

Nama : Muhammad Sidqi Nabhan

NIM : 1103200179

Kelas : TK 45 G09

## Tugas 1

```
# Import pustaka yang diperlukan
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Upload gambar
from google.colab import files
uploaded = files.upload()

# Membaca gambar
filename = list(uploaded.keys())[0]
image = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)

# Tampilkan gambar asli
plt.figure(figsize=(10, 5))
plt.title("Original Image")
plt.imshow(image, cmap='gray')
plt.axis('off')
plt.show()

# 1. Membuat dan Mengaplikasikan Filter Moving Average
def moving_average_filter(image, kernel_size=5):
    kernel = np.ones((kernel_size, kernel_size), np.float32) / (kernel_size**2)
    filtered_image = cv2.filter2D(image, -1, kernel)
    return filtered_image

filtered_image = moving_average_filter(image)
plt.figure(figsize=(10, 5))
plt.title("Moving Average Filter")
plt.imshow(filtered_image, cmap='gray')
plt.axis('off')
plt.show()

# 2. Deteksi Fitur dengan SIFT
sift = cv2.SIFT_create()
keypoints, descriptors = sift.detectAndCompute(image, None)
image_sift = cv2.drawKeypoints(image, keypoints, None,
flags=cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)
```

```

plt.figure(figsize=(10, 5))
plt.title("SIFT Feature Detection")
plt.imshow(image_sift, cmap='gray')
plt.axis('off')
plt.show()

# 3. Representasi Histogram Gambar
def plot_histogram(image):
    plt.figure(figsize=(10, 5))
    plt.title("Image Histogram")
    plt.hist(image.ravel(), bins=256, range=(0, 256), color='blue', alpha=0.7)
    plt.xlabel('Pixel Intensity')
    plt.ylabel('Frequency')
    plt.show()

plot_histogram(image)

# 4. Gaussian Smoothing
def gaussian_smoothing(image, kernel_size=5):
    blurred_image = cv2.GaussianBlur(image, (kernel_size, kernel_size), 0)
    return blurred_image

gaussian_image = gaussian_smoothing(image)
plt.figure(figsize=(10, 5))
plt.title("Gaussian Smoothing")
plt.imshow(gaussian_image, cmap='gray')
plt.axis('off')
plt.show()

# 5. Deteksi Tepi dengan Sobel Filter
def sobel_edge_detection(image):
    grad_x = cv2.Sobel(image, cv2.CV_64F, 1, 0, ksize=3)
    grad_y = cv2.Sobel(image, cv2.CV_64F, 0, 1, ksize=3)
    gradient_magnitude = cv2.magnitude(grad_x, grad_y)
    return np.uint8(gradient_magnitude)

sobel_image = sobel_edge_detection(image)
plt.figure(figsize=(10, 5))
plt.title("Sobel Edge Detection")
plt.imshow(sobel_image, cmap='gray')
plt.axis('off')
plt.show()

# 6. Representasi Fitur dengan Histogram of Oriented Gradients (HOG)
from skimage.feature import hog

def compute_hog(image):
    if len(image.shape) == 3: # Jika gambar berwarna (RGB)
        image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY) # Konversi ke grayscale

```

```
features, hog_image = hog(image, orientations=9, pixels_per_cell=(8, 8),
                           cells_per_block=(2, 2), visualize=True, channel_axis=None)
return hog_image
```

```
hog_image = compute_hog(image)
plt.figure(figsize=(10, 5))
plt.title("Histogram of Oriented Gradients (HOG)")
plt.imshow(hog_image, cmap='gray')
plt.axis('off')
plt.show()
```

Hasil Output :

1. Filter Moving Average

Moving Average Filter



Analisa :

- Filter ini merupakan operasi perataan (smoothing) dengan menggunakan kernel rata-rata.
- Efek ini muncul karena rata-rata intensitas piksel dalam jendela kernel menyebabkan detail tajam (seperti tepi dan teks kecil) menjadi lebih halus
- Gambar hasilnya terlihat lebih kabur dibandingkan gambar aslinya
- Bagian tepi logo "Telkom University" dan huruf menjadi kabur karena perataan intensitas di sekitar piksel tepi
- Area dengan warna seragam (seperti latar belakang putih atau blok abu-abu) tetap hampir sama, tetapi sedikit diratakan untuk menghilangkan noise kecil.

