adalah matriks kecil (misalnya 3x3 atau 5x5) yang digunakan untuk memproses citra melalui konvolusi. Fungsinya tergantung pada nilai-nilai di dalam kernel tersebut. Dalam pemrograman (misalnya dengan OpenCV atau NumPy), kernel diterapkan dengan cara berikut:

   - Langkah kerja:

     a. Kernel diletakkan di atas bagian gambar.
     b. Dilakukan operasi perkalian elemen-matriks antara kernel dan nilai piksel.
     c. Hasilnya dijumlahkan dan menjadi nilai baru piksel tengah.
     d. Kernel digeser satu piksel (horizontal atau vertikal) dan proses diulang.

- Contoh kode dengan OpenCV

```
import cv2

import numpy as np



image = cv2.imread('image.jpg', 0)

kernel = np.array([[0, -1, 0],

                   [-1, 5,-1],

                   [0, -1, 0]])

output = cv2.filter2D(image, -1,
kernel)
```

Pada contoh di atas, kernel berfungsi sebagai sharpening filter, menonjolkan detail gambar.

4. Apa perbedaan antara operator deteksi tepi Sobel, Prewitt, dan Canny? Jelaskan keunggulan dan kelemahan masing-masing.

Operator	Penjelasan	Kelebihan	Kekurangan
Sobel	Menggunakan derivatif pertama dengan bobot lebih besar di tengah (bias arah tertentu).	Lebih tahan terhadap noise dibanding Prewitt.	Kurang akurat pada tepi halus.
Prewitt	Mirip Sobel, namun bobot arah sama	Lebih sederhana dan cepat.	Lebih senditif terhadap noise
Canny	Menggunakan kombinasi filter Gaussian, deteksi gradien, non-max suppression, dan threshold ganda.	Akurat, deteksi tepi tipis dan bersih, tahan noise.	Kompleks, butuh parameter tuning.

5. Apa perbedaan antara transformasi translasi, rotasi, dan skala dalam transformasi geometrik citra? Berikan contohnya masing-masing.

Transformasi	Deskripsi	Contoh Aplikasi
Translasi	Menggeser seluruh citra ke posisi baru tanpa merubah bentuk.	Memindahkan objek dalam citra ke lokasi lain.
Rotasi	Memutar citra terhadap suatu titik pusat (biasanya tengah gambar).	Rotasi wajah atau logo dalam sistem pendeteksi.

Skala	Mengubah ukuran citra (membesar/memperkecil).	Zoom in/out citra atau resizing dataset.

Contoh Kode dengan OpenCV:

```
# Translasi

M = np.float32([[1, 0, 50], [0, 1, 30]])  # Geser 50 piksel ke kanan, 30 ke bawah

translated = cv2.warpAffine(image, M, (width, height))
```

6. Apa yang dimaksud dengan transformasi afine dalam transformasi geometrik? Sebutkan minimal dua contohnya dalam aplikasi pemrosesan citra.

   Transformasi afine adalah transfgormasi linier yang mempertahankan garis lurus dan paralelitas, tapi bisa mengubah ukuran, rotasi, posisi, dan kemiringan objek. Persamaan umum nya:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} tx \\ ty \end{bmatrix}$$

   Contoh aplikasi:

   a. Pemetaan koordinat dalam sistem kamera (georektifikasi).
   b. Koreksi perspektif pada dokumen yang difoto miring.
   c. Normalisasi wajah (face alignment) agar sejajar.

LAPORAN PRAKTIKUM

Pemrosesan Citra Digital

SOFTWARE ENGINEERING
LABORATORY

NAMA : Miqdad Hilya

NIM : 202331012

KELAS : A

DOSEN :

NO.PC :

ASISTEN : 1. Sasikirana Ramadhanty Setiawan Putri

2. Davina Najwa Ermawan

4. Viana Salsabila Fairuz Syahla

INSTITUT TEKNOLOGI PLN

TEKNIK INFORMATIKA

2024/2025

A. Olah gambar yang sama dengan gambar laporan 1 menggunakan metode-metode yang telah dipraktikkan sebelumnya (praktikum ketetanggaan piksel, filter rata-rata, filter median, dan filter batas).

