

LAPORAN MINI PROJECT COMPUTER VISION REAL-TIME OBJECT DETECTION DOCTOR STRANGE

Disusun Untuk Memenuhi Tugas Mini Project

Pelajaran : Computer Vision

Coach Pengajar : Coach Eko P Wahyono



Disusun Oleh :

1. Indra Saepudin
2. Ramanda Kholisandra
3. Izati Nuramadanti
4. Bintari
5. Inayah Khasnaputri A

**ARTIFICIAL INTELLIGENCE FOR GENZ JOBSEEKERS
SYNTHESIS ACADEMY
MAGANG DAN STUDI INDEPENDEN BERSERTIFIKAT
(MSIB)KAMPUS MERDEKA BATCH 3
2022**

➤ **Computer Vision Real-Time Object Detection Doctor Strange**

1. Awal mula penamaan ini terinspirasi dari sebuah film doctor strange dikarenakan gestur yang digunakan pada presentasi menyerupai adegan pada doctor strange saat mengeluarkan mantra.
2. Kita seperti layaknya tokoh Doctor Strange yang sedang mengeluarkan spell atau mantra. Selain bersifat unik, hal ini dapat membantu seseorang untuk mengontrol slide PPT agar bisa sekaligus menjelaskan materi.

➤ **Pembuatan Project Menggunakan Python**

Project ini dibangun dengan bahasa pemrograman Python dan teks editor Visual Studio Code. Langkah awal dalam pembuatannya adalah mengimpor beberapa library yang dibutuhkan. Library HandDetector digunakan untuk mentracking pergerakan tangan kita, library cv2 berfungsi untuk mengelola dan mengkalkulasi pergerakan yang terekam dari gestur tangan kita, library os digunakan untuk membuat program ini bisa menggunakan beberapa fungsi yang tertanam pada Sistem Operasi milik kita, dan library numpy digunakan untuk perhitungan matematika yang terkandung dalam program.

```
from cvzone.HandTrackingModule import HandDetector
import cv2
import os
import numpy as np
```

➤ **Men Setting lebar kamera yang akan merekam pergerakan gestur kita**

Variabel width dan height berfungsi untuk mengatur lebar jendela layar kamera yang akan ditampilkan ke layar monitor kita. gestureThreshold adalah variabel yang nantinya berfungsi sebagai penanda area yang terdaftar sebagai pembaca gestur tangan. folderPath berfungsi sebagai lokasi file presentasi kita. Di dalam camera Setup, kita implementasikan variabel width dan height ke dalam fungsi cap set.

```
# Parameters
width, height = 1280, 720
gestureThreshold = 300
folderPath = "Presentation"

# Camera Setup
cap = cv2.VideoCapture(2)
cap.set(3, width)
cap.set(4, height)
```

➤ **Pembuatan beberapa variabel**

Beberapa variabel ini berfungsi sebagai penampung value-value yang nantinya akan digunakan dalam pengolahan data atau processing. Seperti `imgList` yang berfungsi sebagai penampung banyaknya gambar slide, `delay` untuk mengatur kecepatan pembacaan gestur, `drawMode` untuk mengatur mode dimana kita mau menggambar pada slide tertentu, dan lain-lain.

```
# Variables
imgList = []
delay = 30
buttonPressed = False
counter = 0
drawMode = False
imgNumber = 0
delayCounter = 0
annotations = [[]]
annotationNumber = -1
annotationStart = False
hs, ws = int(120 * 1), int(213 * 1) # width and height of small image
```

➤ **Mengambil jumlah gambar untuk ditayangkan pada slide presentasi**

Pada kode ini, kita akan mengambil gambar yang sudah disediakan untuk bahan presentasi. Gambar yang sudah dilabeli dengan urutan angka, selanjutnya akan dihitung berapa jumlahnya dan juga akan dilakukan penyortiran berdasarkan angka yang paling kecil untuk awalan slide pertamanya.

```
# Get list of presentation images
pathImages = sorted(os.listdir(folderPath), key=len)
print(pathImages)
```

