

JOBSHEET 3
PENGANTAR CITRA DIGITAL DAN VISI KOMPUTER



Dibuat Oleh:
Leon Shan Yoedha Adjie

POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
2025/2026



NAMA

: Leon Shan Yoedha Adjie

NIM

: 2341720136

KELAS

: 3E

PRAKTIKUM

1. Operasi Citra Sederhana

a. Mengubah tingkat kecerahan.

```
import numpy as np
import cv2 as cv
from google.colab.patches import cv2_imshow

"Mengubah tingkat kecerahan citra"
"=====
try:
    brightness = int(input("Masukkan nilai kecerahan: "))
except ValueError:
    print('Error, not a number')

original = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/female.tiff')
brightness_image = np.zeros(original.shape, original.dtype)

#akses per piksel
for y in range(original.shape[0]):
    for x in range(original.shape[1]):
        for c in range(original.shape[2]):
            brightness_image[y,x,c] = np.clip(original[y,x,c] + brightness, 0 , 255)
#cara simpel loop
#brightness_image = cv.convertScaleAbs(original, beta=brightness)

final_frame = cv.hconcat((original, brightness_image))
cv2_imshow(final_frame)
```

Masukkan nilai kecerahan: 50
/tmp/ipython-input-122727855.py:19: RuntimeWarning: overflow encountered in scalar add
brightness_image[y,x,c] = np.clip(original[y,x,c] + brightness, 0 , 255)



NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

TUGAS PRAKTIKUM D1

1. Implementasikan inverse citra pada Google Colaboratory menggunakan formula yang terdapat pada bagian Ulasan Teori, sehingga menghasilkan keluaran seperti berikut:



Jawab:

```
gambar_awal = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/KTP7.jpeg')
gambar_awal = cv.resize(gambar_awal, (0,0), fx=0.5, fy=0.5)
gambar_inverse = 255 - gambar_awal

final_frame = cv.hconcat((gambar_awal, gambar_inverse))
cv2_imshow(final_frame)
```



2. Implementasikan transformasi contrast pada Google Colaboratory menggunakan formula yang terdapat pada bagian Ulasan Teori untuk kontras, sehingga menghasilkan keluaran seperti berikut:

Mengubah kontras dan tingkat kecerahan citra

Masukkan tingkat kecerahan [-255 - 255]: 50

Masukkan kontras [1.0 - 3.0]: 2





NAMA
NIM
KELAS

: Leon Shan Yoedha Adjie
: 2341720136
: 3E

Jawab:

```
import numpy as np
import cv2 as cv
from google.colab.patches import cv2_imshow

"Mengubah kontras dan tingkat kecerahan citra"
"=====
try:
    brightness = int(input("Masukkan tingkat kecerahan [-255 - 255]: "))
except ValueError:
    print('Error, not a number')
try:
    contrast = float(input("Masukkan kontras [1.0 - 3.0]: "))
except ValueError:
    print('Error, not a number')

original = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/female.tiff')
modified_image = np.zeros(original.shape, original.dtype)

#akses per piksel
for y in range(original.shape[0]):
    for x in range(original.shape[1]):
        for c in range(original.shape[2]):
            modified_image[y, x, c] = np.clip(contrast * original[y, x, c] + brightness, 0, 255)
#cara simpel loop

final_frame = cv.hconcat((original, modified_image))
cv2_imshow(final_frame)
```

Masukkan tingkat kecerahan [-255 - 255]: 50
Masukkan kontras [1.0 - 3.0]: 2.0



3. Implementasikan transformasi logarithmic brightness pada Google Colaboratory menggunakan formula yang terdapat pada bagian Ulasan Teori untuk transformasi log, sehingga menghasilkan keluaran seperti berikut:

Mengubah tingkat kecerahan citra dengan Transformasi Log

Masukkan nilai kecerahan: 50





NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

Jawab:

```
print("Mengubah tingkat kecerahan citra dengan Transformasi Log")
print("-----")
try:
    brightness = int(input("Masukkan tingkat kecerahan [-255 - 255]: "))
except ValueError:
    print('Error, not a number')
    brightness = 0

original = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/female.tiff')
if original is None:
    raise ValueError("Gambar tidak ditemukan. Periksa path gambar.")

# Konversi ke float dan tambahkan brightness
img_float = original.astype(np.float32) + brightness
img_float = np.clip(img_float, 0, 255)

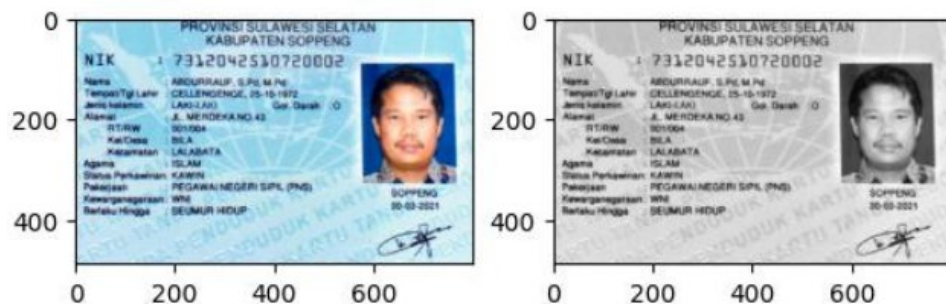
# Transformasi log
c = 255 / np.log(1 + np.max(img_float))
log_img = c * np.log(1 + img_float)
log_img = np.clip(log_img, 0, 255).astype(np.uint8)

# Gabungkan original dan hasil log brightness
final_frame = cv.hconcat((original, log_img))
cv2_imshow(final_frame)
```



