

# FROM PIXELS TO COLORS:

**Real-Time Color  
Detection in Images  
with OpenCV**

ATHAYA ZAHRANI I

PORTOFOLIO PROJECT MODULE 3

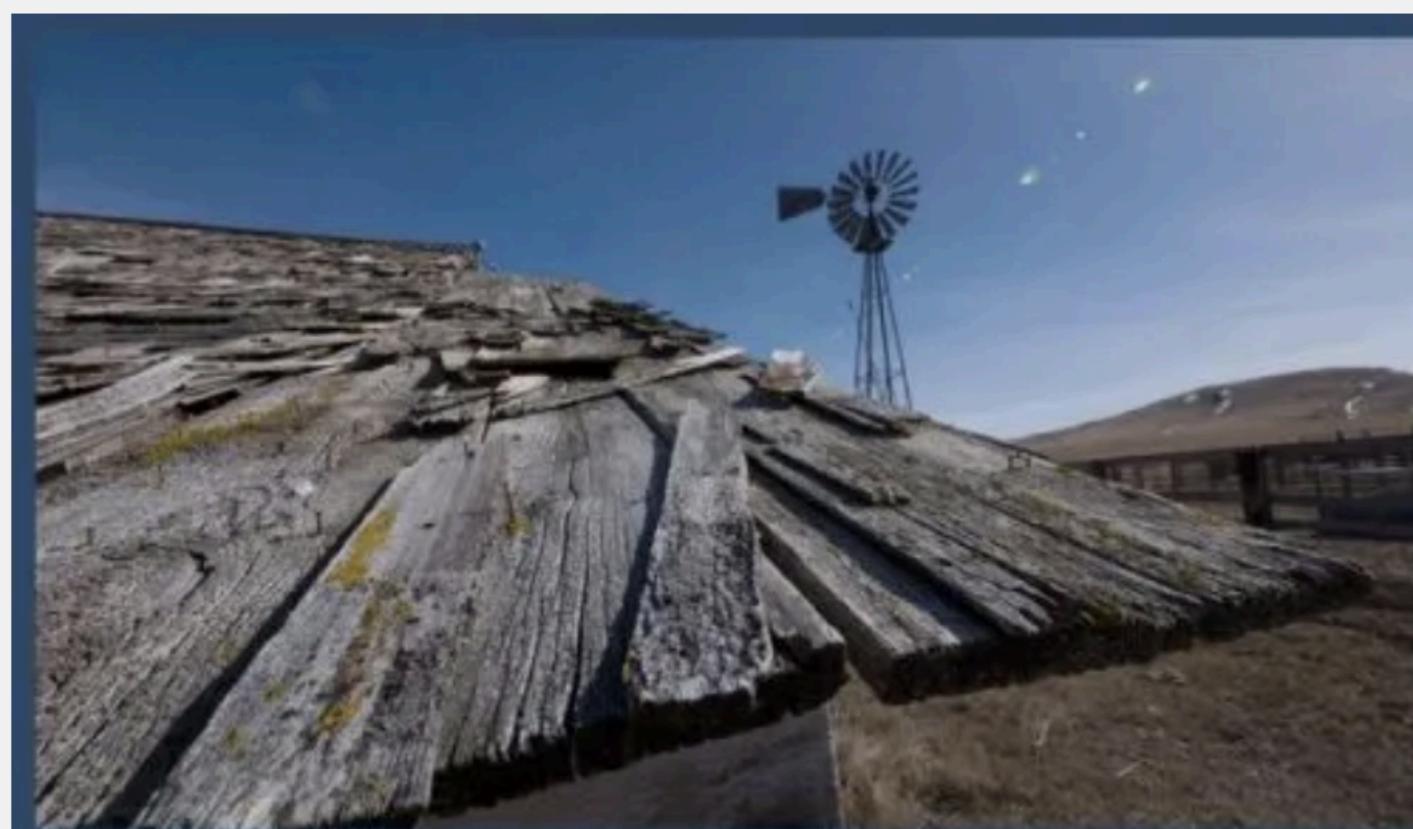
TUE, 3/9/24



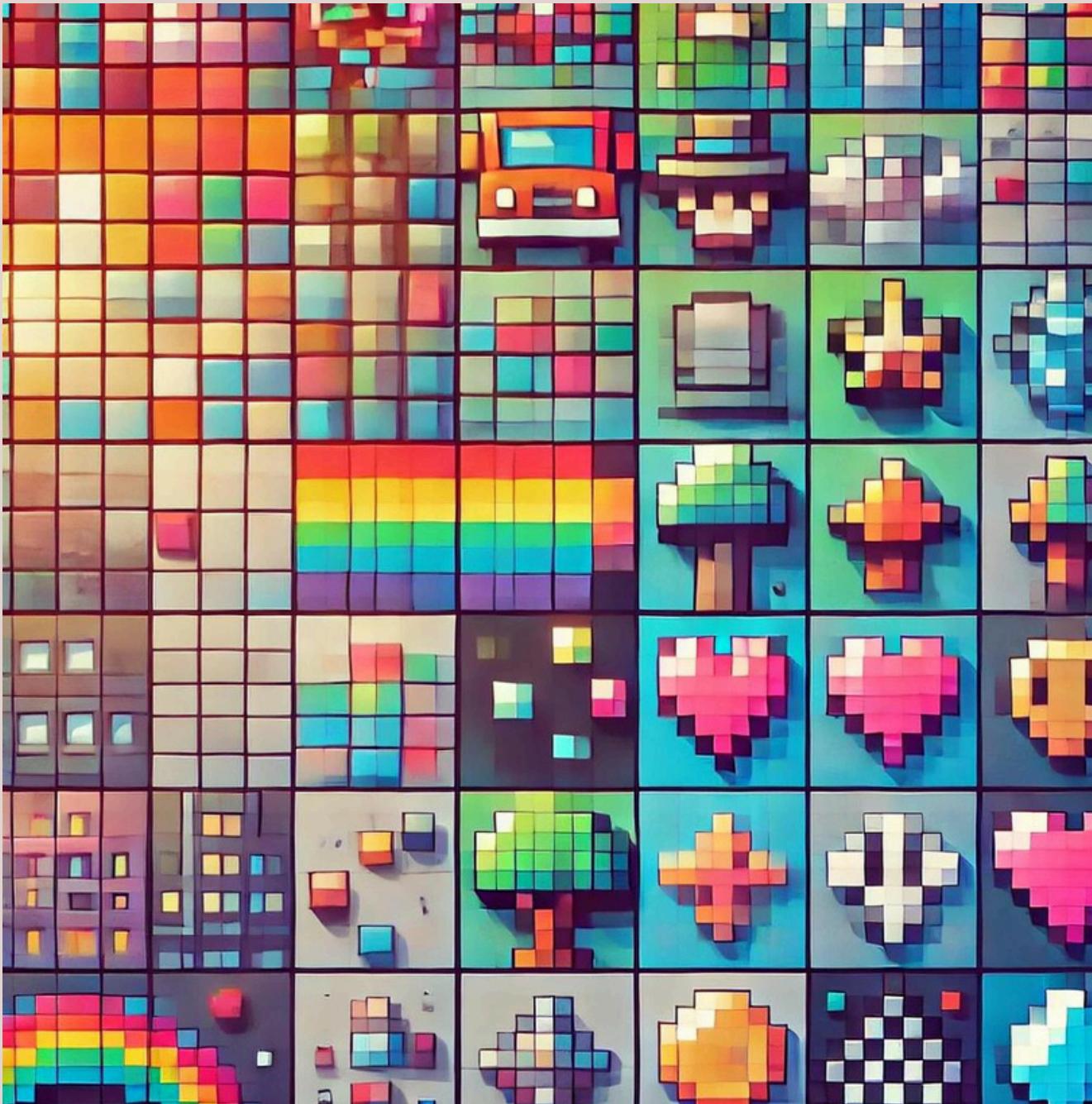
# OPENING



# OPENING



# SEKILAS

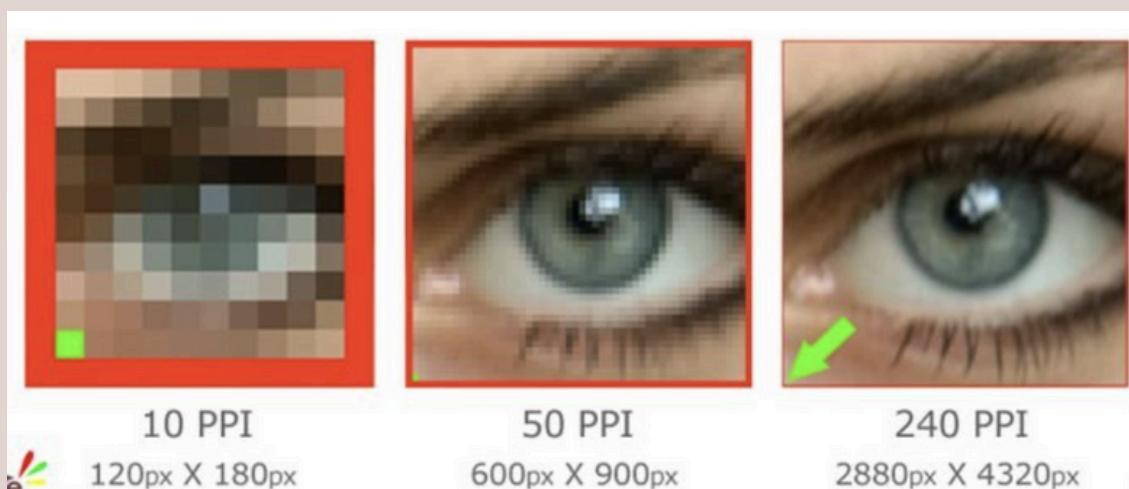