➤ **Mengambil dan menampilkan isi slide ke layar monitor**

Setelah beberapa slide sudah kita kumpulkan dalam 1 variabel, selanjutnya kita akan mencoba menampilkannya ke depan layar monitor. Pada variabel *success* terdapat fungsi untuk menampilkan hasil video realtime ke depan layar monitor. Selanjutnya, pada variabel *imgCurrent* berisi *path* gambar slide presentasi dan setiap urutan dari slide-slide tersebut. Selanjutnya di variabel *hands* berfungsi untuk mendeteksi gestur tangan yang tertangkap kamera.

```
while True:
    # Get image frame
    success, img = cap.read()
    img = cv2.flip(img, 1)
    pathFullImage = os.path.join(folderPath, pathImages[imgNumber])
    imgCurrent = cv2.imread(pathFullImage)

    # Find the hand and its landmarks
    hands, img = detectorHand.findHands(img) # with draw
    # Draw Gesture Threshold line
    cv2.line(img, (0, gestureThreshold), (width, gestureThreshold), (0, 255, 0), 10)
```

➤ **Membuat pengkondisian saat gestur tangan sudah terbaca**

Saat tangan sudah dapat ter-*tracking* oleh system, maka selanjutnya kita akan membuat beberapa aksi yang bisa digunakan tangan kita seperti membuat gestur untuk menggambar pada slide powerpoint-nya. Kita tetapkan dahulu lebar tangkapan layar yang bisa digunakan saat kita menggunakan gestur menggambar.

```

if hands and buttonPressed is False: # If hand is detected

    hand = hands[0]
    cx, cy = hand["center"]
    lmList = hand["lmList"] # List of 21 Landmark points
    fingers = detectorHand.fingersUp(hand) # List of which fingers are up

    # Constrain values for easier drawing
    xVal = int(np.interp(lmList[8][0], [width // 2, width], [0, width]))
    yVal = int(np.interp(lmList[8][1], [150, height-150], [0, height]))
    indexFinger = xVal, yVal

```

➤ Menggunakan gestur jari untuk menggeser slide

Pada bagian ini, kita akan menerapkan fungsi yang apabila ibu jari kita diacungkan, maka slide presentasi akan beralih ke slide selanjutnya, sedangkan saat jari kelingking kita diacungkan, maka slide akan berganti ke slide sebelumnya. Nilai matriks pada variabel `fingers` merepresentasikan posisi jari-jari kita. Jika nilainya 1, maka artinya posisi jarinya terangkat, namun jika nilainya 0 maka berarti posisi jarinya masih menggenggam.

```

if cy <= gestureThreshold: # If hand is at the height of the face
    if fingers == [1, 0, 0, 0, 0]:
        print("Left")
        buttonPressed = True
        if imgNumber > 0:
            imgNumber -= 1
            annotations = [[]]
            annotationNumber = -1
            annotationStart = False
    if fingers == [0, 0, 0, 0, 1]:
        print("Right")
        buttonPressed = True
        if imgNumber < len(pathImages) - 1:
            imgNumber += 1
            annotations = [[]]
            annotationNumber = -1
            annotationStart = False

```

➤ **Menambahkan aksi menggambar dan menggeser pointer**

Saat posisi 2 jari telunjuk dan tengah terangkat, maka kita bisa menggerakkan posisi pointer pada slide powerpoint. Namun jika kita hanya mengacungkan jari telunjuk saja, maka kita bisa melakukan aksi seperti menggambar pada slide-nya. Seperti sebelumnya, kita menginisialisasi posisi jari dengan menggunakan nilai matriks dalam sebuah list.

```
if fingers == [0, 1, 1, 0, 0]:
    cv2.circle(imgCurrent, indexFinger, 12, (0, 0, 255), cv2.FILLED)

if fingers == [0, 1, 0, 0, 0]:
    if annotationStart is False:
        annotationStart = True
        annotationNumber += 1
        annotations.append([])
    print(annotationNumber)
    annotations[annotationNumber].append(indexFinger)
    cv2.circle(imgCurrent, indexFinger, 12, (0, 0, 255), cv2.FILLED)

else:
    annotationStart = False
```

➤ **Membuat aksi menghapus coretan yang kita buat**

Nah jika tadi kita membuat perintah agar bisa menggambar pada slide presentasi, kita juga akan membuat perintah agar bisa menghapus coretan yang sudah kita buat. Dengan menggunakan gestur mengangkat 3 jari yaitu jari telunjuk, tengah, dan manis secara bersamaan secara sejajar dengan kepala kita.

```
else:
    annotationStart = False

if fingers == [0, 1, 1, 1, 0]:
    if annotations:
        annotations.pop(-1)
        annotationNumber -= 1
        buttonPressed = True
```

