

LAMPIRAN 3 CODING

A. Server Tree Counting

```
def cek(filename):
    image = cv2.imread(filename)
    hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
    threshold, binary_image = cv2.threshold(hsv_image[:, :,
2], 120, 210, cv2.THRESH_BINARY | cv2.THRESH_OTSU)
    image_denoised = filters.median(binary_image,
selem=np.ones((5, 5)))
    binary_image = image_denoised
    kernel = np.ones((4, 4), dtype='uint8')
    morph_closed_image = cv2.morphologyEx(binary_image,
cv2.MORPH_CLOSE, kernel)
    dist_transform_image =
cv2.distanceTransform(morph_closed_image, cv2.DIST_L2,
cv2.DIST_MASK_PRECISE)
    border_size = 25
    dist_border = cv2.copyMakeBorder(dist_transform_image,
border_size, border_size, border_size, border_size,
cv2.BORDER_CONSTANT |
cv2.BORDER_ISOLATED, 0)
    gap = 8

    template_file = cv2.imread('template.jpg')
    template_file = cv2.resize(template_file, (64, 64))
    template_file = cv2.cvtColor(template_file,
cv2.COLOR_BGR2HSV)

    _, kernel2 = cv2.threshold(template_file[:, :, 2], 120,
210, cv2.THRESH_BINARY | cv2.THRESH_OTSU)

    # median filtering the teampalte also...!!

    image_denoised = filters.median(kernel2,
selem=np.ones((5, 5)))
    kernel2 = image_denoised
    for i in range(len(kernel2)):
        for j in range(len(kernel2[i])):
```

```

        if kernel2[i][j] != 0:
            kernel2[i][j] = 1

    kernel2 = cv2.copyMakeBorder(kernel2, gap, gap, gap, gap,
cv2.BORDER_CONSTANT | cv2.BORDER_ISOLATED, 0)
    dist_trans_template = cv2.distanceTransform(kernel2,
cv2.DIST_L2, cv2.DIST_MASK_PRECISE)
    template_matched = cv2.matchTemplate(dist_border,
dist_trans_template, cv2.TM_CCOEFF_NORMED)

    # Threshold the template matched image
    mn, mx, _, _ = cv2.minMaxLoc(template_matched)
    th, peaks = cv2.threshold(template_matched, 0.10, 0.60,
cv2.THRESH_BINARY)

    # Convert the peak values to 8-bit unsigned integers
    peaks8u = cv2.convertScaleAbs(peaks)

    # Find the contours in the peaks
    contours, _ = cv2.findContours(peaks8u, cv2.RETR_CCOMP,
cv2.CHAIN_APPROX_SIMPLE)

    copy = cv2.imread(filename)

    count = 0

    for i in range(len(contours)):
        if cv2.contourArea(contours[i]) < 150:
            continue

    # Hitung centroid dari kontur
    M = cv2.moments(contours[i])
    if M["m00"] != 0:
        cX = int(M["m10"] / M["m00"])
        cY = int(M["m01"] / M["m00"])

    # Gambar satu titik di centroid
    cv2.circle(copy, (cX, cY), 3, (0, 255, 0), -1)

```

```
# Tambahkan label angka pada titik
cv2.putText(copy, str(count), (cX - 10, cY -
10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 2,
cv2.LINE_AA)

count += 1

plt.axis('off')
plt.imshow(copy)
plt.savefig(filename.replace(".jpg", "_HASIL.jpg"),
bbox_inches='tight', dpi=1000)
```



B. Server Ndzi

```
pip install rasterio
import rasterio
import numpy as np
import matplotlib.pyplot as plt

file_path = 'apayak.tiff'

with rasterio.open(file_path) as src:
    citra = src.read()
    citra = np.moveaxis(citra, 0, -1)

    band_merah = citra[:, :, 2].astype(float)
    band_nir = citra[:, :, 3].astype(float)

    # Hitung NDVI
    ndvi = (band_nir - band_merah) / (band_nir + band_merah)

    # Visualisasi NDVI
    plt.imshow(ndvi, cmap='RdYlGn')
    plt.colorbar()
    plt.title('NDVI')
    plt.xlabel('Kolom')
    plt.ylabel('Baris')
    plt.show()

    # Klasifikasi NDVI berdasarkan kerapatan vegetasi
    ndvi_rendah = ndvi < 0.2 # Non-vegetasi atau vegetasi
    rendah
    ndvi_sedang = (ndvi >= 0.2) & (ndvi < 0.5) # Vegetasi
    sedang
    ndvi_tinggi = ndvi >= 0.5 # Vegetasi tinggi

    # Hitung proporsi area untuk masing-masing kategori
    proporsi_rendah = np.sum(ndvi_rendah) / ndvi.size
    proporsi_sedang = np.sum(ndvi_sedang) / ndvi.size
    proporsi_tinggi = np.sum(ndvi_tinggi) / ndvi.size
    print(f"Proporsi area NDVI rendah (non-vegetasi atau
    rendah/merah): {proporsi_rendah:.2%}")
```

```
print(f"Proporsi area NDVI sedang (vegetasi sedang/kuning):  
{proporsi_sedang:.2%}")  
print(f"Proporsi area NDVI tinggi (vegetasi tinggi/hijau):  
{proporsi_tinggi:.2%}")
```



LAMPIRAN 4 FOTO KEGIATAN



