

KECERDASAN TIRUAN (UAS)

Nama: Tulus Prasetyo

Nim: 230741094

Prodi : Ilmu Komputer

Penjelasan Object Detection Project

```
from ultralytics import YOLO # Untuk memuat model YOLO dan melakukan deteksi
import cv2 # Untuk pemrosesan gambar dan video
import streamlit as st # Untuk membuat antarmuka web
from PIL import Image # Untuk manipulasi gambar
import numpy as np # Untuk operasi berbasis array
from collections import Counter # Untuk menghitung jumlah objek yang terdeteksi
```

Fungsi:

- Mengimpor pustaka yang diperlukan untuk memuat model YOLO, membaca data dari kamera, dan menampilkan hasil deteksi di aplikasi Streamlit.

2. Fungsi load_model

python

Salin kode

```
@st.cache_resource
def load_model(model_path):
    return YOLO(model_path)
```

Penjelasan:

- **@st.cache_resource:** Cache hasil dari fungsi load_model agar model hanya dimuat satu kali. Ini meningkatkan efisiensi.
- **YOLO(model_path):** Memuat model YOLO dari file yang ditentukan oleh model_path.

3. Fungsi display_results

python

Salin kode

```
def display_results(image, results):  
    boxes = results.boxes.xyxy.cpu().numpy() # Koordinat bounding box  
    scores = results.boxes.conf.cpu().numpy() # Skor kepercayaan  
    labels = results.boxes.cls.cpu().numpy() # Indeks kelas  
    names = results.names # Nama kelas objek  
  
    detected_objects = []  
  
    for i in range(len(boxes)):  
        if scores[i] > 0.5: # Ambang batas kepercayaan  
            x1, y1, x2, y2 = boxes[i].astype(int)  
            label = names[int(labels[i])]  
            score = scores[i]  
            detected_objects.append(label)  
            cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2) # Gambar kotak  
            cv2.putText(image, f'{label}: {score:.2f}', (x1, y1 - 10),  
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2) # Teks label  
  
    return image, detected_objects
```

Penjelasan:

- Mengambil hasil deteksi (bounding box, skor, label) dari model YOLO.
- Gambar kotak di sekitar objek yang terdeteksi pada gambar.
- Mengembalikan gambar yang sudah diberi kotak dan daftar objek yang terdeteksi.

4. Fungsi main

python

Salin kode

```
def main():  
    st.title("Real-time Object Detection with YOLO")
```

```

st.sidebar.title("Settings")

model_path = "yolo11n.pt" # Path ke model YOLO
model = load_model(model_path)

# Tombol checkbox untuk memulai/menghentikan deteksi objek
run_detection = st.sidebar.checkbox("Start/Stop Object Detection",
key="detection_control")

# Jika checkbox diaktifkan
if run_detection:
    cap = cv2.VideoCapture(0) # Buka kamera
    st_frame = st.empty() # Placeholder untuk frame video
    st_detection_info = st.empty() # Placeholder untuk informasi deteksi

    while True:
        ret, frame = cap.read()
        if not ret:
            st.warning("Failed to capture image.")
            break

        frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Konversi ke RGB untuk
Streamlit
        results = model.predict(frame, imgsz=640) # Deteksi objek

        frame, detected_objects = display_results(frame, results[0]) # Proses hasil deteksi

        st_frame.image(frame, channels="RGB", use_column_width=True) # Tampilkan
frame
        if detected_objects:
            object_counts = Counter(detected_objects)

```

```

        detection_info = "\n".join([f"{obj}: {count}" for obj, count in
object_counts.items()])
    else:
        detection_info = "No objects detected."

    st_detection_info.text(detection_info) # Tampilkan informasi deteksi

    if not st.session_state.detection_control: # Hentikan jika checkbox dimatikan
        break

    cap.release()

```

Penjelasan:

- **Tujuan utama:** Membuat antarmuka web untuk mendeteksi objek secara real-time menggunakan kamera.
- Jika tombol checkbox diaktifkan, aplikasi:
 - Membuka kamera.
 - Melakukan deteksi dengan YOLO untuk setiap frame video.
 - Menampilkan hasil deteksi (kotak, label, informasi) di antarmuka web.
- Jika checkbox dimatikan, aplikasi berhenti.

