

**PANDUAN LANGKAH-LANGKAH INSTALASI DAN IMPLEMENTASI
PROGRAM DETEKSI WARNA OBJEK MENGGUNAKAN COMPUTER
VISION DENGAN OPENCV DI JUPYTER NOTEBOOK**



Disusun Oleh:

- | | |
|---------------------------------|--------------|
| 1. Muh. Alfin Ikram Mullah R.L. | (5323600001) |
| 2. M. Nafis Ar Rosyid As Salam | (5323600019) |
| 3. Pipit Handayani Tiyas P. | (5323600023) |
| 4. Kevin Fiqer | (5323600024) |
| 5. M. Bayu Iskandar | (5323600025) |

**PROGRAM STUDI SARJANA TERAPAN
TEKNOLOGI REKAYASA MULTIMEDIA
JURUSAN TEKNOLOGI MULTIMEDIA KREATIF
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA**

2024

Urutan Cara Kerja Menggunakan Kode Deteksi Warna

Berikut adalah langkah-langkah terperinci dari instalasi hingga menjalankan kode deteksi warna menggunakan **Anaconda**, **Jupyter Notebook**, dan pustaka Python.

1. Instalasi Anaconda

1. **Download Anaconda:**

- Kunjungi situs resmi [Anaconda](#) dan unduh versi yang sesuai dengan sistem operasi Anda (Windows, macOS, atau Linux).

2. **Install Anaconda:**

- Jalankan file instalasi dan ikuti instruksi di layar.
- Pilih opsi untuk menambahkan Anaconda ke PATH (opsional, tetapi direkomendasikan).

3. **Verifikasi Instalasi:**

- Buka terminal atau Command Prompt dan ketik:
 - `conda --version`
 - Jika berhasil, akan menampilkan versi Anaconda yang terinstal.

2. Membuat dan Mengaktifkan Environment di Anaconda

1. **Membuat Environment Baru:**

- Buka terminal atau Anaconda Prompt, lalu buat environment baru dengan nama, misalnya, `deteksi_warna`:
- `conda create -n deteksi_warna python=3.9`

2. **Aktifkan Environment:**

- Jalankan perintah:
 - `conda activate deteksi_warna`
 - Anda akan melihat nama environment aktif di awal prompt terminal.

3. Instalasi Jupyter Notebook

1. **Install Jupyter Notebook di Environment:**

- Pastikan environment aktif, lalu jalankan perintah berikut untuk menginstal Jupyter Notebook:
 - `conda install jupyter`

2. **Jalankan Jupyter Notebook:**

- Masih dalam environment aktif, jalankan perintah berikut untuk membuka Jupyter Notebook:
 - `jupyter notebook`
 - Browser akan terbuka secara otomatis, menampilkan antarmuka Jupyter Notebook.

4. Instalasi Library yang Dibutuhkan

1. Install OpenCV, NumPy, dan Matplotlib:

- Pastikan Anda berada di environment deteksi_warna, lalu jalankan perintah berikut:
- `pip install opencv-python opencv-python-headless numpy matplotlib`

2. Verifikasi Instalasi Library:

- Buka Jupyter Notebook, buat file baru dengan ekstensi .ipynb, lalu jalankan kode berikut untuk memastikan library terinstal dengan benar:
- `import cv2`
- `import numpy as np`
- `import matplotlib.pyplot as plt`
-
- `print("Library berhasil diinstal!")`

5. Menjalankan Kode Deteksi Warna

1. Salin Kode Deteksi Warna:

- Gunakan kode berikut untuk mendeteksi warna:
- `import cv2`
- `import numpy as np`
- `import time`
-
- `# Rentang warna dalam HSV`
- `color_ranges = {`
- `"Merah": [(0, 50, 50), (10, 255, 255)],`
- `"Hijau": [(36, 50, 50), (85, 255, 255)],`
- `"Biru": [(96, 50, 50), (130, 255, 255)],`
- `"Kuning": [(26, 50, 50), (35, 255, 255)]`
- `}`
-
- `# Fungsi untuk mendeteksi warna`
- `def detect_color(hsv_frame):`
- `for color_name, (lower, upper) in color_ranges.items():`
- `mask = cv2.inRange(hsv_frame, np.array(lower, np.uint8), np.array(upper,`
- `np.uint8))`
- `if cv2.countNonZero(mask) > 0:`
- `return color_name`
- `return "Tidak ada warna terdeteksi"`
-
- `# Akses kamera`
- `cap = cv2.VideoCapture(0)`
-
- `while True:`
- `ret, frame = cap.read()`
- `if not ret:`
- `break`

-
- `hsv_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)`
- `detected_color = detect_color(hsv_frame)`
-
- `cv2.putText(frame, f"Warna: {detected_color}", (10, 50),`
`cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)`
- `cv2.imshow("Deteksi Warna", frame)`
-
- `if cv2.waitKey(1) & 0xFF == ord('q'):`
`break`
-
- `cap.release()`
- `cv2.destroyAllWindows()`

2. Jalankan Kode:

- Klik **Run** pada Jupyter Notebook untuk menjalankan kode.
- Kamera akan terbuka, dan warna yang terdeteksi akan ditampilkan di layar.

6. Penjelasan Cara Kerja Masking

1. Input Gambar:

- Kamera menangkap gambar real-time sebagai input.

2. Konversi ke HSV:

- Gambar diubah ke format HSV menggunakan `cv2.cvtColor()`.

3. Pembuatan Mask:

- Mask dibuat untuk setiap warna berdasarkan rentang HSV.
- Piksel putih pada mask menunjukkan area yang sesuai dengan warna target.

4. Analisis Piksel:

- Warna dianggap terdeteksi jika jumlah piksel putih pada mask melebihi ambang batas.

5. Visualisasi Hasil:

- Bounding box dan nama warna ditampilkan pada layar.