1. Langkah pertama yang akan say alakukan adalah meng import semua library yang di perlukan

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
#202331085_Miqdad Hilya Hasan
```

2. Kemudian saya akan emngimport citra yagn akah di olah

```
img = cv2.imread('harimau_siberia.jpg')
#202331085_Miqdad Hilya Hasan
```

3. Setelah itu, citra tersebut saya ubah channel warna nya dari RGB ke GRAY dengan library cv2 menggunakan fungsi cvtColor

```
img = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
#202331085_Miqdad Hilya Hasan
```

4. Kemudian saya mempraktikkan ketetanggan piksel dengan mengubah tipe data pada citra menjadi float sehingga akurasi perhitungan pada data menjadi lebih akurat, kemudian membuat matriks kosong yang nilai nya terikat pada nilai dari citra tersebut

```
Tetangga = img.copy().astype(float)

m1,n1 = Tetangga.shape
output1 = np.empty([m1,n1])

print("Shape Copy Citra : ", Tetangga.shape)
print("Shape Ouptput1 : ", output1.shape)

print('m1 : ', m1)
print('n1 : ', n1)
#202331085_Miqdad Hilya Hasan
```

Output:

```
Shape Copy Citra :  (1371, 1920)
Shape Ouptput1 :  (1371, 1920)
m1 :  1371
n1 :  1920
```

5. Kemudian saya akan mempraktekkan filter rata rata yang berfungsi untuk mengurangi noise dengan cara mendeteksi dan menghitung jumlah dari suatu pixel dan 8 pixel di sekitar nya yang kemudian di kalikan 1/9 untuk mendapatkan rata rata nilai yang akan di simpan pada array output1

```
for baris in range (0, m1-1):
    for kolom in range (0, n1-1):
        a = baris
        b = kolom
        jumlah = Tetangga[a-1, b-1] + Tetangga[a-1,b] + Tetangga[a-1, b+1] +\
                 Tetangga[a, b-1] + Tetangga[a, b] + Tetangga[a,b+1] +\
                 Tetangga[a+1, b-1] + Tetangga[a+1, b] + Tetangga[a+1, b+1]
        output1[a,b] = 1/9*jumlah
```

6. Selanjutnya, saya akan membuat subplots yang akan menampilkan hasil dari filter beserta citra sebelum di olah

```
fig, axs = plt.subplots(2,1, figsize=(20,20))
ax = axs.ravel()

ax[0].imshow(img, cmap='gray')
ax[0].set_title('Input Citra')

ax[1].imshow(output1, cmap='gray')
ax[1].set_title('Output filter rata-rata')

plt.show()
#202331085_Miqdad Hilya Hasan
```

Output:

7. Kemudian, saya akan menerapkan median blur pada citra untuk menguraangi noise menggunakan fungsi medianBlur di library cv2

```
img_median = img.copy()
img_median = cv2.cvtColor(img_median, cv2.COLOR_BGR2RGB)
img_median_after = cv2.medianBlur(img_median, 31)

fig, axs = plt.subplots(2,1, figsize = (20,20))
ax = axs.ravel()

ax[0].imshow(img, cmap='gray')
ax[0].set_title('Gambar Aseli')

ax[1].imshow(img_median_after, cmap='gray')
ax[1].set_title('Ouptut Filter Median')

plt.show()
#202331085_Miqdad Hilya Hasan
```

8. Selanjutnya, saya akan membuat subplots yang akan menampilkan hasil dari filter beserta citra sebelum di olah. Saya menetapkan nilai intensitas blur sebesar 31 (nilaii harus ganjil) yang ada pada sintaks: img_median_after = cv2.medianBlur(img_median, 31)

```
img_median = img.copy()
img_median = cv2.cvtColor(img_median, cv2.COLOR_BGR2RGB)
img_median_after = cv2.medianBlur(img_median, 31)

fig, axs = plt.subplots(2,1, figsize = (20,20))
ax = axs.ravel()

ax[0].imshow(img, cmap='gray')
ax[0].set_title('Gambar Aseli')

ax[1].imshow(img_median_after, cmap='gray')
ax[1].set_title('Ouptut Filter Median')

plt.show()
#202331085_Miqdad Hilya Hasan
```

Output:

9. Kemudian say akaan mengaplkikasikan filter batas untuk meminimalkan noise salt and pepper di lakukan dengan meng copy image original dan mengubah tipe datanya menjadi float dan memasukkan dimensi yang tipe data nya sudah berubah tersebut ke dalam array dengan nested loop yang juga akan membuat array ketetanggan pixel dan emnemukan nilai maksimum dan minimum pada tiap piksel di ketetanggan tersebut. Setelah di temukan, jika nilai suatu piksel kurang dari minimum, Ganti dengan nilai minimum, jike nilai suatu piksel lebih besar dari maksikum, Ganti nilai piksel tersebut dengan nilai maksimum.