➤ **Mengatur agar coretan yang kita buat tidak saling menyatu garisnya**

Saat melakukan percobaan membuat coretan pada slide power pointnya, kita mengalami sebuah kendala yang dimana saat kita menggambar 2 buah garis yang memiliki posisi yang berbeda, maka akan ada garis tambahan yang menghubungkan ke-2 garis tersebut. Maka kita buat sebuah fungsi dimana setiap kali kita menggambar, maka titik koordinat dari notasi satu ke notasi lain akan dikumpulkan dan dipisahkan dalam list bersarang.

```
for i, annotation in enumerate(annotations):
    for j in range(len(annotation)):
        if j != 0:
            cv2.line(imgCurrent, annotation[j - 1], annotation[j], (0, 0, 200), 12)
```

➤ **Menggabungkan antara slide dengan kamera**

Untuk memudahkan kita dalam mengontrol slide dan gesture kita yang terekam dalam capture video secara real time, kita akan membuat jendela kamera yang terletak di pojok kanan atas jendela slide. Sehingga kita bisa secara leluasa melihat slide dan melihat apakah gestur tangan kita terbaca pada kameranya atau tidak.

```
imgSmall = cv2.resize(img, (ws, hs))
h, w, _ = imgCurrent.shape
imgCurrent[0:hs, w - ws: w] = imgSmall
```

➤ **Menampilkan jendela slide dan kamera ke layar monitor saat program dijalankan**

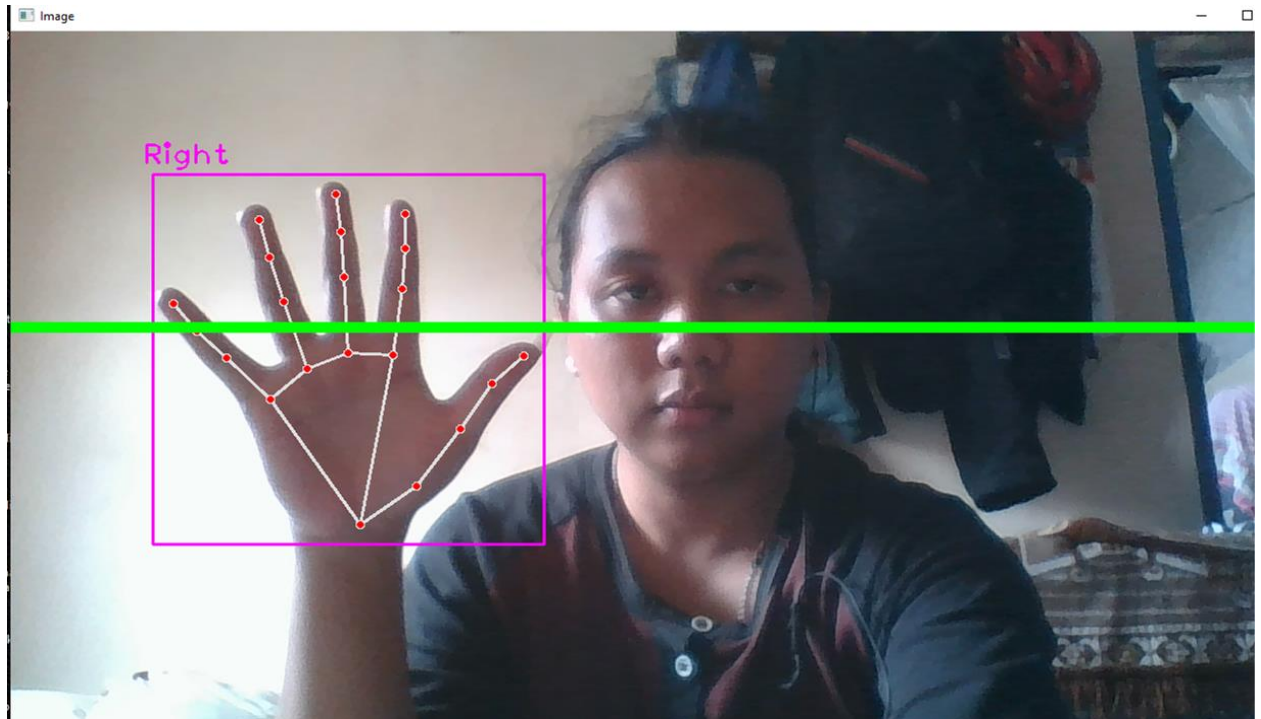
Jika dirasa settingan kita sudah diatur semua, maka tinggal tampilkan slide dan jendela tangkapan layar ke monitor untuk melakukan presentasi. Akan terdapat 2 jendela yang dimana jendela pertama untuk tangkapan layar untuk kita, dan jendela kedua untuk tampilan slide beserta layar kecil capture video seperti jendela pertama namun lebih kecil ukurannya yang terletak di pojok kanan atas slide. Saat presentasi selesai, kita bisa mengetikkan huruf “q” untuk menstop program aplikasinya

```
cv2.imshow("Slides", imgCurrent)
cv2.imshow("Image", img)

key = cv2.waitKey(1)
if key == ord('q'):
    break
```