5. Pemanggilan main

python

Salin kode

```

if __name__ == "__main__":
    main()

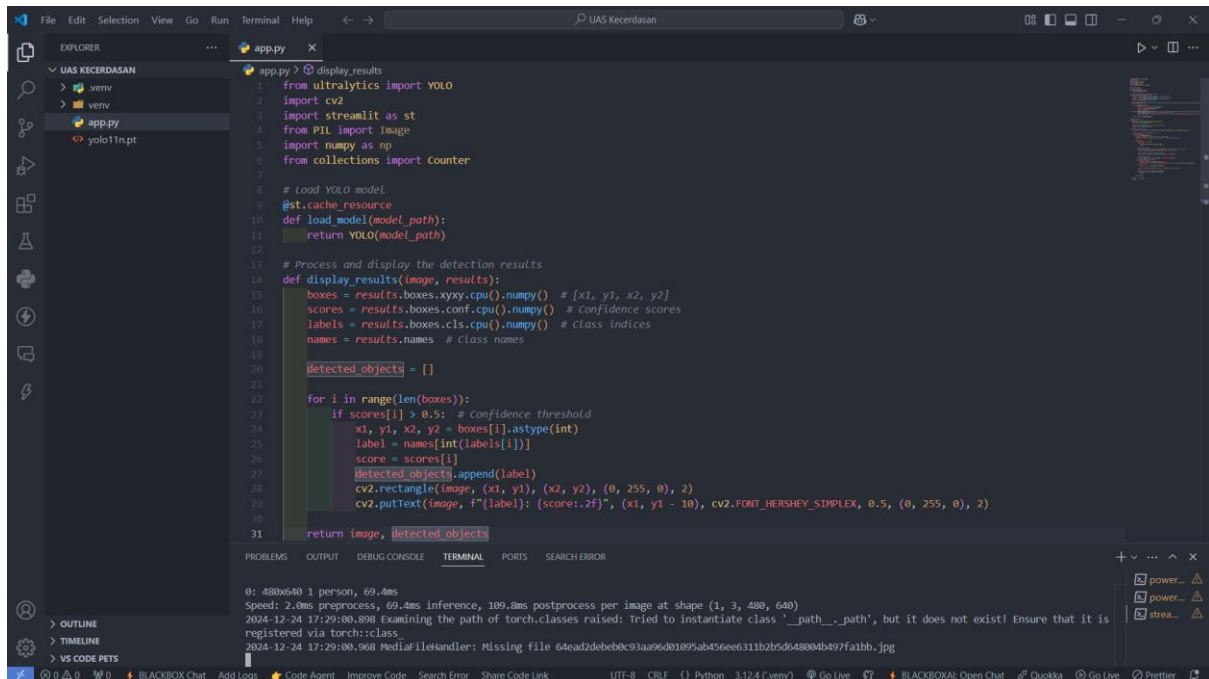
```

Penjelasan:

- Mengeksekusi fungsi main saat skrip dijalankan sebagai file utama.
- Pastikan typo `_name_` diperbaiki menjadi `__name__`, dan `_main_` menjadi `__main__`.