```
citra1 = img.copy().astype(float)

m1,n1 = citra1.shape
output1 = np.empty([m1,n1])
for baris in range (0, m1-1):
    for kolom in range (0, n1-1):
        a = baris
        b = kolom
        arr = np.array ([citra1[a-1,b-1], citra1[a-1,b], citra1[a-1,b+1],
                         citra1[a,b-1], citra1[a,b], citra1[a,b+1],
                         citra1[a+1,b-1], citra1[a+1,b], citra1[a+1,b+1]])

        minPiksel = np.amin(arr)
        maxPiksel = np.amax(arr)

        if citra1[baris, kolom] < minPiksel:
            ouptut[baris, kolom] = minPiksel
        else:
            if citra1[baris, kolom] > maxPiksel:
                ouptput1[baris, kolom] = macPiksel
            else:
                output1[baris, kolom] = citra1[baris, kolom]

fig, axs = plt.subplots(2,1, figsize = (20,20))
ax = axs.ravel()

ax[0].imshow(img, cmap='gray')
ax[0].set_title('Gambar Aseli')

ax[1].imshow(output1, cmap='gray')
ax[1].set_title('Ouptut Filter Batas')

plt.show()
#202331085_Miqdad Hilya Hasan
```

Output:

B. Olah gambar yang sama dengan gambar laporan 1 menggunakan metode-metode yang telah dipraktikkan sebelumnya (praktikum konvolusi, noise snp, noise gaussian, dan noise speckle).

1. Langkah pertama yang akan saya lakukan adalah import library yang akan saya gunakan

```
import cv2
import numpy as np
import matplotlib.pyplot as plt
#Miqdad Hilya Hasan_202331085
```

2. Kemudian, saya akan meng import citra yang akan di olah dan mengganti channel warna nya dari RGB ke GRAY

```
harimau = cv2.imread("harimau_siberia.jpg")
harimau = cv2.cvtColor(harimau, cv2.COLOR_BGR2GRAY)
print(harimau.shape)

plt.imshow(harimau, cmap="gray")
#Miqdad Hilya Hasan_202331085
```

3. Kemudian, saya akan menyiapkan kernel konvolusi sebagai berikut:

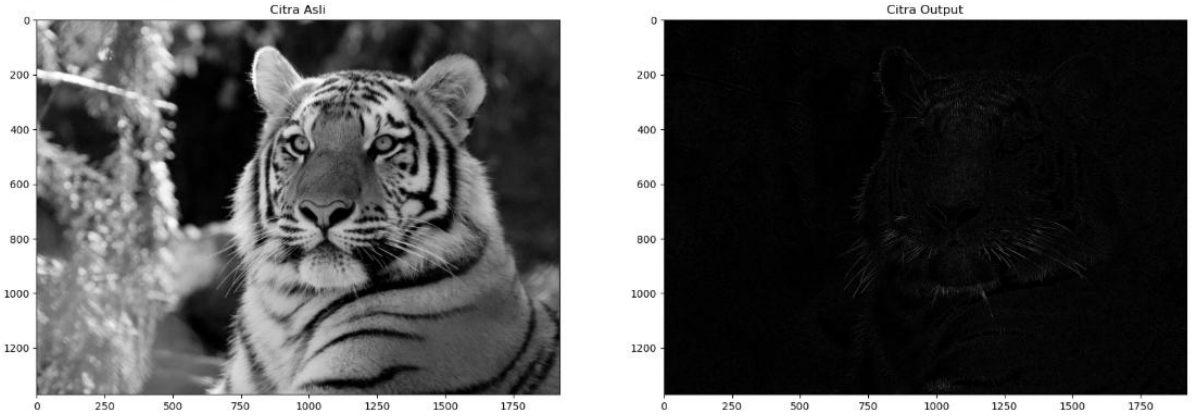
```
kernel = np.array([[-1, 0, -1],
                   [0, 4, 0],
                   [-1, 0, -1]])
#Miqdad Hilya Hasan_202331085
```

4. Langkah sleanjutnya, saya akan memproses citra dengan filter 2D dengan menggunakan function yang sudah di sediakan oleh library cv2, dan menampilkan citra sebelum dan setelah di olah

