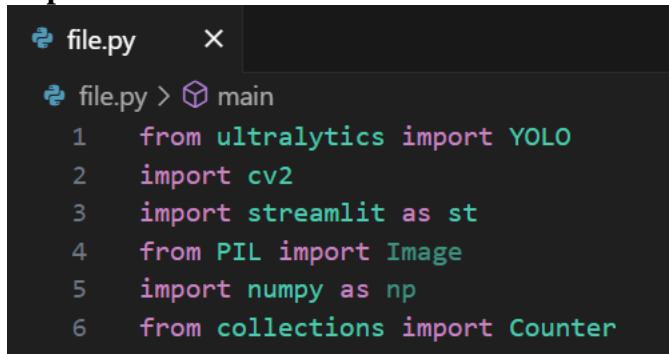


NAMA : Artha Akbarrullah Sawiran
NIM : 230741109
PRODI : Ilmu Komputer
MAKUL: Kecerdasan Buatan

PENJELASAN KODE DAN TUTORIAL MENJALANKAN APLIKASI

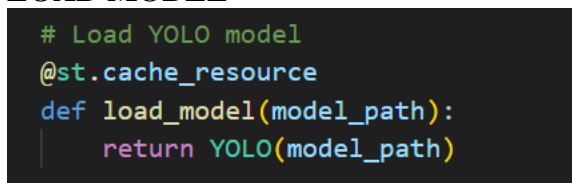
Import Modul



```
file.py  X
file.py > main
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
```

- ultralytics: Digunakan untuk memuat model YOLO.
- cv2: Untuk manipulasi gambar dan video.
- streamlit: Untuk membangun antarmuka aplikasi web.
- PIL.Image dan numpy: Untuk pengolahan gambar.
- collections.Counter: Untuk menghitung jumlah objek yang terdeteksi.

LOAD MODEL



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

Memuat model YOLO menggunakan pustaka ultralytics.

Model hanya dimuat sekali menggunakan dekorator `@st.cache_resource`, sehingga aplikasi tidak perlu memuat ulang model setiap kali dijalankan, mempercepat proses.

- a. `display_results`

```

# Process and display the detection results
def display_results(image, results):
    boxes = results.boxes.xyxy.cpu().numpy() # [x1, y1, x2, y2]
    scores = results.boxes.conf.cpu().numpy() # Confidence scores
    labels = results.boxes.cls.cpu().numpy() # Class indices
    names = results.names # Class names

    detected_objects = []

    for i in range(len(boxes)):
        if scores[i] > 0.5: # Confidence threshold
            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)
            cv2.putText(image, f"{label}: {score:.2f}", (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)

    return image, detected_objects

```

Fungsi utama: Menggambar kotak batas dan label pada frame gambar/video berdasarkan hasil prediksi YOLO.

Jika confidence score di atas 0.5, bounding box akan digambar pada gambar dengan label objek.

b. Fungsi Utama: main

```

file.py > main
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             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()

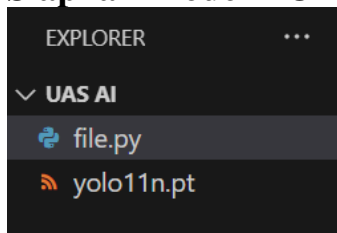
if __name__ == "__main__":
    main()

```

- Membuat antarmuka Streamlit dengan judul dan sidebar untuk mengontrol deteksi.
- Menggunakan kamera untuk menangkap video dan menjalankan model YOLO untuk mendeteksi objek secara real-time.
- Menampilkan video feed dengan bounding box dan informasi deteksi pada halaman web Streamlit.