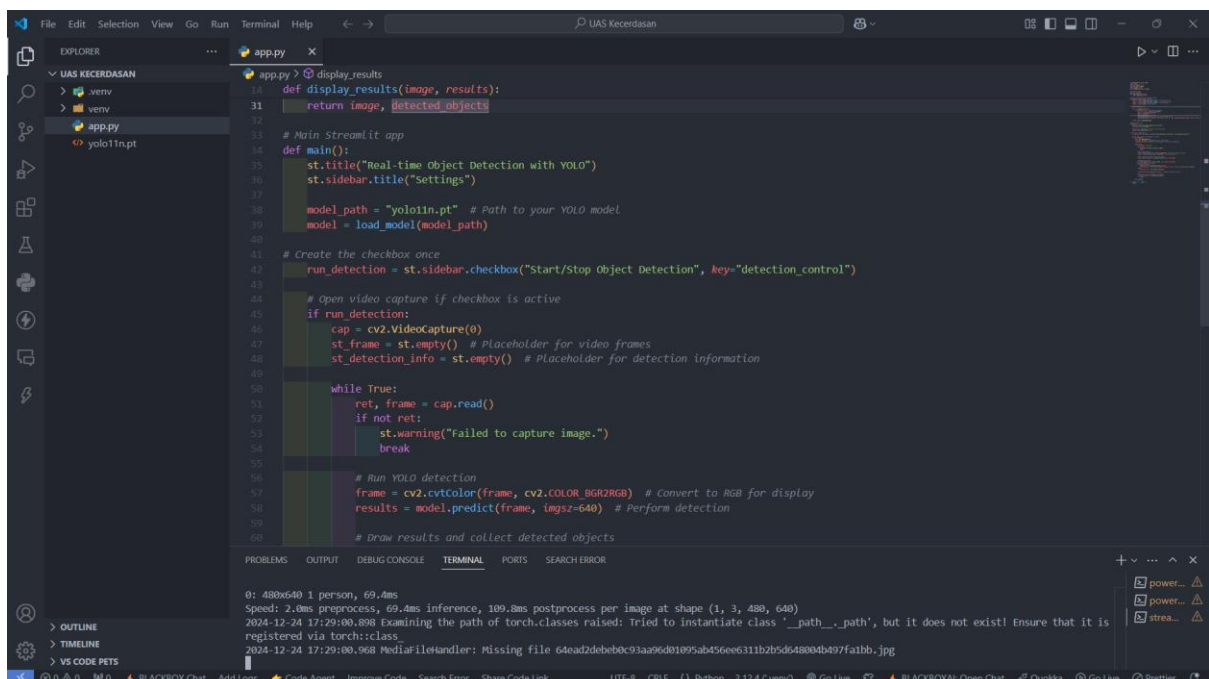
Kesimpulan

Kode ini adalah aplikasi Streamlit untuk deteksi objek real-time menggunakan YOLO. Pengguna dapat mengaktifkan/mematikan deteksi melalui antarmuka web, melihat hasil deteksi dalam video, dan mendapatkan informasi tentang objek yang terdeteksi.



```
1 from ultralytics import YOLO
2 import cv2
3 import streamlit as st
4 from PIL import Image
5 import numpy as np
6 from collections import Counter
7
8 # Load YOLO model
9 @st.cache_resource
10 def load_model(model_path):
11     return YOLO(model_path)
12
13 # Process and display the detection results
14 def display_results(image, results):
15     boxes = results.boxes.xyxy.cpu().numpy() # [x1, y1, x2, y2]
16     scores = results.boxes.conf.cpu().numpy() # Confidence scores
17     labels = results.boxes.cls.cpu().numpy() # Class indices
18     names = results.names # Class names
19
20     detected_objects = []
21
22     for i in range(len(boxes)):
23         if scores[i] > 0.5: # Confidence threshold
24             x1, y1, x2, y2 = boxes[i].astype(int)
25             label = names[int(labels[i])]
26             score = scores[i]
27             detected_objects.append(label)
28             cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
29             cv2.putText(image, f'{label}: {score:.2f}', (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
30
31     return image, detected_objects
```