4. Implementasikan transformasi grayscale menggunakan metode averaging, lightness, dan luminance pada Google Colaboratory menggunakan formula yang terdapat pada bagian Ulasan Teori, sehingga menghasilkan keluaran seperti berikut:

a. Averaging





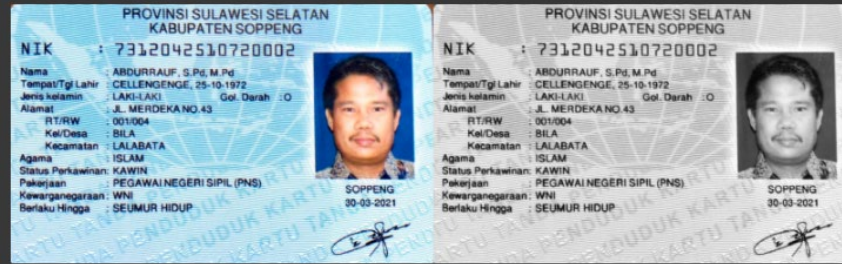
NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

Jawab:

```
gambar_awal = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/KTP7.jpeg')
gambar_awal = cv.resize(gambar_awal, (0,0), fx=0.5, fy=0.5)
# Rumus: Gray = (R + G + B) / 3
gambar_gray_avg = np.mean(gambar_awal, axis=2).astype(np.uint8)

# Ubah ke 3 channel agar bisa digabungkan dengan gambar_awal
gambar_gray_avg_bgr = cv.merge([gambar_gray_avg, gambar_gray_avg, gambar_gray_avg])

# Gabungkan gambar awal dan hasil grayscale averaging
final_frame = cv.hconcat((gambar_awal, gambar_gray_avg_bgr))
cv2_imshow(final_frame)
```



b. Lightness

```
<matplotlib.image.AxesImage at 0x781d2577f7f0>
```



Jawab:

```
gambar_awal = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/KTP7.jpeg')
gambar_awal = cv.resize(gambar_awal, (0,0), fx=0.5, fy=0.5)
# Rumus: Light = (Max Warna + Min Warna) / 2
max_channel = np.max(gambar_awal, axis=2)
min_channel = np.min(gambar_awal, axis=2)
gray_lightness = ((max_channel + min_channel) / 2).astype(np.uint8)
gray_lightness_bgr = cv.merge([gray_lightness, gray_lightness, gray_lightness])

# Gabungkan gambar awal dan hasil grayscale averaging
final_frame = cv.hconcat((gambar_awal, gray_lightness_bgr))
cv2_imshow(final_frame)
```

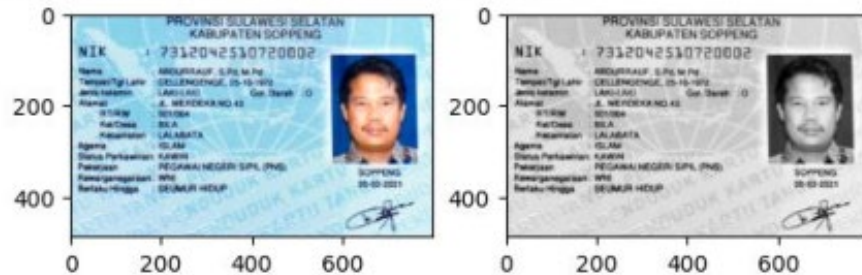




NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

c. Luminance

```
<matplotlib.image.AxesImage at 0x781d70a42fe0>
```



Jawab:

```
gambar_awal = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/KTP7.jpeg')
gambar_awal = cv.resize(gambar_awal, (0,0), fx=0.5, fy=0.5)
# Rumus: Light = (Max Warna + Min Warna) / 2
gray_luminance = (0.114 * gambar_awal[:, :, 0] + 0.587 * gambar_awal[:, :, 1] + 0.299 * gambar_awal[:, :, 2]).astype(np.uint8)
gray_luminance_bgr = cv.merge([gray_luminance, gray_luminance, gray_luminance])

# Gabungkan gambar awal dan hasil grayscale averaging
final_frame = cv.hconcat((gambar_awal, gray_luminance_bgr))
cv2.imshow('final_frame')
```





NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

5. Tampilkanlah warna tertentu pada citra, dan ubah warna lain menjadi grayscale. Misal, tampilkan warna biru pada citra masukan dan ubah bagian lain yang tidak berwarna biru menjadi grayscale seperti pada contoh berikut:



Jawab:

```
gambar_awal = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/KTP7.jpeg')
gambar_awal = cv.resize(gambar_awal, (0,0), fx=0.4, fy=0.4)
# Buat citra grayscale
img_gray = cv.cvtColor(gambar_awal, cv.COLOR_BGR2GRAY)
img_gray_bgr = cv.cvtColor(img_gray, cv.COLOR_GRAY2BGR)

# Range threshold warna biru pada BGR
lower_blue = np.array([100, 0, 0], dtype=np.uint8)
upper_blue = np.array([255, 120, 120], dtype=np.uint8)

# Mask untuk warna biru
mask_blue = cv.inRange(gambar_awal, lower_blue, upper_blue)

# Buat hasil: warna biru tetap, selain biru jadi grayscale
gambar_mencolok = gambar_awal.copy()
gambar_mencolok[mask_blue == 0] = img_gray_bgr[mask_blue == 0]

# Gabungkan dan tampilkan
final = cv.hconcat([gambar_awal, gambar_mencolok])
cv2_imshow(final)
```





NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

D2. Operasi Aritmatika dan Logika

1. Buat Gamma Correction sesuai dengan petunjuk berikut:

Percobaan ini akan meminta anda membuat Gamma Correction. Pada percobaan ini, nilai Gamma akan diset dengan meminta masukan dari pengguna. Berikut adalah kode untuk meminta masukan nilai dari pengguna. Lanjutkan kode tersebut dengan membuat image dengan gamma correction sesuai rumus yang telah diberikan.

Jawab:

```
img = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/female.tiff')
img = cv.resize(img, (0,0), fx=1, fy=1)

# Gamma Correction
# Rumus:  $I_{out} = 255 * (I_{in} / 255) ** gamma$ 
img_float = img.astype(np.float32) / 255.0
img_gamma = np.power(img_float, 1.0/gamma)
img_gamma = np.clip(img_gamma * 255, 0, 255).astype(np.uint8)

# Tampilkan hasil
final = cv.hconcat([img, img_gamma])
cv2_imshow(final)
```

Gamma Correction pada citra

Masukkan nilai Gamma: 3



2. Buat Simulasi Image Depth

Jawab:

```
bit_depth = 2
level = 255 / (pow(2, bit_depth) - 1)

# Baca citra grayscale
original = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/female.tiff', cv.IMREAD_GRAYSCALE)

# Proses kuantisasi
quant_img = np.round(original / level) * level
quant_img = np.clip(quant_img, 0, 255).astype(np.uint8)

# Gabungkan gambar asli dan hasil kuantisasi
gabungan = cv.hconcat([original, quant_img])
cv2_imshow(gabungan)
```





NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

3. Buat modul Average Denoising

Jawab:




```
import glob

def psnr(img1, img2):
    mse = np.mean((img1.astype(np.float32) - img2.astype(np.float32)) ** 2)
    if mse == 0:
        return float('inf')
    PIXEL_MAX = 255.0
    return 20 * np.log10(PIXEL_MAX / np.sqrt(mse))

# Baca citra asli (tanpa noise)
img_asli = cv.imread('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/galaxy.jpg')



# Baca semua gambar noise ke list
cv_img = []
for img in glob.glob('/content/drive/MyDrive/Pengolahan Citra dan Visi Komputer/Week 3/Gambar/noises/noises/*.jpg'):
    n = cv.imread(img)
    if n is None:
        print(f"File {img} gagal dibaca!")
    cv_img.append(n)
cv_img = np.array(cv_img) # shape: (jumlah_gambar, tinggi, lebar)

jumlah_list = [10, 20, 40, 80, 100]
for jumlah in jumlah_list:
    avg_img = np.mean(cv_img[:jumlah], axis=0)
    avg_img = np.clip(avg_img, 0, 255).astype(np.uint8)
    nilai_psnr = psnr(img_asli, avg_img)
    print(f"PSNR untuk average dari {jumlah} gambar noisy: {nilai_psnr:.2f} dB")
    gabungan = cv.hconcat([img_asli, avg_img])
    cv2_imshow(gabungan)
```

No.	Jumlah Citra di Average	Image Hasil	Nilai PSNR (db)
1.	10		19.73
2.	20		19.84
3.	40		19.89



NAMA : Leon Shan Yoedha Adjie
NIM : 2341720136
KELAS : 3E

4.	80		19.92
5.	100		19.92

Dari hasil PSNR yang anda catat pada tabel diatas, kesimpulan yang dapat diambil adalah:

Semakin banyak jumlah citra noisy yang di-average, nilai PSNR hasil average semakin tinggi. Nilai PSNR yang tinggi menunjukkan hasil denoising lebih mendekati citra asli (kualitas lebih baik). Average denoising sangat efektif untuk noise Gaussian jika tersedia banyak variasi gambar noisy dari citra yang sama.

Link Google Colab:

https://colab.research.google.com/github/Holycious/PCVK_GANJIL_2025/blob/main/Week2.ipynb

Link Github:

https://github.com/Holycious/PCVK_GANJIL_2025/blob/main/Week2.ipynb