

LAPORAN TUGAS PENGOLAHAN CITRA VIDEO



Disusun Oleh :
Moh. Iqbal Fatchurozi
5024201009

Dosen Pengampu :
Dr. Eko Mulyanto Yuniarno, S.T.,M.T.

**DEPARTEMEN TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2022**

1 Deskripsi Tugas

Membuat program yang dapat mengganti background pada video.

2 Metode 1

Dalam metode pertama ini, dengan cara mendeteksi warna latar belakang secara solid dengan menentukan lower hsv dan upper hsv warna yang akan ditimpa dengan gambar yang menjadi background.

2.1 Penjelasan

Library yang digunakan adalah opencv dan numpy.

```
[1]: import cv2 as cv
import numpy as np
```

Modul atau library cv2 atau dapat disebut OpenCV (Open Source Computer Vision Library) adalah perpustakaan fungsi pemrograman yang terutama ditujukan untuk visi komputer waktu nyata.

Modul atau library **numpy** adalah modul yang menyediakan fungsi-fungsi matematika yang bergantung pada array multidimensi. NumPy merupakan singkatan dari Numerical Python. NumPy merupakan salah satu library Python yang berfungsi untuk proses komputasi numerik. NumPy memiliki kemampuan untuk membuat objek N-dimensi array. Array merupakan sekumpulan variabel yang memiliki tipe data yang sama. Kelebihan dari NumPy Array adalah dapat memudahkan operasi komputasi pada data.

```
[2]: # Membuat objek video capture untuk membaca video dari kamera webcam.
cap = cv.VideoCapture(0)

while True:
    # Mendapatkan frame setiap detik
    ret, frame = cap.read()
    # Mengubah warna frame menjadi hsv
    hsv = cv.cvtColor(frame, cv.COLOR_BGR2HSV)
    f2 = cv.imread('img/1.jpg')
    # Mendapatkan ukuran dari frame
    b, c, d = frame.shape
    # Mengubah ukuran gambar latar belakang menjadi ukuran frame
    f2 = cv.resize(f2, (c, b), interpolation=cv.INTER_LINEAR)
    # Menentukan lower dan upper bound warna (disini menggunakan warna
    ↪ merah)
    lower_bound = np.array([136, 87, 111], dtype=np.uint8)
    upper_bound = np.array([180, 255, 255], dtype=np.uint8)
```

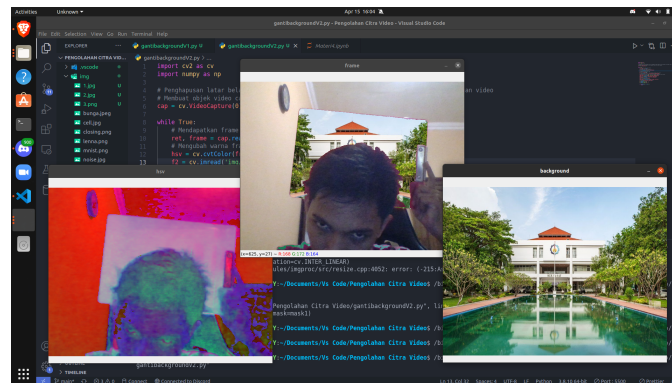
```

mask0 = cv.inRange(hsv, lower_bound, upper_bound)
output_img1 = cv.bitwise_and(f2, f2, mask=mask0)
# Membalik hasil mask
mask1 = cv.bitwise_not(mask0)
output_img2 = cv.bitwise_and(frame, frame, mask=mask1)
# Menggabungkan hasil dari output 1 dan 2
fg = cv.add(output_img1, output_img2)
cv.imshow('hsv', hsv)
cv.imshow('background', f2)
cv.imshow('frame', fg)

if cv.waitKey(1) == ord('q'):
    break

```

2.2 Hasil



3 Metode 2

Dalam metode kedua ini, dengan cara memakai mediapipe untuk mencari selfie segmentation. Setelah didapatkan maka diluar selfie segmentation akan ditimpa dengan gambar latar belakang.

3.1 Penjelasan

Library yang digunakan adalah os, opencv, numpy, dan mediapipe .

```

[4]: import cv2 as cv
import numpy as np
import mediapipe as mp

```

Modul atau library cv2 atau dapat disebut OpenCV (Open Source Computer Vision Library) adalah perpustakaan fungsi pemrograman yang terutama ditujukan untuk visi komputer waktu nyata.

Modul atau library **numpy** adalah modul yang menyediakan fungsi-fungsi matematika yang bergantung pada array multidimensi. NumPy merupakan singkatan dari Numerical Python. NumPy merupakan salah satu library Python yang berfungsi untuk proses komputasi numerik. NumPy memiliki kemampuan untuk membuat objek N-dimensi array. Array merupakan sekumpulan variabel yang memiliki tipe data yang sama. Kelebihan dari NumPy Array adalah dapat memudahkan operasi komputasi pada data.

Modul atau library **mediapipe** adalah modul yang menyediakan fungsi-fungsi yang bergantung pada pipeline yang dibuat oleh Google. MediaPipe menawarkan solusi ML lintas platform yang dapat disesuaikan untuk media live dan streaming.

Menyimpan gambar latar belakang.

```
[5]: bg_image = cv.imread("img/1.jpg")
```

selfie segmentation adalah menyegmentasikan wajah yang ada di dalam frame. Atau dapat disebut dengan menyeleksi orang yang ada di dalam frame.

```
[6]: mp_selfie_segmentation = mp.solutions.selfie_segmentation
selfie_segmentation = mp_selfie_segmentation.SelfieSegmentation(
    model_selection=1)
```

```
[7]: # Membuat objek video capture untuk membaca video dari kamera webcam.
cap = cv.VideoCapture(0)
while cap.isOpened():
    _, frame = cap.read()
    # Membalik frame ke arah horizontal
    frame = cv.flip(frame, 1)
    height, width, channel = frame.shape
    # Membuat objek frame yang akan dikirim ke mediapipe, sebelum itu
    →dirubah dulu warna menjadi RGB.
    RGB = cv.cvtColor(frame, cv.COLOR_BGR2RGB)
    # Mendapatkan hasil
    results = selfie_segmentation.process(RGB)
    # extract segmented mask
    mask = results.segmentation_mask
    # Mengeluarkan output dari hasil segmentasi mask
    cv.imshow("mask", mask)
    key = cv.waitKey(1)
    if key == ord('q'):
        break
    condition = np.stack(
        (results.segmentation_mask,) * 3, axis=-1) > 0.5
    # Mengubah ukuran gambar latar belakang ke ukuran yang sama dengan
    →frame asli
    bg_image = cv.resize(bg_image, (width, height))
```

```

# menyatukan frame dengan background image (latar belakang)
output_image = np.where(condition, frame, bg_image)
cv.imshow("Output", output_image)
cv.imshow("Frame", frame)
if key == ord('q'):
    break
    # jika 'c' ditekan, maka gambar latar belakang akan diganti
    → dengan gambar yang baru
elif key == ord('c'):
    bg_image = cv.imread('img/quito.jpg')

```

INFO: Created TensorFlow Lite XNNPACK delegate for CPU.

3.2 Hasil