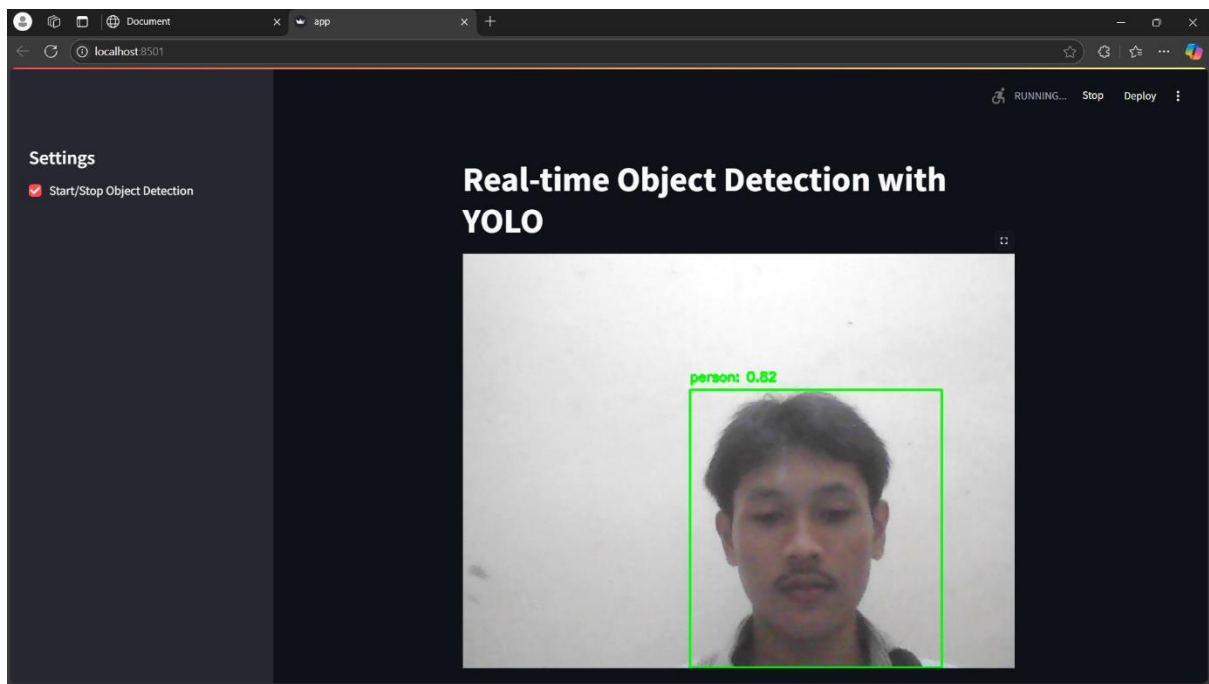
0: 480x640 1 person, 69.4ms
Speed: 2.0ms preprocess, 69.4ms inference, 109.8ms postprocess per image at shape (1, 3, 480, 640)
2024-12-24 17:29:00.898 Examining the path of torch.classes raised: Tried to instantiate class '._path_.path', but it does not exist! Ensure that it is registered via torch.class.
2024-12-24 17:29:00.968 MediaFileHandler: Missing file 64ead2debe0c93aa96d01095ab456ee6311b2b5d648004b497fa1bb.jpg



```
14 def display_results(image, results):
15     return image, detected_objects
16
17 # Main Streamlit app
18 def main():
19     st.title("Real-time Object Detection with YOLO")
20     st.sidebar.title("Settings")
21
22     model_path = "yolo11n.pt" # Path to your YOLO model
23     model = load_model(model_path)
24
25     # Create the checkbox once
26     run_detection = st.sidebar.checkbox("Start/Stop Object Detection", key="detection_control")
27
28     # Open video capture if checkbox is active
29     if run_detection:
30         cap = cv2.VideoCapture(0)
31         st_frame = st.empty() # Placeholder for video frames
32         st_detection_info = st.empty() # Placeholder for detection information
33
34         while True:
35             ret, frame = cap.read()
36             if not ret:
37                 st.warning("Failed to capture image.")
38                 break
39
40             # Run YOLO detection
41             frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert to RGB for display
42             results = model.predict(frame, imgsz=640) # Perform detection
43
44             # Draw results and collect detected objects
```