MENJALANKAN APLIKASI

Siapkan Model YOLO:



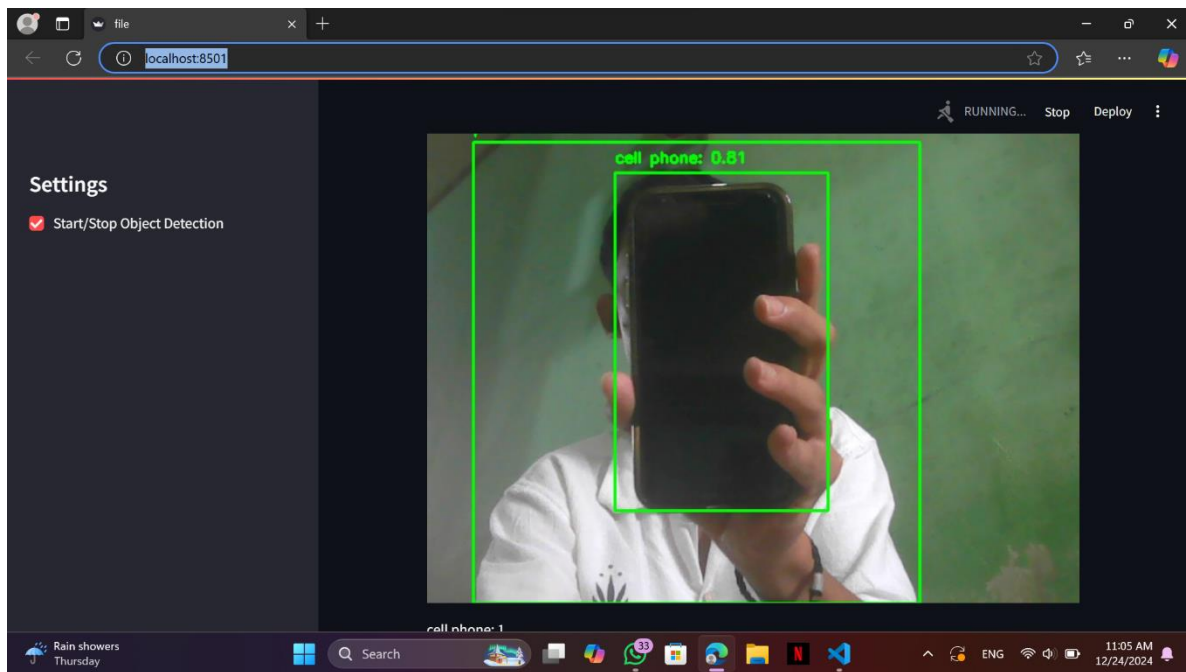
- Unduh file model YOLO (yolo11n.pt).
- Tempatkan file model di folder yang sama dengan file file.py.

MENJALANKAN APLIKASI

Jalankan Aplikasi:

- Gunakan perintah berikut di terminal:
streamlit run file.py
- Streamlit akan membuka browser dan memuat antarmuka aplikasi.

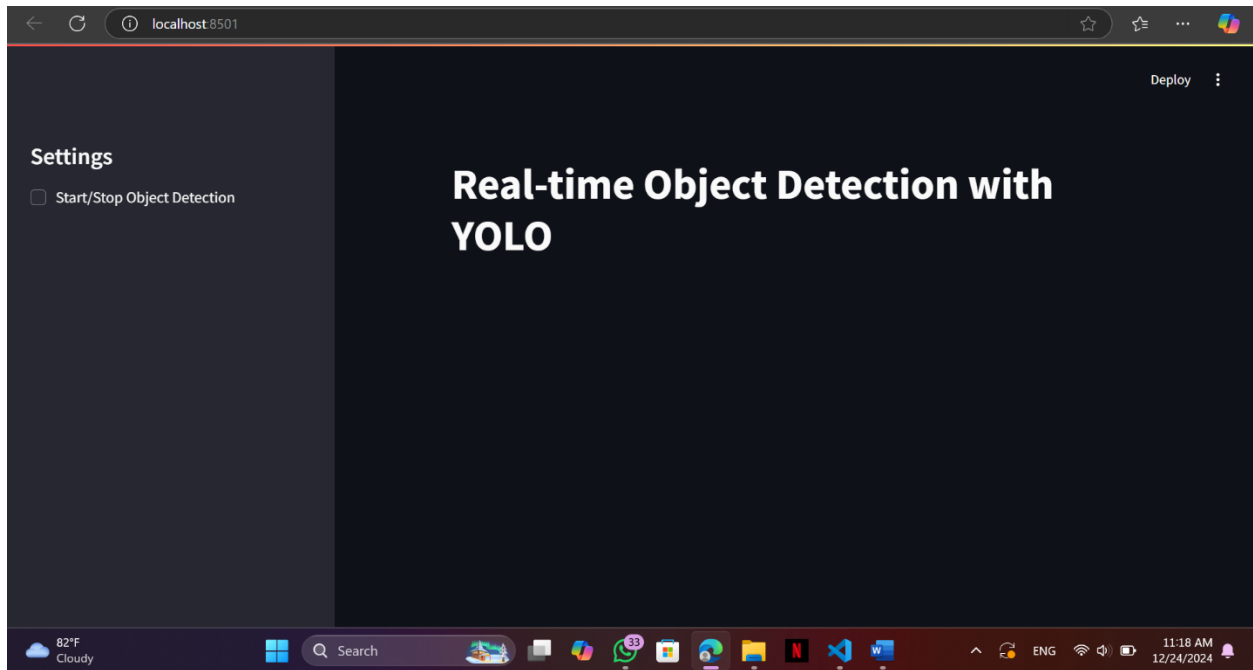
Menggunakan Aplikasi:



- Centang checkbox **Start/Stop Object Detection** di sidebar untuk memulai deteksi.
- Aplikasi akan:
 - Menampilkan video dari kamera.
 - Menggambar kotak batas pada objek yang terdeteksi.
 - Menampilkan jumlah dan jenis objek di sidebar.

Hentikan Aplikasi:

- Klik ulang checkbox untuk menghentikan deteksi.
- Tekan `Ctrl+C` di terminal untuk keluar sepenuhnya dari aplikasi.



DEPLOY

1.BUKA CMD DAN MASUKKAN FOLDER YANG SUDAH DI BUAT

2.LALU MASUKAN VIRTUALMENT DENGAN PRINTAH (VENV\SCRIPTS\ACTIVATE)

3.LALU COBA DENGAN CODINGAN (STREAMLIT RUN ("NAMA FILE".PY)

JIKA SUDAH MUNCUL SEPerti ini :

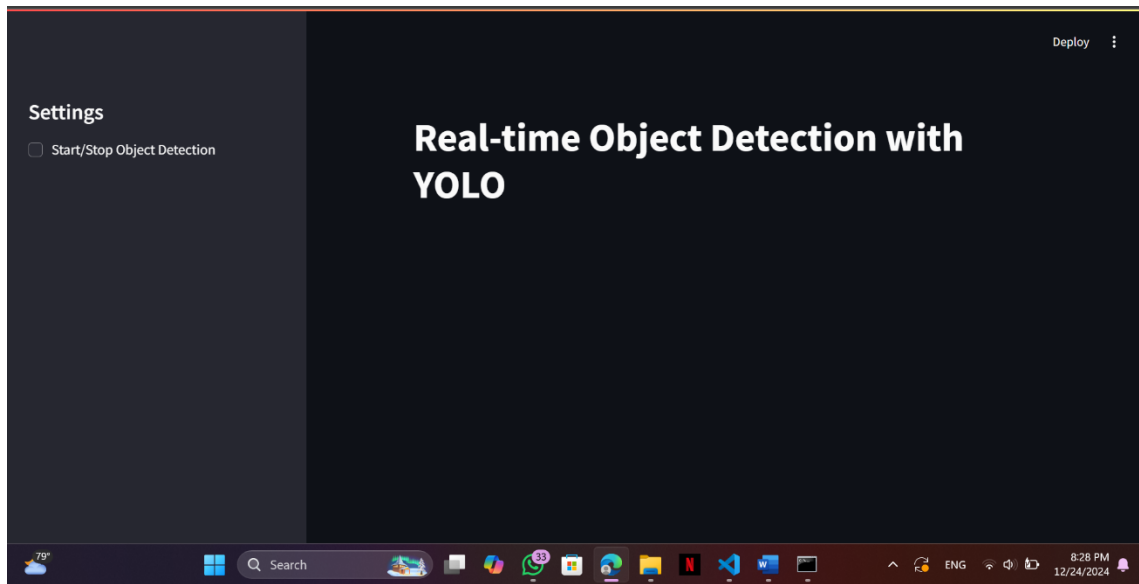
Local URL: <http://localhost:8502>

Network URL: <http://192.168.1.20:8502>

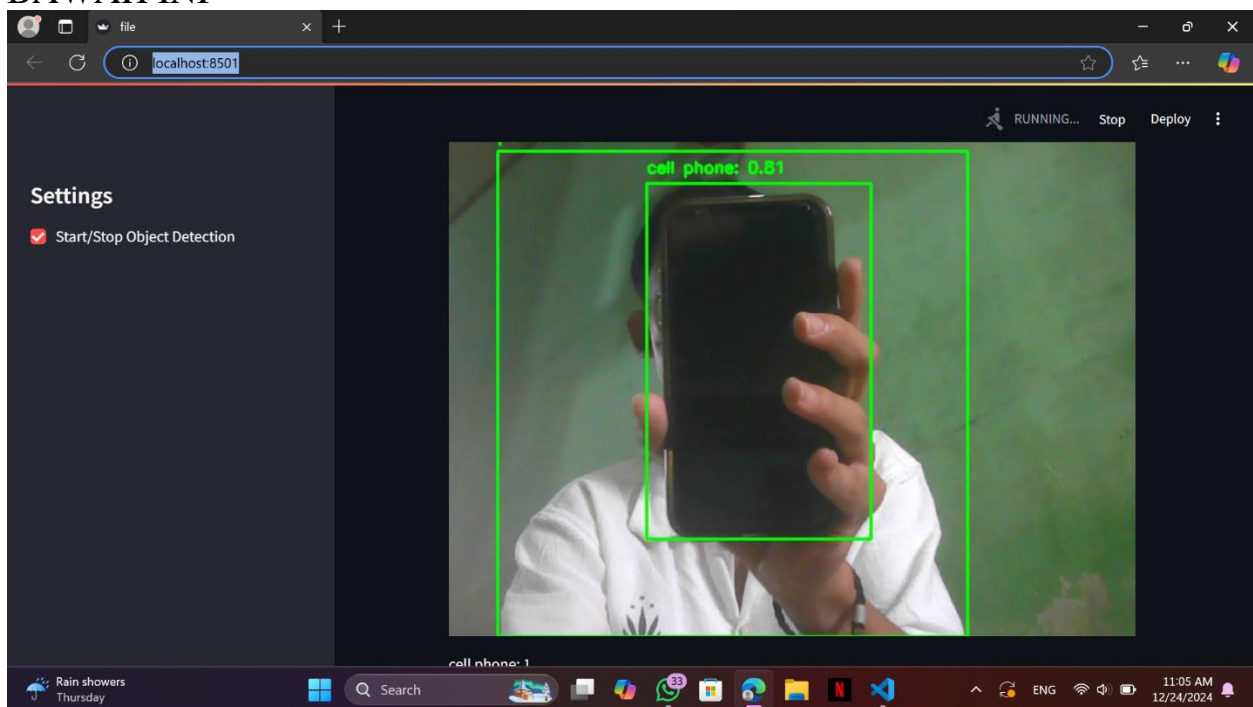
DENGAN PERINTAH:

<http://localhost:8502>

NANTI AKAN ADA GAMBAR YANG MUNCUL SEPerti BERIKUT INI:



JIKA INGIN MENDETEKSI BAGIAN REAL-TIME TINGGAL TEKAN AJA LALU CENTANGKAN MULAI DETEKSI MAKA AKAN MUNCUL GAMBAR DETEKSI OBJEK DENGAN KAMERA SEPERTI CONTOH DI BAWAH INI



SETELAH ITU AKAN MUNCUL KAMERA UNTUK MENDETEKSI WAJAH/HANDPHONE SEPERTI CONTOH DI ATAS