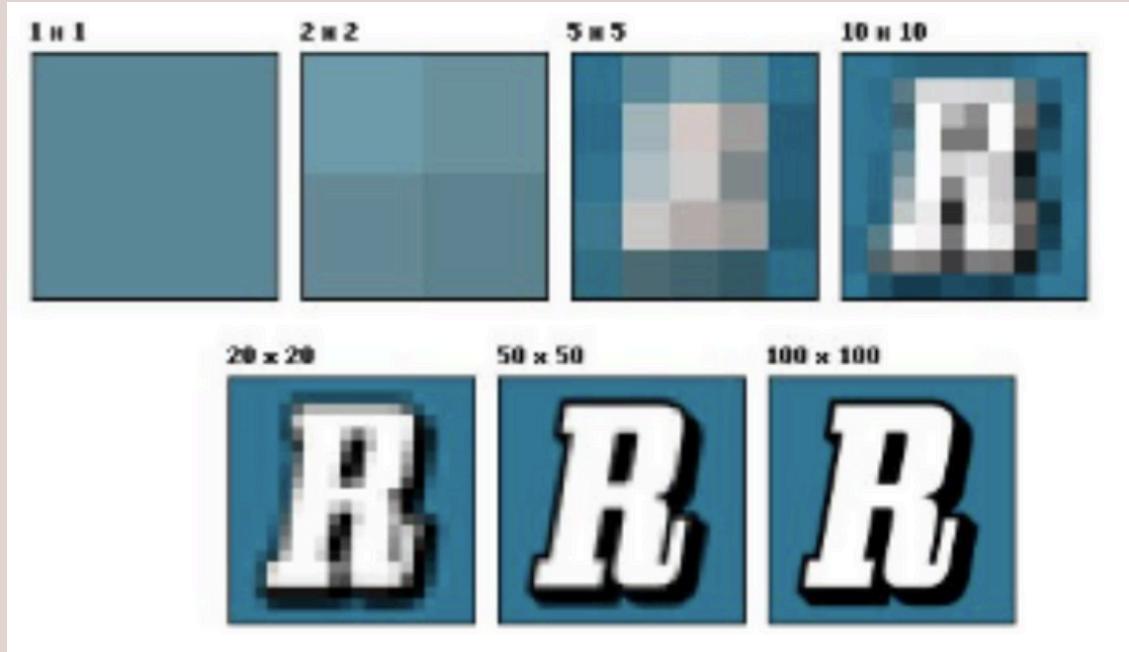


Dalam dunia **pengolahan citra digital**, kemampuan untuk mendeteksi dan mengenali warna secara real-time adalah salah **satu keterampilan dasar yang penting** (Game, robotika, gambar, video).