0: 480x640 1 person, 69.4ms
Speed: 2.0ms preprocess, 69.4ms inference, 109.8ms postprocess per image at shape (1, 3, 480, 640)
2024-12-24 17:29:00.898 Examining the path of torch.classes raised: Tried to instantiate class '._path_.path', but it does not exist! Ensure that it is registered via torch.class.
2024-12-24 17:29:00.968 MediaFileHandler: Missing file 64ead2debe0c93aa96d01095ab456ee6311b2b5d648004b497fa1bb.jpg

```
File Edit Selection View Go Run Terminal Help UAS Kecerdasan
EXPLORER
  UAS KECERDASAN
    .venv
    .venv
    app.py
    yolo11n.pt
  app.py
    def main():
        # Draw results and collect detected objects
        frame, detected_objects = display_results(frame, results[0])
        # Display video feed
        st_frame.image(frame, channels="RGB", use_column_width=True)
        # Display detection information
        if detected_objects:
            object_counts = Counter(detected_objects)
            detection_info = "\n".join([f"{obj}: {count}" for obj, count in object_counts.items()])
        else:
            detection_info = "No objects detected."
        st_detection_info.text(detection_info) # Update detection info text
        # Break the loop if checkbox is unchecked
        if not st.session_state.detection_control:
            break
        cap.release()
    if __name__ == "__main__":
        main()
    80
    81
    82
    83
    84
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
0: 480x640 1 person, 69.4ms
Speed: 2.0ms preprocess, 69.4ms inference, 109.8ms postprocess per image at shape (1, 3, 480, 640)
2024-12-24 17:29:00.898 Examining the path of torch.classes raised: tried to instantiate class '._path_._path_', but it does not exist! Ensure that it is
registered via torch.class
2024-12-24 17:29:00.968 MediaFileHandler: Missing file 64ead2debebc93aa96d81095ab456ee6311b2b5d648004b97fa1bb.jpg
```