➤ Hasil Program

Tampilan Jendela pertama untuk kamera nya



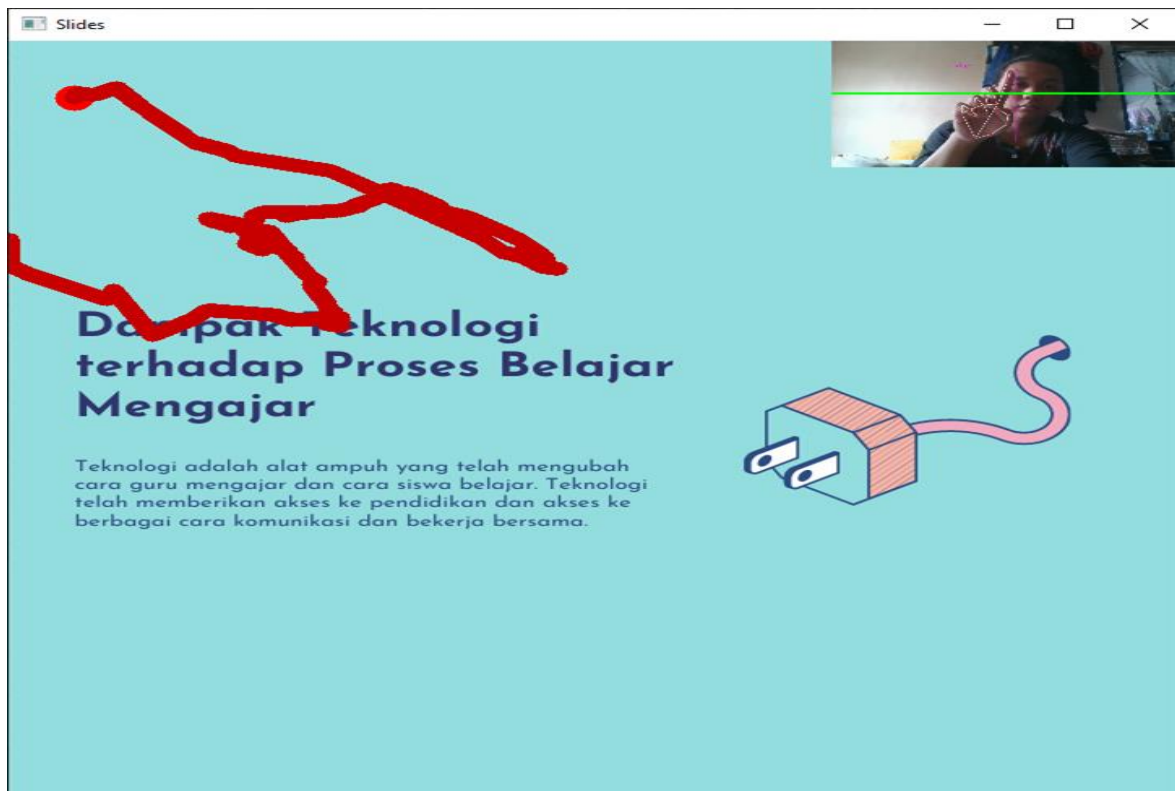
➤ Tampilan jendela slide



- Gesture mengangkat 2 jari untuk menggerakan pointer nya



- Menggunakan jari telunjuk untuk menggambar pada halaman slide



- Menghapus coretan dengan gestur mengangkat 3 jari secara bersamaan dan sejajar



- Menampilkan slide selanjutnya dengan mengangkat jari kelingking



- Mengembalikan ke slide sebelumnya dengan mengangkat ibu jari



➤ **Kesimpulan**

Dengan dibuatnya project ini, diharapkan bisa membantu bagi orang yang sedang melakukan presentasi di depan publik agar saat menyampaikan materi, tidak perlu harus mengklik pada laptop untuk mengganti slide. Cukup dengan gestur jari maka kita bisa mengganti slide dan menggambar sebuah pola pada layar. Ini membuat presentasi menjadi lebih fleksibel karena kita bisa fokus ke materi dan para audien tanpa repot-repot bolak-balik ke laptop untuk mengganti slide.