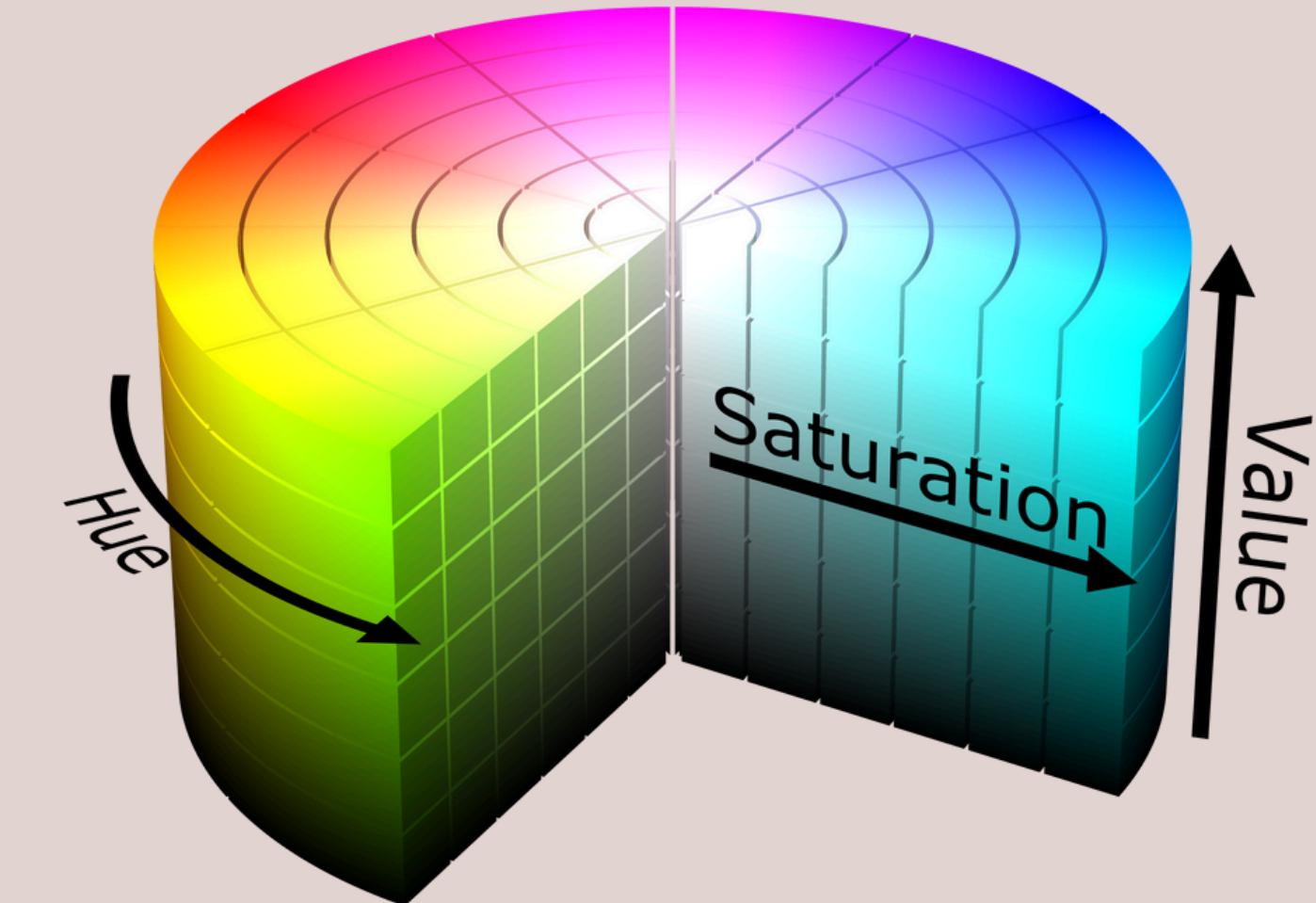
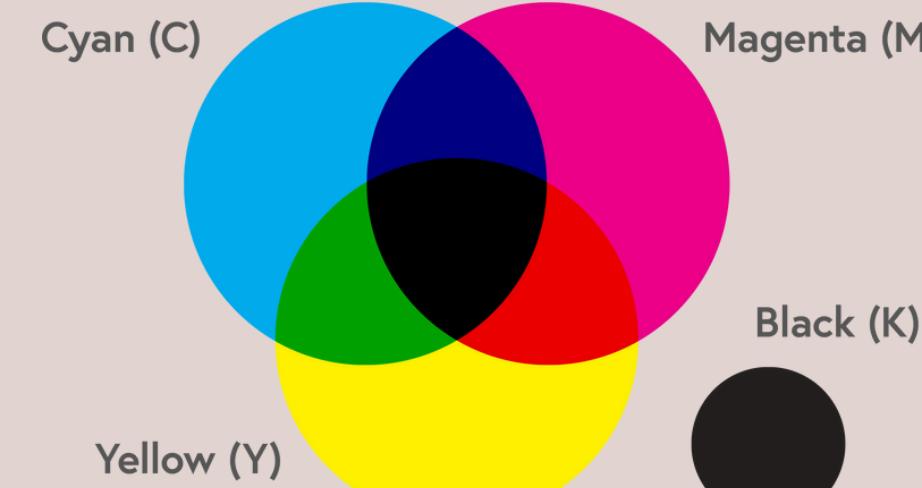
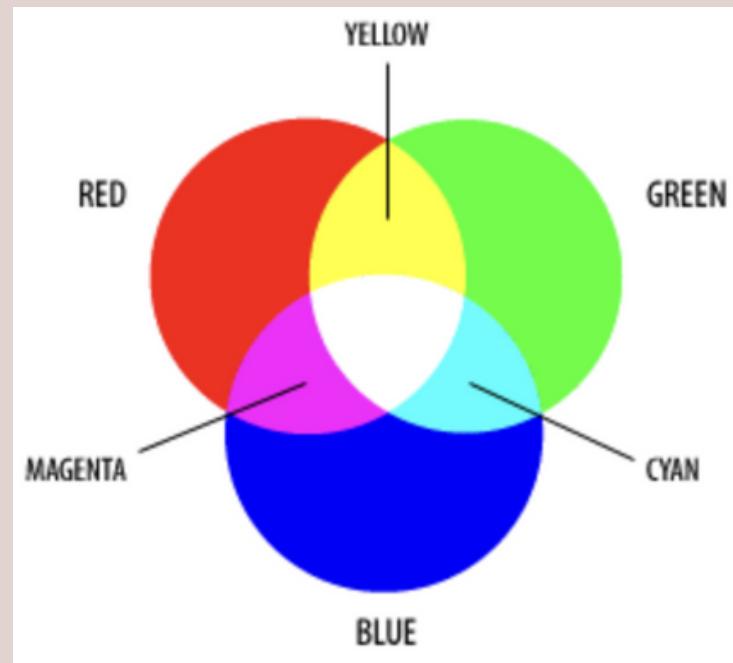
Kode Python yang melakukan **deteksi warna secara real-time menggunakan kamera dan pustaka OpenCV**.

# TENTANG PIXELS



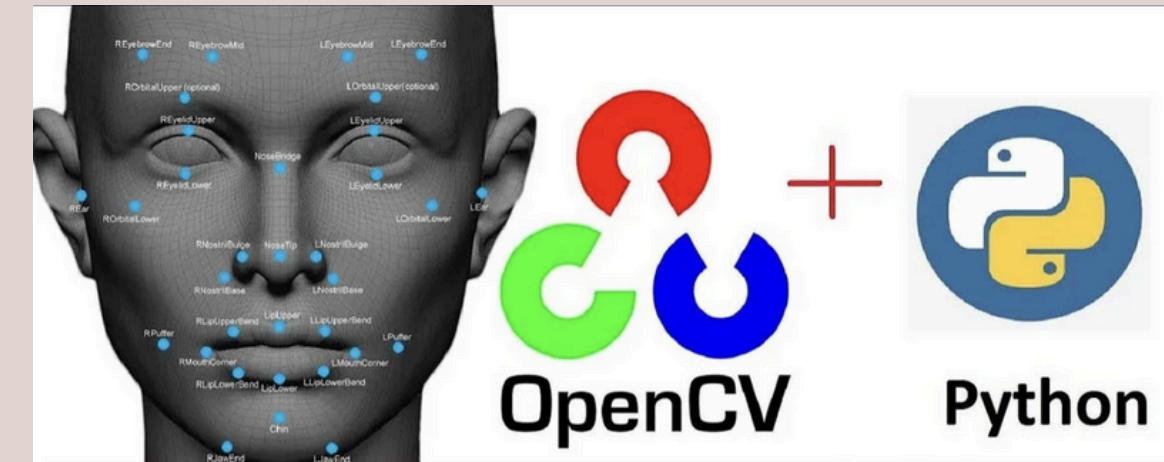
- **Elemen terkecil** gambar digital.
- Biasanya **berbentuk persegi** dan **mewakili satu titik** pada gambar.
- Setiap gambar digital terdiri dari jutaan piksel yang **disusun dalam grid** untuk membentuk gambar yang utuh.
- **Memiliki koordinat** pada gambar yang memiliki warna tertentu.
- **Resolusi** gambar, seperti 1920x1080, menunjukkan jumlah piksel dalam gambar (**lebar x tinggi**).