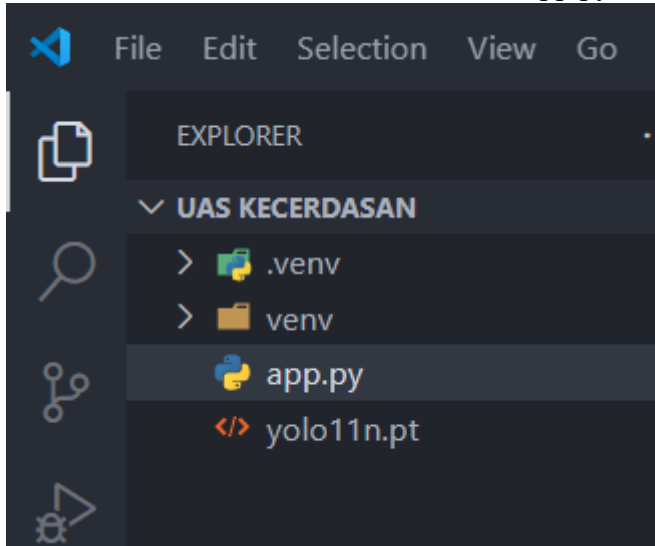


Tutorial Sampai ke Hasil Deploy

1. Step Pertama

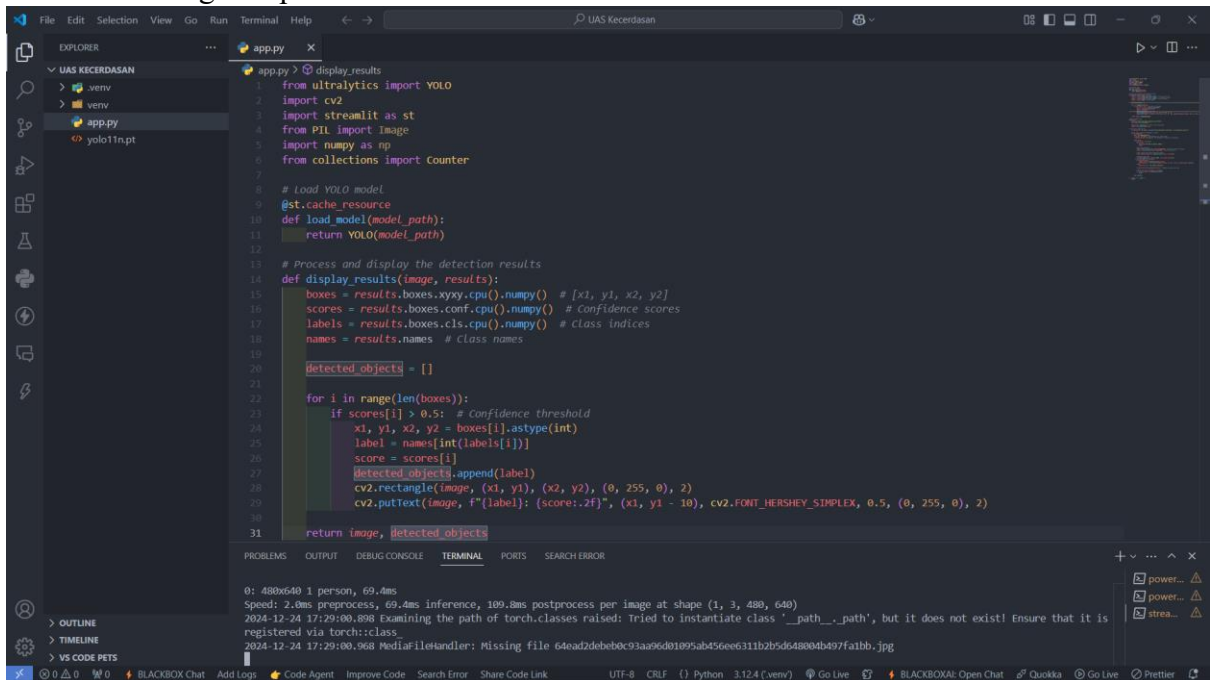
Buat Folder baru untuk Proyek, Contohnya: UAS KECERDASAN setelah folder di buat lalu masukkan file yolo11n.pt

Didalam folder tersebut, buat file utama app.py untuk menjalankan kode python.



2. Lalu

Buatkan Codingan seperti berikut



```
1: def display_results:
2:     def display_results(image, results):
31:         return image, detected_objects
32:
33: # Main Streamlit app
34: def main():
35:     st.title("Real-time Object Detection with YOLO")
36:     st.sidebar.title("Settings")
37:
38:     model_path = "yolo11n.pt" # Path to your YOLO model
39:     model = load_model(model_path)
40:
41:     # Create the checkbox once
42:     run_detection = st.sidebar.checkbox("Start/Stop Object Detection", key="detection_control")
43:
44:     # Open video capture if checkbox is active
45:     if run_detection:
46:         cap = cv2.VideoCapture(0)
47:         st_frame = st.empty() # Placeholder for video frames
48:         st_detection_info = st.empty() # Placeholder for detection information
49:
50:         while True:
51:             ret, frame = cap.read()
52:             if not ret:
53:                 st.warning("Failed to capture image.")
54:                 break
55:
56:             # Run YOLO detection
57:             frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert to RGB for display
58:             results = model.predict(frame, imgsz=640) # Perform detection
59:
60:             # Draw results and collect detected objects
61:
62:             # Display video feed
63:             st_frame.image(frame, channels="RGB", use_column_width=True)
64:
65:             # Display detection information
66:             if detected_objects:
67:                 object_counts = Counter(detected_objects)
68:                 detection_info = "\n".join([f"{obj}: {count}" for obj, count in object_counts.items()])
69:             else:
70:                 detection_info = "No objects detected."
71:
72:             st_detection_info.text(detection_info) # Update detection info text
73:
74:             # Break the loop if checkbox is unchecked
75:             if not st.session_state.detection_control:
76:                 break
77:
78:             cap.release()
79:
80: if __name__ == "__main__":
81:     main()
82:
83:
84:
```

0: 480x640 1 person, 69.4ms
Speed: 2.0ms preprocess, 69.4ms inference, 109.8ms postprocess per image at shape (1, 3, 480, 640)
2024-12-24 17:29:00.898 Examining the path of torch.classes raised: tried to instantiate class '_path__path', but it does not exist! Ensure that it is registered via torch.class.
2024-12-24 17:29:00.968 MediaFileHandler: Missing file 64ead2debeb0c93aa96d01095ab456ee6311b2b5d648004b497fa1bb.jpg

```
60:             # Draw results and collect detected objects
61:             frame, detected_objects = display_results(frame, results[0])
62:
63:             # Display video feed
64:             st_frame.image(frame, channels="RGB", use_column_width=True)
65:
66:             # Display detection information
67:             if detected_objects:
68:                 object_counts = Counter(detected_objects)
69:                 detection_info = "\n".join([f"{obj}: {count}" for obj, count in object_counts.items()])
70:             else:
71:                 detection_info = "No objects detected."
72:
73:             st_detection_info.text(detection_info) # Update detection info text
74:
75:             # Break the loop if checkbox is unchecked
76:             if not st.session_state.detection_control:
77:                 break
78:
79:             cap.release()
80:
81: if __name__ == "__main__":
82:     main()
83:
84:
```

0: 480x640 1 person, 69.4ms
Speed: 2.0ms preprocess, 69.4ms inference, 109.8ms postprocess per image at shape (1, 3, 480, 640)
2024-12-24 17:29:00.898 Examining the path of torch.classes raised: tried to instantiate class '_path__path', but it does not exist! Ensure that it is registered via torch.class.
2024-12-24 17:29:00.968 MediaFileHandler: Missing file 64ead2debeb0c93aa96d01095ab456ee6311b2b5d648004b497fa1bb.jpg

3. Selanjutnya

Buka Command Promp dan masukkan variabel folder yang sudah di buat, lalu masukkan virtualment dengan perintah .venv\scripts\activate


```
Microsoft Windows [Version 10.0.22631.4541]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>cd ..
C:\Users>cd ..
C:\>cd UAS Kecerdasan
C:\UAS Kecerdasan>.env\scripts\activate
(.venv) C:\UAS Kecerdasan>streamlit run app.py

You can now view your Streamlit app in your browser.

Local URL: http://localhost:8501
Network URL: http://172.20.10.4:8501

2024-12-24 19:56:32.125 Examining the path of torch.classes raised: Tried to instantiate class '._path_._path', but it
does not exist! Ensure that it is registered via torch::class_

0: 480x640 1 person, 122.2ms
Speed: 4.5ms preprocess, 122.2ms inference, 7.5ms postprocess per image at shape (1, 3, 480, 640)
2024-12-24 19:56:39.286 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Pl
ease utilize the 'use_container_width' parameter instead.

0: 480x640 1 person, 119.0ms
Speed: 2.8ms preprocess, 119.0ms inference, 1.0ms postprocess per image at shape (1, 3, 480, 640)
2024-12-24 19:56:39.382 The 'use_column_width' parameter has been deprecated and will be removed in a future release. Pl
ease utilize the 'use_container_width' parameter instead.

object_counts = Counter(detected_objects)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SEARCH ERROR
PS C:\UAS Kecerdasan>
```

Lalu kita klik Enter , dan Centangkan bagian Start/Stop Object Detecion