## 2. Deteksi fitur dengan SIFT

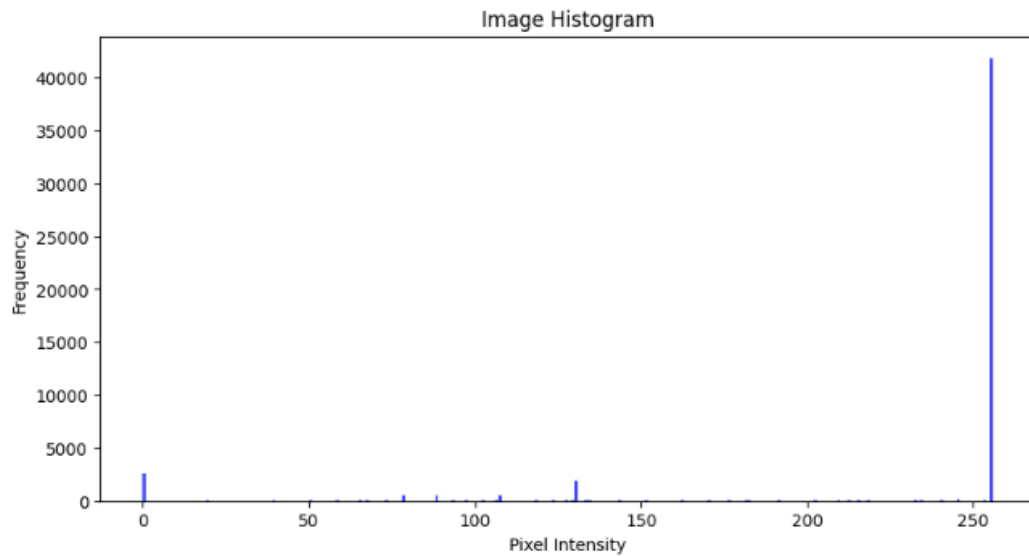
### SIFT Feature Detection



Analisa :

- Gambar menunjukkan banyak lingkaran dengan orientasi di seluruh gambar
- Gambar keluaran menunjukkan titik-titik kunci (keypoints) pada area yang dianggap penting
- Lingkaran mewakili lokasi fitur yang dianggap signifikan oleh SIFT, seperti sudut, tepi, atau pola tekstur lokal.
- Sebagian besar keypoints terdeteksi pada teks "*Telkom University*" dan bagian logo karena area ini memiliki banyak kontras lokal dan pola sudut yang kuat
- Area putih atau abu-abu polos (latar belakang) memiliki sangat sedikit atau tidak ada keypoints karena kurangnya perubahan intensitas lokal

### 3. Representasi Histogram Gambar



Analisa :

- Sebagian besar piksel memiliki nilai intensitas maksimum (255), menunjukkan gambar memiliki banyak area putih atau terang.
- Ada sedikit piksel dengan intensitas 0, menunjukkan adanya beberapa area hitam pada gambar.
- Hampir tidak ada piksel pada intensitas menengah (50–200). Ini menunjukkan gambar memiliki kontras yang sangat tinggi atau merupakan gambar biner (hanya hitam dan putih dengan sedikit gradasi)

#### 4. Gaussian Smoothing

##### Gaussian Smoothing



Analisa :

- Objek-objek dengan perbedaan intensitas yang tinggi, seperti tepi pada teks dan logo, menjadi lebih lembut atau kabur ini terlihat pada tulisan "Telkom University" yang kehilangan ketajaman pada hurufnya
- Gaussian smoothing menggunakan kernel berbentuk Gaussian (berbentuk lonceng), yang memberikan bobot lebih tinggi pada pixel di tengah kernel dan bobot yang lebih rendah pada pixel di sekitarnya.
- Kernel dalam kode ini memiliki ukuran 5x5, yang cukup besar untuk menghasilkan efek kabur yang signifikan

## 5. Sobel Filter

Sobel Edge Detection



Analisa :

- Sobel filter menghitung perubahan intensitas pada arah horizontal (x) dan vertikal (y), sehingga sensitif terhadap tepi di kedua arah.
- Penggabungan gradien dari kedua arah menghasilkan magnitudo gradien yang lebih menonjolkan tepi-tepi gambar
- Jika gambar asli memiliki noise, Sobel filter bisa menangkap perubahan intensitas kecil yang seharusnya diabaikan. Namun, karena Gaussian smoothing telah dilakukan sebelumnya, efek noise cukup diminimalkan

## 6. Representasi Fitur dengan HOG

Histogram of Oriented Gradients (HOG)



Analisa :

- Informasi tekstur halus dihilangkan, dan hanya pola utama berdasarkan gradien arah yang ditampilkan. Hal ini disebabkan oleh proses pembagian gambar menjadi blok kecil (sel HOG), di mana gradien dihitung dan dikelompokkan ke dalam orientasi tertentu.
- Teknik ini mengabstraksi informasi yang relevan untuk analisis fitur dengan mempertahankan pola gradien yang signifikan.
- HOG dapat menangkap bentuk dan orientasi tanpa terpengaruh oleh intensitas absolut piksel, sehingga sangat berguna untuk tugas seperti deteksi objek atau pengenalan gambar.