# TENTANG COLORS

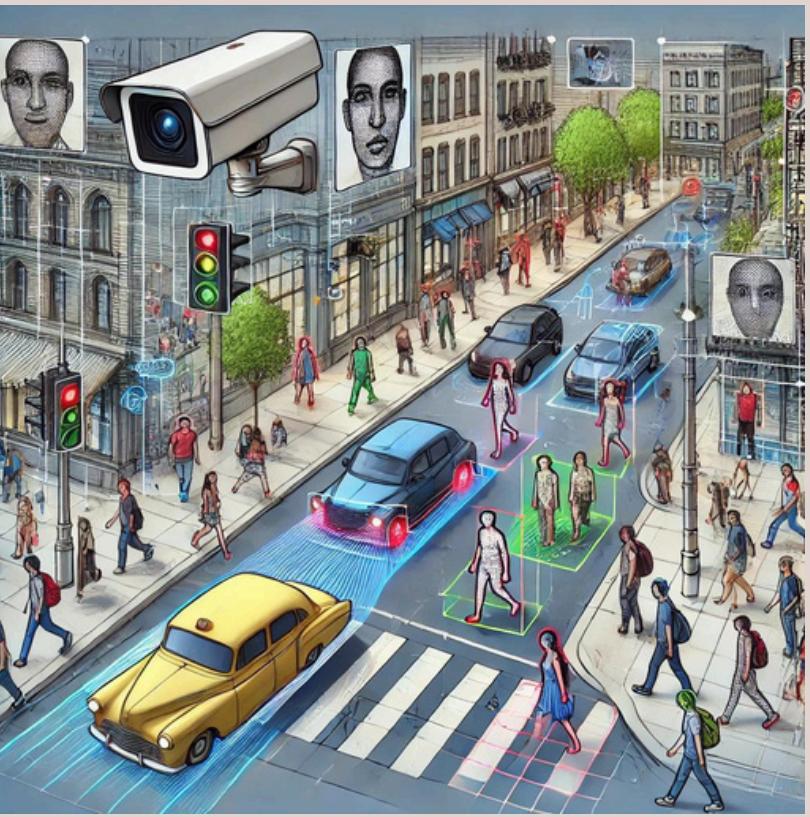


- **Sifat visual dari piksel** yang ditentukan oleh intensitas cahaya.
- Ditentukan oleh **model warna** (RGB, CMYK, HSV).
- Warna dalam piksel **dapat bervariasi berdasarkan tingkat intensitas**.

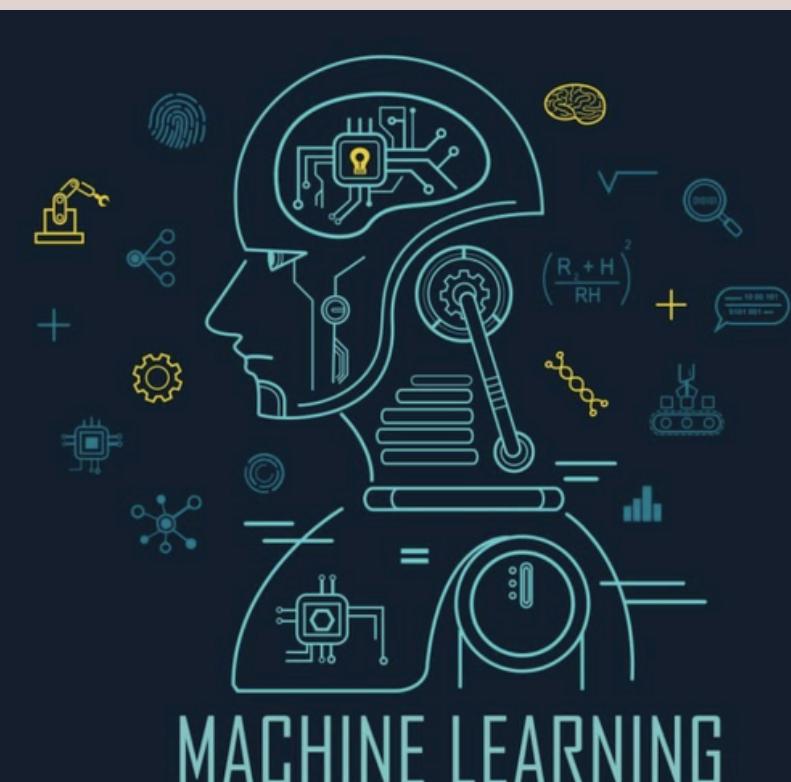
# APA ITU OPEN CV?



- **Open Source Computer Vision Library**
- Aplikasi **pengolahan gambar**.
- Menyediakan berbagai fungsi untuk **pemrosesan gambar, analisis video, deteksi objek**, dll.
- Mendukung penggunaan **computer system** (hardware dan software yang bekerja bersama).



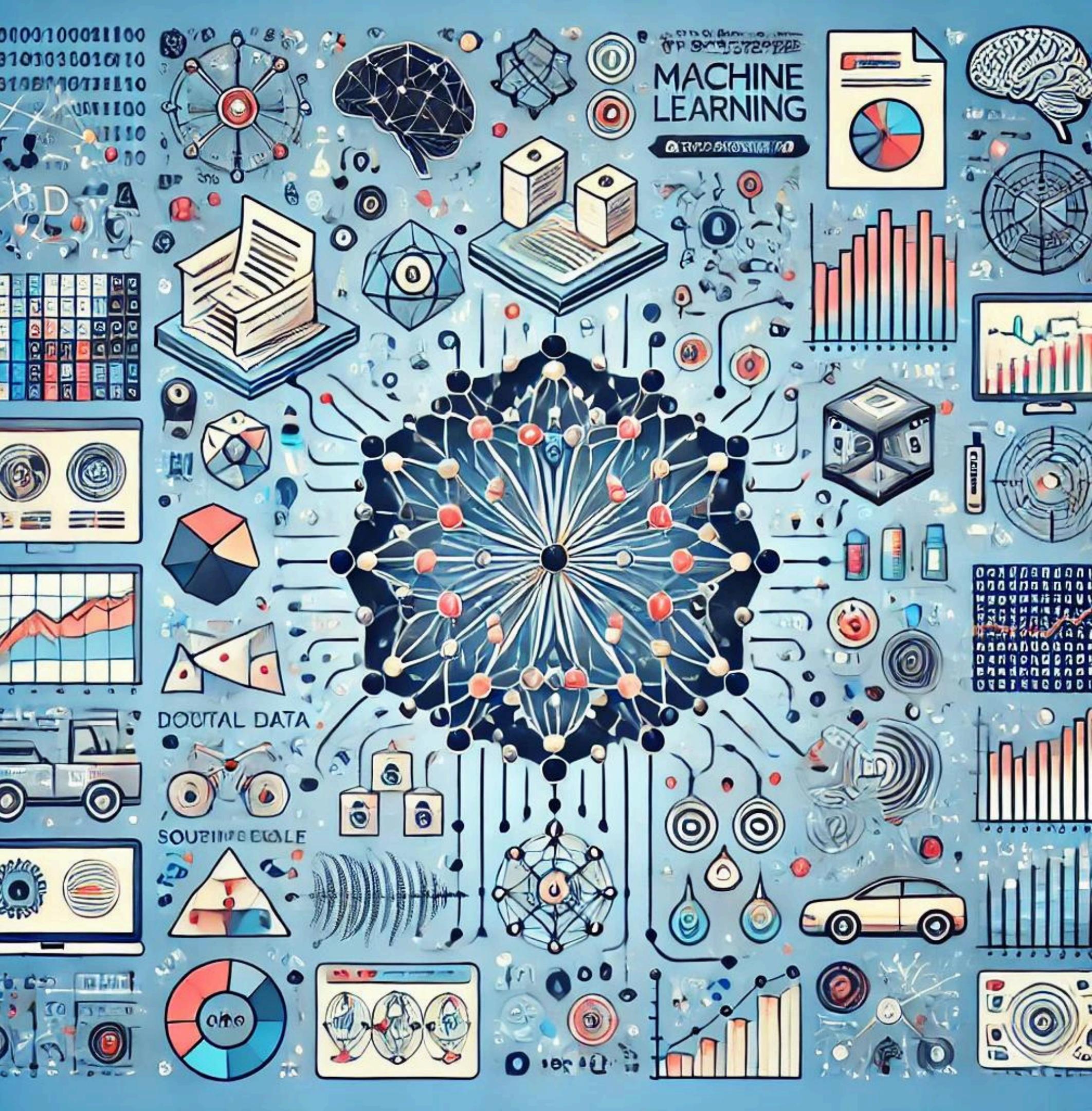
# COMPUTER SYSTEM VS MACHINE LEARNING

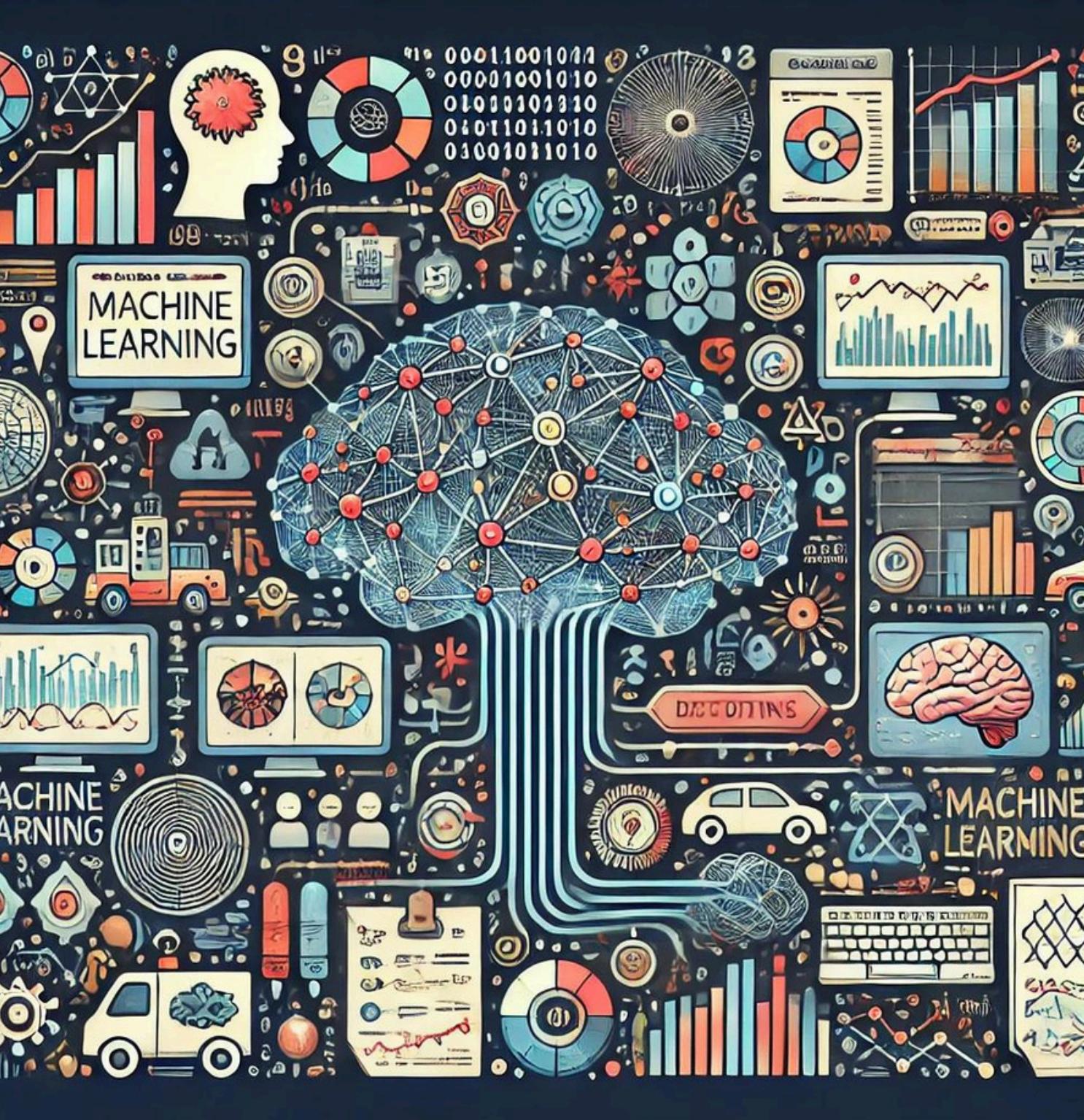


- Machine learning bergantung pada computer system untuk **menjalankan algoritmanyanya**.
- Machine learning diintegrasikan dalam computer system sebagai **bagian dari perangkat lunak**.
- Peneliti menggunakan computer system dengan **kapasitas penyimpanan tinggi** dan **kemampuan pemrosesan cepat** untuk mengembangkan dan menguji algoritma baru pada **dataset besar**.

# CODING

## FROM PIXELS TO COLORS





# INISIASI KAMERA

```
import cv2

# Initialize the camera
camera = cv2.VideoCapture(0)
camera.set(cv2.CAP_PROP_FRAME_WIDTH, 1280)
camera.set(cv2.CAP_PROP_FRAME_HEIGHT, 720)
```

- Parameter 0 menunjukkan penggunaan kamera default pada komputer.
- Mengatur ukuran frame video menjadi  $1280 \times 720$  piksel.

# PENENTUAN WARNA

Tiga parameter HSV:

- **Hue: Jenis** warna (0-179 derajat).
- **Saturation:Intensitas** warna (0-255 satuan).  
Saturasi rendah berarti warna lebih abu-abu.
- **Value: Kecerahan** warna (0-255 satuan).

```
def get_color_name(hue, saturation, value):
    """Determine the color name based on HSV values."""

    # If saturation and value are both low, the color is grayscale
    if value < 50:
        return "BLACK"
    elif saturation < 20:
        if value > 200:
            return "WHITE"
        else:
            return "GRAY"

    # Determining the hue color ranges
    if hue < 10 or hue >= 170:
        return "RED"
    elif hue < 25:
        return "ORANGE"
    elif hue < 35:
        return "YELLOW"
    elif hue < 85:
        return "GREEN"
    elif hue < 105:
        return "CYAN"
    elif hue < 130:
        return "BLUE"
    elif hue < 160:
        return "PURPLE"
    elif hue < 170:
        return "PINK"
    else:
        return "RED"
```



```
while True:  
    # Capture frame-by-frame  
    ret, frame = camera.read()  
    if not ret:  
        break
```

# FRAME VIDEO

## LOOP TAK TERBATAS

Berhenti ketika klik ESC

## RET

Boolean yang menunjukkan  
apakah frame berhasil dibaca

## IF NOT RET

Jika frame tidak terbaca

# KONVERSI WARNA DAN PUSAT FRAME

```
# Convert from BGR to HSV
hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
height, width, _ = frame.shape

# Determine the center point of the frame
cx, cy = width // 2, height // 2

# Get HSV values from the center point
pixel_center = hsv_frame[cy, cx]
hue, saturation, value = pixel_center
```

- **Mengambil nilai HSV dari piksel** yang berada di titik tengah frame.
- **Memecah nilai HSV** dari piksel tengah menjadi tiga variabel yang terpisah.





# PENENTUAN WARNA

- `get_color_name()` untuk **menentukan nama warna berdasarkan nilai HSV.**
- Mengonversi nilai BGR ke integer dan menyimpannya dalam variabel **b, g, dan r.**

```
# Determine the color name
color = get_color_name(hue, saturation, value)

# Get BGR values from the center point for the circle and text color
pixel_center_bgr = frame[cy, cx]
b, g, r = map(int, pixel_center_bgr)
```

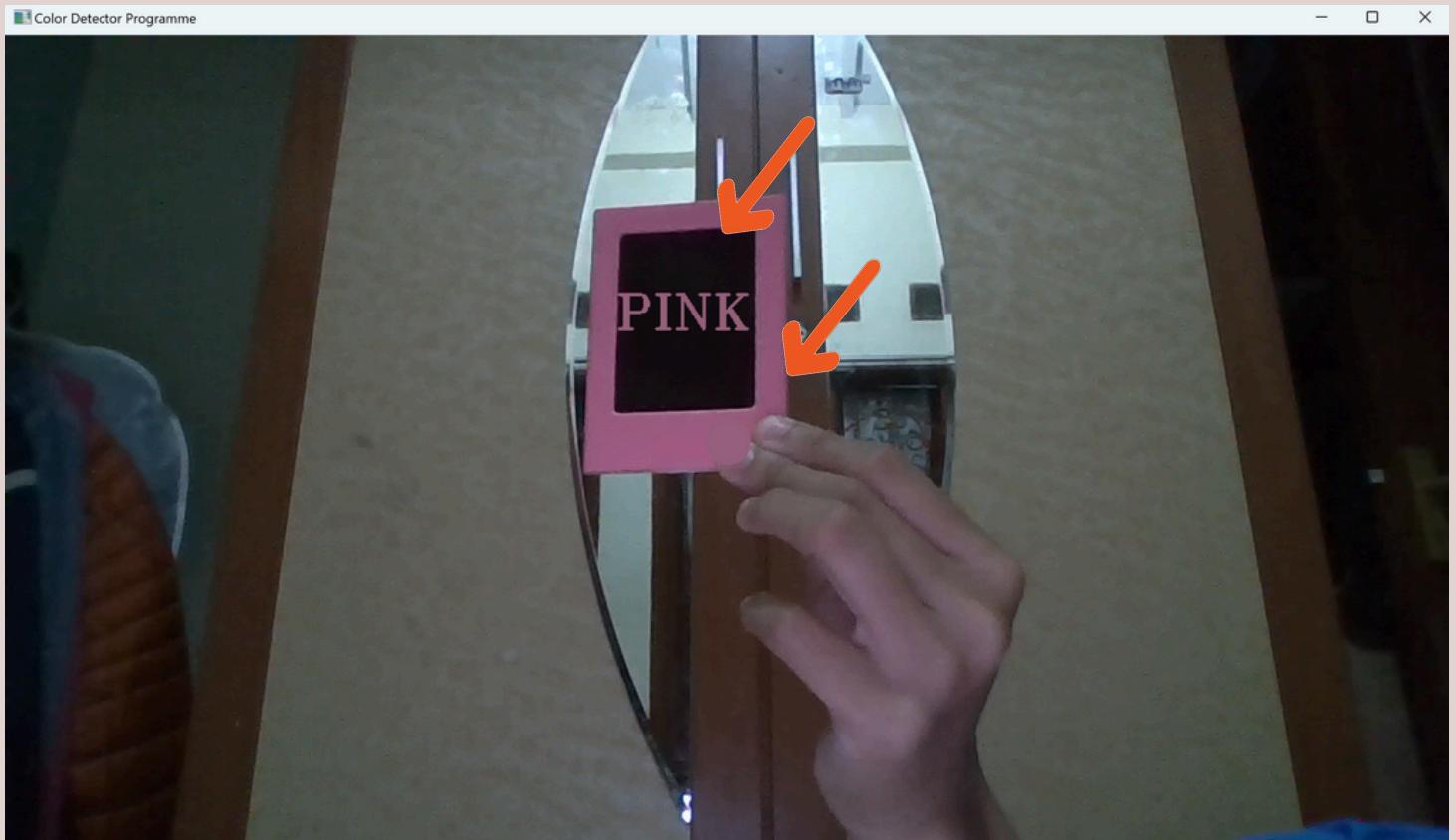
# MENAMPILKAN LINGKARAN DAN TEKS

- $(cx, cy)$ : **Coordinates** for the centre of the circle.
- 20: **Circle radius**
- $(b, g, r)$ : **Circle colour** in BGR format
- -1: **Fills the circle**.
- $(cx - 100, cy - 100)$ : **Text position**
- cv2.FONT\_HERSHEY\_TRIPLEX: The **font type** used
- 1.5: **Font size**

```
# Draw a circle in the center of the frame with the detected color
cv2.circle(frame, (cx, cy), 20, (b, g, r), -1) # Use detected color

# Display the color name on the frame with the detected color and chosen font
cv2.putText(frame, color, (cx - 100, cy - 100), cv2.FONT_HERSHEY_TRIPLEX, 1.5, (b, g, r), 2)

# Show the frame with the color detector title
cv2.imshow("Color Detector Programme", frame)
```



# CLOSING

```
# Exit the loop when 'ESC' is pressed
if cv2.waitKey(1) & 0xFF == 27:
    break

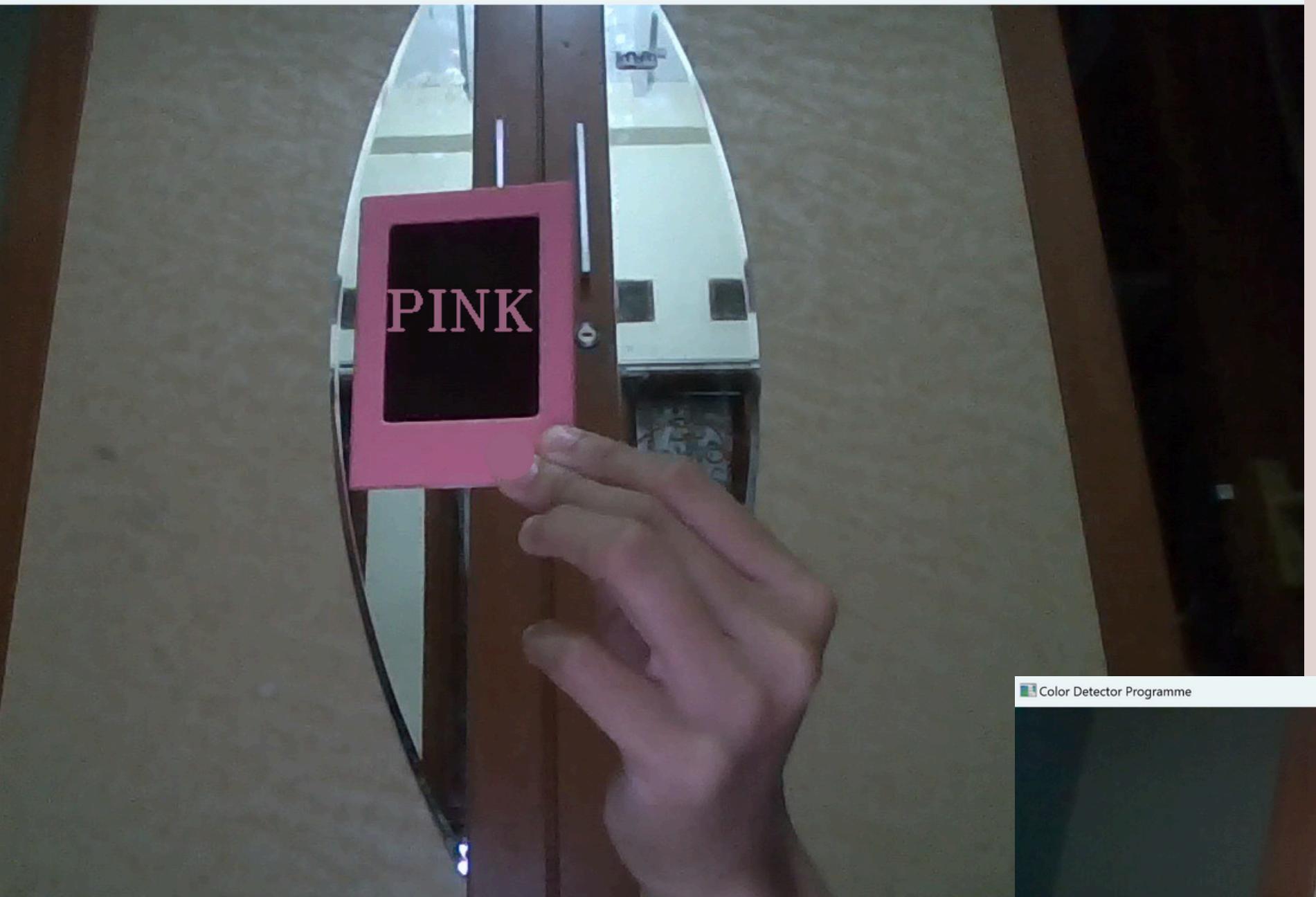
# Release the camera and close all OpenCV windows
camera.release()
cv2.destroyAllWindows()
```

- “cv2.waitKey(1)” : Akan menunggu setiap 1 milidetik untuk input keyboard (**real-time**)
- **ESC -> Break**
- **Kamera terhenti** dan **menutup semua jendela** OpenCV yang terbuka



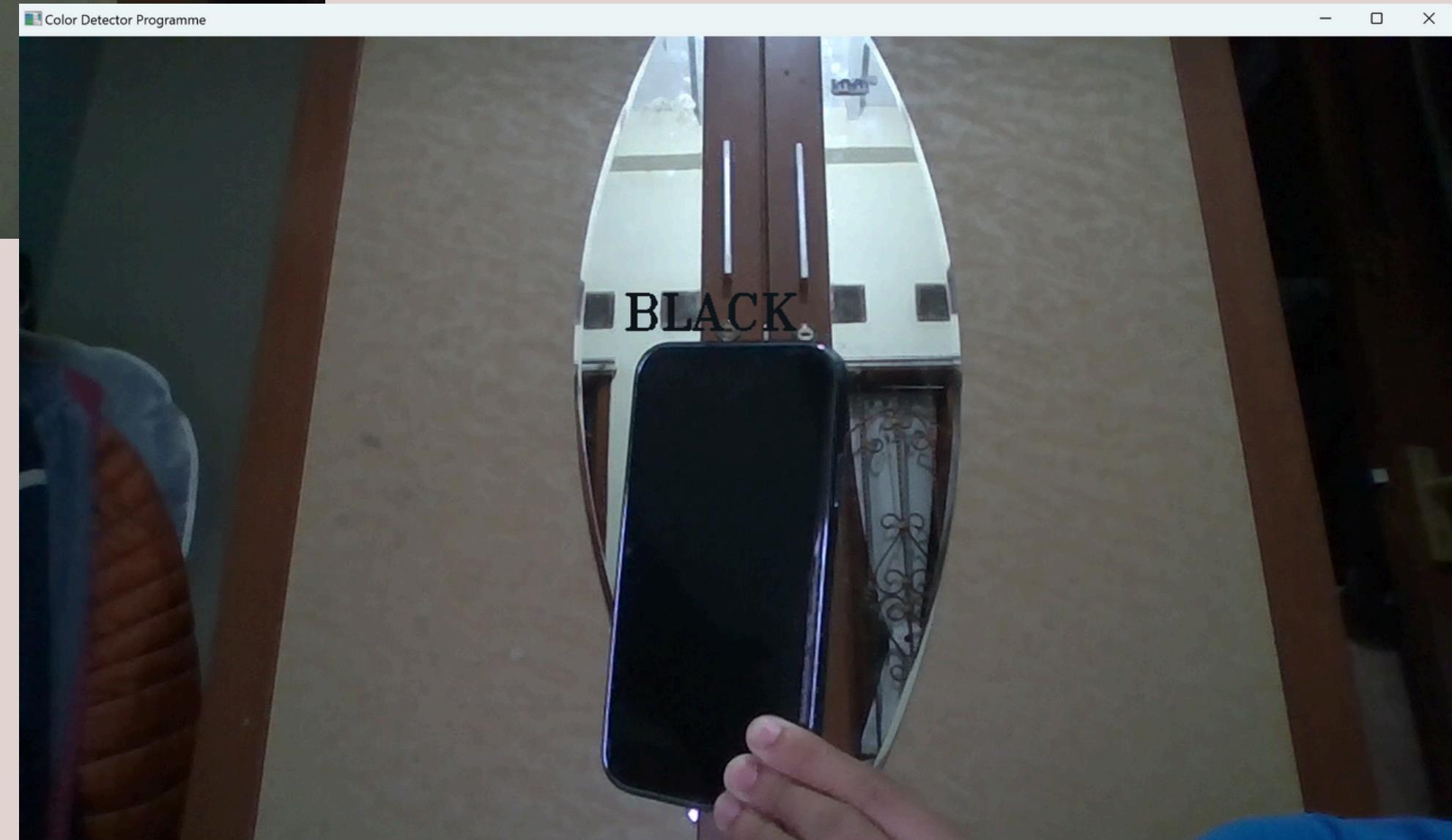
# THE RESULT

**Disclaimer:** Deteksi warna mungkin saja tidak akurat



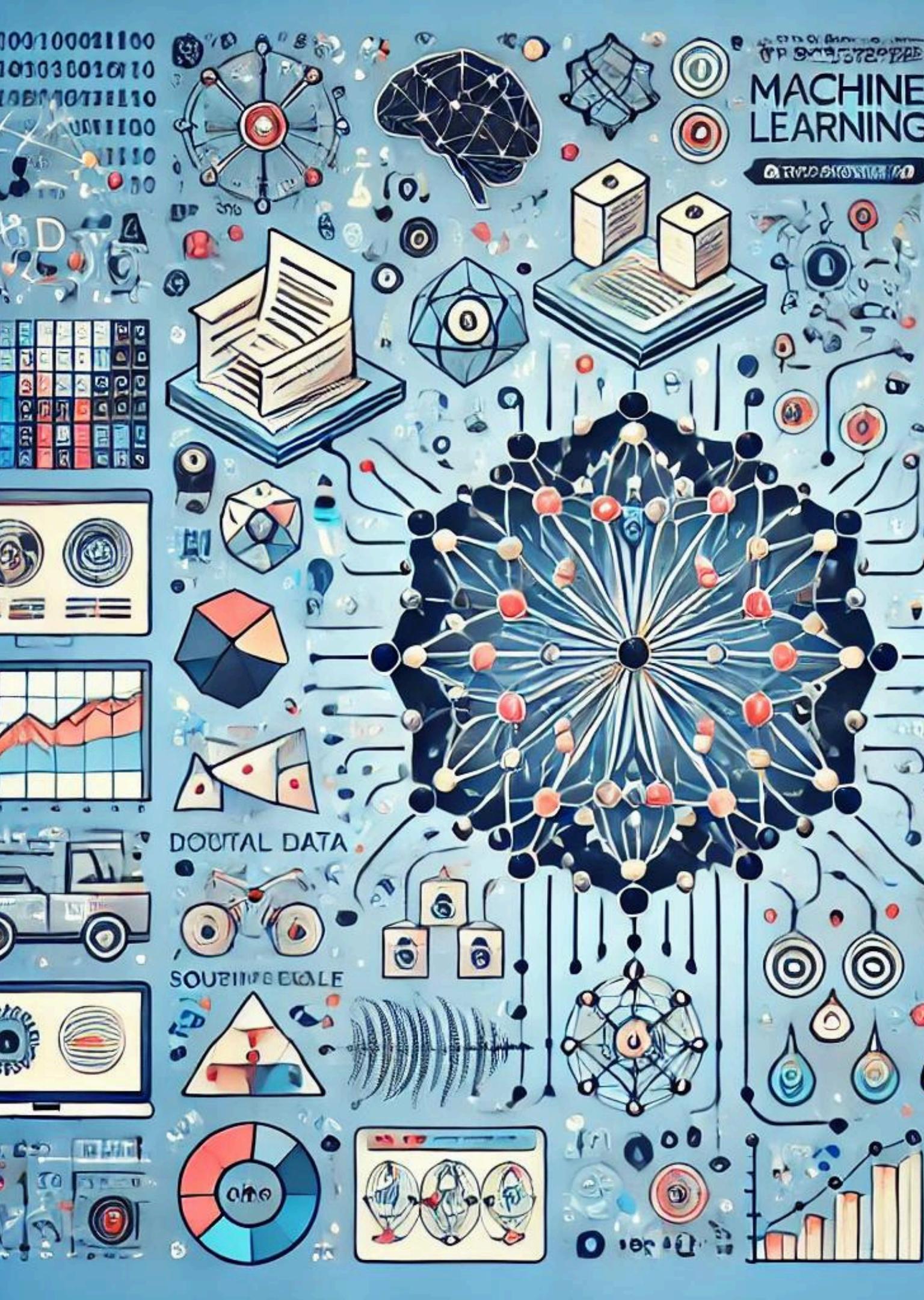
# THE RESULT

**Disclaimer:** Deteksi warna mungkin saja tidak akurat



# FUTURE PROJECT

- Klasifikasi warna yang lebih rumit
- Sensititas scanning
- Pengenalan objek



# From Pixels to Colors: Real-Time Color Detection in Images with OpenCV in Python



Athaya Zahrani Irmansyah

6 min read · 7 hours ago



1



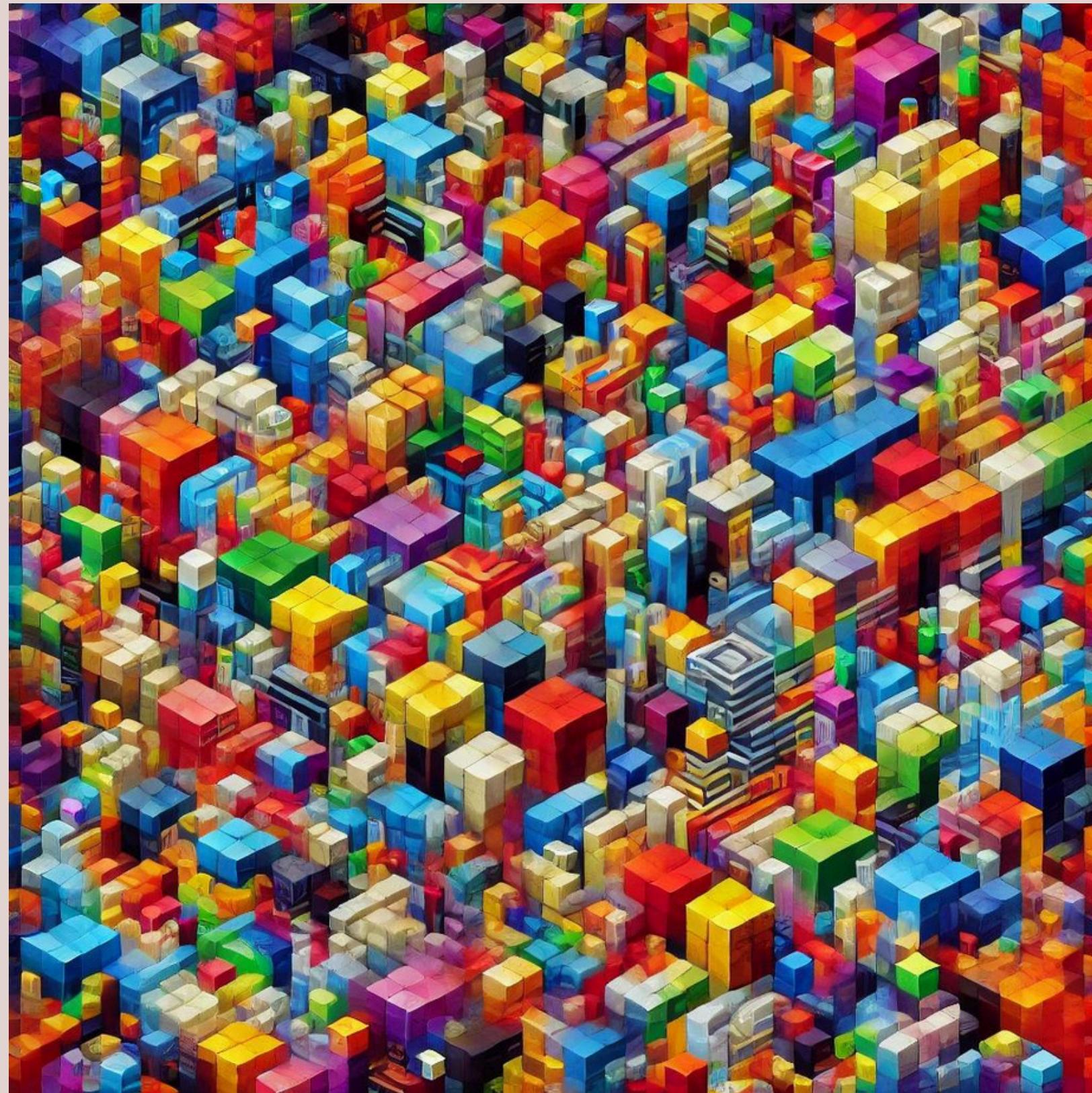
...

*In the world of digital image processing and computer vision, the ability to detect and recognize colour in real-time is one of the important basic skills. By using Python and the OpenCV library, we can create an application that can recognize colours from videos or images (in pixel form) taken directly by the camera. This article will cover Python code that performs real-time colour detection using the camera and the OpenCV library.*

**6 MINS READ IN  
MEDIUM**



**LET'S TRY!**



**THANK YOU**

**ATHAYA ZAHRANI I**  
PORTFOLIO PROJECT MODULE 3  
TUE, 3/9/24