```
outputharimau = cv2.filter2D(harimau, -1, kernel)

fig, axs = plt.subplots(1,2, figsize=(20,20))
ax = axs.ravel()

ax[0].imshow(harimau, cmap="gray")
ax[0].set_title("Citra Asli")

ax[1].imshow(outputharimau, cmap="gray")
ax[1].set_title("Citra Output")

plt.show
#Miqdad Hilya Hasan_202331085
```

Output:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



5. Selkanjutnya, saya akan mengaplikasikan noise snp, gaussian, dan speckle pada citra menggunakan fungsi random_noise di module sikmage.util

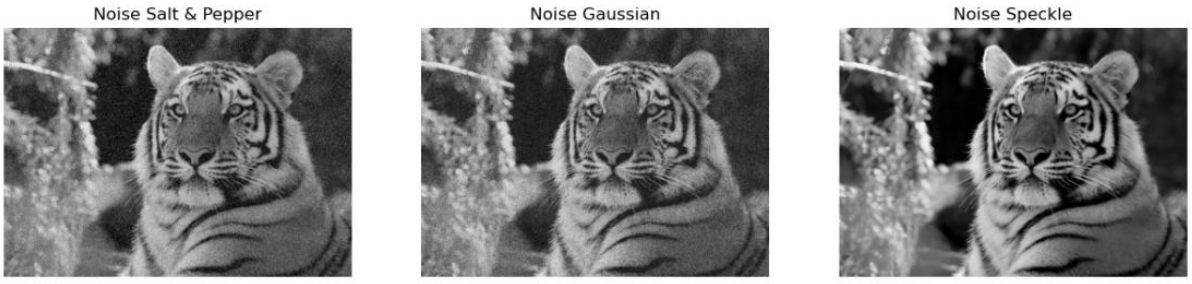
```
from skimage.util import random_noise
#Miqdad Hilya Hasan_202331085
```

6. Lalu, saya akan menerapkan ketiga noise tersebut pada citra. Salt and pepper noise di aplikasikan dengan menggunakan fungsi randoim noise mode s&p dengan jumlah yang memperngaruhi filter sebanyak 20%, gaussian noise dengan menggunakan fungsi yang sama namun menggunakan mode gaussian, dan measn bernilai 0 dan variance bernilai 0.1. lalunoise speckel dengan fungsi yang sama dan mode speckle, menggunakan parameter default, dan menampilkan citra yang suddah di berikan ketiga noise tersebut

```
noise_img_snp = random_noise(harimau, mode= "s&p", amount = 0.2)

#add gaussioan noise to the image
noise_img_gaussian = random_noise(harimau, mode= "gaussian", mean = 0, var = 0.1)

#add speckle noise tothe image
noise_img_speckle = random_noise(harimau, 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')

fig, axs = plt.subplots(1, 3, figsize=(15, 15))  # 3 baris x 3 kolom
ax = axs.ravel()  # ubah jadi array 1 dimensi untuk akses mudah

# Salt & Pepper
ax[0].imshow(noise_img_snp, cmap="gray")
ax[0].set_title("Noise Salt & Pepper")
ax[0].axis("off")
# Gaussian
ax[1].imshow(noise_img_gaussian, cmap="gray")
ax[1].set_title("Noise Gaussian")
ax[1].axis("off")
# Speckle
ax[2].imshow(noise_img_speckle, cmap="gray")
ax[2].set_title("Noise Speckle")
ax[2].axis("off")
#Miqdad Hilya Hasan_202331085
```

Output:

```
(-0.5, 1919.5, 1370.5, -0.5)
```



C. Jelaskan kembali materi deteksi tepi dengan gambar "parkir.jpg".

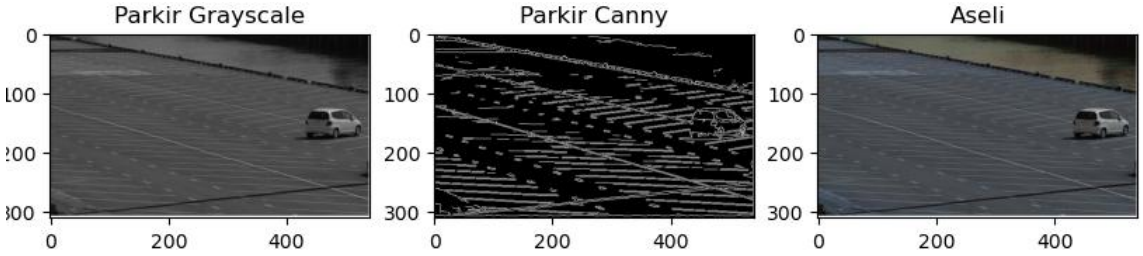
1. Pada praktikum tersebut, citra yang di gunakan di oolah dengan memproses citra tersebut menggunakna fungsi Canny di library Opencv atau cv2 yang di berikan alias cv pada kode tersebut. Fungsi canny tersebut menerapkan alforitma canny, pada fungsi tersebut, parameter yang di gunakan adalah (gray, 75, 150). parameter tersebut akan mementukan citra mana yang akan di olah, dan thereshold minimum dan maklsimum. Threshold yang ada di antara nilai maksimum dan minimum akan di terima dan di ubah nilia pixel nya menjadi maksimum (255) dan yang di tolak akan menjadi nilai minimum (0). Pada praktikum ini, nilai minimum adalah 75, dan maksimum nya adalah 150, dan citra yang di gunakan Bernama gray.

```
gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY)
edges = cv.Canny(gray, 75, 150)
```

2. Kemudian, akan di lakukan output berupa citra yang sudah menjadi grayscale, citra dengan channel warna asli, dan citra yang sudah di aplikasikan algoritma canny

```
fig, axs = plt.subplots(1,3, figsize = (10,10))
ax = axs.ravel()

axs[0].imshow(gray, cmap = "gray")
axs[0].set_title("Parkir Grayscale")

axs[1].imshow(edges, cmap = "gray")
axs[1].set_title("Parkir Canny")

axs[2].imshow(image, cmap = "gray")
axs[2].set_title("Aseli")
```

Ouptut:



D. Aplikasikan praktikum geometrik kemarin untuk mengambil plat "R 6792 XH" dari gambar plat.jpg

1. Langkah pertama yang saya lakukan adalah mengimpor semua library yang di butuhkan, lalu mengimport citra yang akan di gunakan, serta menggant cahnnel nya dari BGR ke RGB.

```
import numpy as np
import cv2 as cv
import matplotlib.pyplot as plt
from skimage import transform

plat = cv.imread('plat.jpg')
platR = cv.cvtColor(plat, cv.COLOR_BGR2RGB)
print(platR.shape)
plt.imshow(platR)
#Miqdad Hilya Hasan_202331085
```

Output:

```
(338, 600, 3)
<matplotlib.image.AxesImage at 0x17d6b05f200>
```

2. Kemudian, saya akan membuat array Bernama src yang membuat area berbentuk persegi pada koordinat yang nanti akan di tentukan pada citra, kemudoian, saya menggunakan webapp yang di sediakan oleh pemberi tugas untuk menentukan koordinat dari target yang akan di crop. Setelah koordinat di temukan, saya membuat array yang akan menyimpan koordinat tersebut, array tersebut Bernama crop. Lalu saya menggunakan fungsi projective transform dan melakukan estiomasi mapping dengan parameter berupa array src dan crop yang sudah saya buat sebelumnya, lalu saya akan menggunakan fungsi transform.warp untuk mengubah citra yang sudah di crop menjadi lurus atau rata (sebelumnya target tyang di crop tidak berbentuk persegi Panjang rata) dengan dimensi 50x300. Kemudian, saya melakukan output berupa cotra utuh dan citra yang sudah di crop

```
src = np.array([
    [0,0],
    [0,50],
    [300,50],
    [300,0]
])

crop = np.array([
    [374,191],
    [347,254],
    [520,268],
    [543,191]
])

tform = transform.ProjectiveTransform()
tform.estimate(src,crop)

warped = transform.warp(platR, tform, output_shape=(50,300))

fig, axs = plt.subplots(1,2, figsize = (15,15))
ax = axs.ravel()

axs[0].imshow(warped)
axs[0].set_title("Warped img 1")

axs[1].imshow(platR)
axs[1].plot(crop[:,0], crop[:,1], '.r')
axs[1].set_title("Gambar Asli")

for a in axs:
    a.axis('Off')

plt.tight_layout()
plt.show()
#Miqdad Hilya Hasan_202331085
```

Output: