

## PENJELASAN CODE DAN SS HASIL

NAMA : AGUNGMAULANA

NIM : 230741104

PRODI/ KELAS : ILKOM / 3A

MATA KULIAH : KECERDASAN TIRUAN

DOSEN : ZIKRI WAHYUZI, M.Pd

### CODE

```
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  import base64 # Untuk encoding gambar ke base64
8  import pygame
9  import random
0
1  st.markdown("""
2  <style>
3  .navbar {
4      background-color: #303b32; /* Warna latar belakang */
5      padding: 20px; /* Memberikan lebih banyak ruang di dalam navbar */
6      text-align: center; /* Pusatkan elemen di dalam navbar */
7      font-size: 18px; /* Ukuran font navbar */
8      height: 60px; /* Tinggi navbar */
9      line-height: 20px; /* Untuk mengatur jarak vertikal teks */
0  }
1  .navbar a {
2      color: white; /* Warna teks */
3      text-decoration: none; /* Hapus garis bawah pada teks */
4      padding: 15px; /* Tambahkan ruang di sekitar teks */
5      font-weight: bold; /* Membuat teks lebih tebal */
6      font-size: 36px; /* Ukuran font tautan */
7  }
8  .navbar a:hover {
9      background-color: #008000; /* Warna latar saat hover */
0      border-radius: 5px; /* Sudut membulat */
1  }
2  .sinyal-container {
3      position: fixed;
4      bottom: 20px;
5      left: 20px;
6      background-color: z#303b32; /* Warna latar belakang sinyal */
7      color: white;
8      padding: 10px;
9      border-radius: 8px;
0      font-size: 14px;
1      font-weight: bold;
2      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2); /* Tambahkan efek bayangan */
3      z-index: 1000;
4      display: flex;
5      align-items: center;

```

```

}
.sinyal-container img {
    width: 24px;
    height: 24px;
    margin-right: 8px;
}
</style>
<div class="navbar">
    <a href="#section1">PEROJEK UAS AGUNG MAULANA</a>
</div>
<div class="sinyal-container">
    
    JEKPOTT BROO
</div>
"""", unsafe_allow_html=True)

# Fungsi untuk menambahkan latar belakang
def set_background(image_path, size="cover"):
    with open(image_path, "rb") as file:
        base64_image = base64.b64encode(file.read()).decode()

    css = f"""
<style>
.stApp {{
    background-image: url("data:image;base64,{base64_image}");
    background-size: {size}; /* Menentukan ukuran gambar latar belakang */
    background-repeat: no-repeat;
    background-attachment: fixed;
}}
</style>
"""

    st.markdown(css, unsafe_allow_html=True)

# Fungsi untuk menampilkan GIF di sidebar dengan bentuk lingkaran dan digeser atas-bawah
def display_animation(gif_path):
    with open(gif_path, "rb") as file:
        gif_data = file.read()
    base64_gif = base64.b64encode(gif_data).decode()
    st.sidebar.markdown(
        f"""
<div style="text-align: center; height: 250px; overflow-y: auto;">
    
</div>
"""
    )

```

```
89         </div>
90         """
91     unsafe_allow_html=True
92 )
93
94
95 # Load YOLO model
96 @st.cache_resource
97 def load_model(model_path):
98     return YOLO(model_path)
99
100 # Process and display the detection results
101 def display_results(image, results):
102     boxes = results.boxes.xyxy.cpu().numpy() # [x1, y1, x2, y2]
103     scores = results.boxes.conf.cpu().numpy() # Confidence scores
104     labels = results.boxes.cls.cpu().numpy() # Class indices
105     names = results.names # Class names
106
107     detected_objects = []
108
109     for i in range(len(boxes)):
110         if scores[i] > 0.5: # Confidence threshold
111             x1, y1, x2, y2 = boxes[i].astype(int)
112             label = names[int(labels[i])]
113             score = scores[i]
114             detected_objects.append(label)
115             cv2.rectangle(image, (x1, y1), (x2, y2), (0, 255, 0), 2)
116             cv2.putText(image, f'{label}: {score:.2f}', (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
117
118     return image, detected_objects
119
120 # Main Streamlit app
121 def main():
122     # Menambahkan CSS untuk kotak pada sidebar
123     st.markdown("""
124     <style>
125     .sidebar-title {
126         background-color: #70ff8f;
127         padding: 6px;
128         border-radius: 6px;
129         font-size: 16px;
130         font-weight: bold;
131         text-align: center;
132     }
133     </style>
134     """)
135
136     sidebar = st.sidebar
137     sidebar.title("Sidebar Title")
```



```

def main():
    }
    </style>
    """ , unsafe_allow_html=True)

    # Tampilkan animasi di sidebar
    display_animation("planets-universe.gif") # Path ke file GIF

    # Menampilkan judul dalam kotak
    st.sidebar.markdown('<div class="sidebar-title">MENU</div>', unsafe_allow_html=True)
    st.markdown(
        '<h1 style="color: white;">HASIL DETEKSI</h1>',
        unsafe_allow_html=True
    )

    # Atur background
    set_background("download (1).jpeg", size="100% 100%") # Path ke gambar latar belakang

    model_path = "yolo1in.pt" # Path to your YOLO model
    model = load_model(model_path)

    # Gunakan radio button sebagai kontrol
    detection_status = st.sidebar.radio(
        "",
        options=["🔴 LOM MENJADI LAAH", "🟢 BARU IDUP ALHAMDULILAH"],
        index=0,
        key="detection_control"
    )

    # Jalankan deteksi hanya jika radio button diatur ke "Aktif"
    if detection_status == "🟢 BARU IDUP ALHAMDULILAH":
        cap = cv2.VideoCapture(0)
        st_frame = st.empty() # Placeholder untuk tampilan video
        st_detection_info = st.empty() # Placeholder untuk informasi deteksi

        while True:
            ret, frame = cap.read()
            if not ret:
                st.warning("Gagal menangkap gambar dari kamera.")
                break

            # Jalankan deteksi dengan YOLO
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert ke RGB
            results = model.predict(frame, imgsz=460) # Lakukan prediksi

            # Jalankan deteksi dengan YOLO
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB) # Convert ke RGB
            results = model.predict(frame, imgsz=460) # Lakukan prediksi

            # Tampilkan hasil dan daftar objek yang terdeteksi
            frame, detected_objects = display_results(frame, results[0])

            # Tampilkan video
            st_frame.image(frame, channels="RGB", use_column_width=True)

            # Tampilkan informasi objek yang terdeteksi
            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 = "Tidak ada objek terdeteksi."

            st_detection_info.text(detection_info) # Perbarui teks informasi deteksi

        cap.release()

    if __name__ == "__main__":
        main()

```





## Penjelasan Tentang Masing Masing Code

### 1. from ultralytics import YOLO

Mengimpor modul YOLO dari library Ultralytics, yang menyediakan implementasi dari algoritma YOLO (You Only Look Once) untuk deteksi objek.

Digunakan untuk memanfaatkan model YOLO untuk keperluan seperti deteksi objek atau pelacakan.

### 2. import cv2

Mengimpor library OpenCV (dengan nama modul cv2), yang digunakan untuk pengolahan gambar dan video.

Fungsi umum: membaca/menulis file gambar, melakukan deteksi objek, face detection, transformasi gambar, dan sebagainya.

### 3. import streamlit as st

Mengimpor Streamlit, library Python untuk membangun aplikasi web interaktif, terutama untuk visualisasi data atau model machine learning.

Fungsi utamanya dalam kode ini adalah memungkinkan pengguna membuat antarmuka pengguna (UI) untuk aplikasi berbasis Python.

### 4. from PIL import Image

Mengimpor modul Image dari Pillow (PIL), sebuah library Python untuk manipulasi gambar.

Biasanya digunakan untuk membaca gambar, mengubah format gambar, atau memodifikasi gambar seperti resizing, cropping, atau filtering.

### 5. import numpy as np

Mengimpor NumPy, library Python untuk operasi numerik, dengan nama alias np.

Fungsi umum: membuat dan memanipulasi array multidimensi (misalnya, representasi gambar), melakukan operasi matematis, atau mengoptimalkan perhitungan numerik.

### 6. from collections import Counter

Mengimpor Counter dari modul built-in collections di Python.

Counter digunakan untuk menghitung frekuensi elemen dalam sebuah iterable, seperti daftar atau string.

#### 7. import base64

Mengimpor modul base64, yang digunakan untuk encoding dan decoding data dalam format Base64.

Format Base64 sering digunakan untuk mengencode data biner (seperti gambar) menjadi teks ASCII agar dapat dikirimkan atau disimpan dengan aman.

#### 8. import pygame

Mengimpor pygame, library Python untuk membuat game atau simulasi multimedia.

pygame mendukung fitur seperti rendering grafik, pemutaran suara, dan event handling (keyboard/mouse).

#### 9. import random

Mengimpor modul random, library bawaan Python untuk menghasilkan angka atau data secara acak.

Digunakan untuk berbagai fungsi seperti memilih elemen acak dari daftar, mengacak urutan, atau menghasilkan angka dalam rentang tertentu.

Sebelum lanjut ke penjelasan selanjutnya disini kalian perlu install modul yang belum disiapkan dari python seperti :

pip install ultralytics

pip install opencv-python

pip install streamlit

pip install Pillow

pip install numpy

pip install pygame

#### 10. Membuat Navbar dan sinyal

```
st.markdown("""
```

```
<style>
```

```
.navbar {
```

```
background-color: #303b32;  
/* Warna latar belakang */  
padding: 20px; /*  
Memberikan lebih banyak ruang  
di dalam navbar */  
text-align: center; /* Pusatkan  
elemen di dalam navbar */  
font-size: 18px; /* Ukuran  
font navbar */  
height: 60px; /* Tinggi  
navbar */  
line-height: 20px; /* Untuk  
mengatur jarak vertikal teks */  
}  
.navbar a {  
color: white; /* Warna  
teks */  
text-decoration: none; /*  
Hapus garis bawah pada teks  
*/  
padding: 15px; /*  
Tambahkan ruang di sekitar  
teks */  
font-weight: bold; /*  
Membuat teks lebih tebal */  
font-size: 36px; /* Ukuran  
font tautan */  
}  
.navbar a:hover {  
background-color:
```

```
#008000; /* Warna latar saat
hover */
border-radius: 5px; /*
Sudut membulat */
}

.sinyal-container {
    position: fixed;
    bottom: 20px;
    left: 20px;
    background-color:
z#303b32; /* Warna latar
belakang sinyal */
    color: white;
    padding: 10px;
    border-radius: 8px;
    font-size: 14px;
    font-weight: bold;
    box-shadow: 0 4px 8px
rgba(0, 0, 0, 0.2); /*
Tambahkan efek bayangan
*/
    z-index: 1000;
    display: flex;
    align-items: center;
}

.sinyal-container img {
    width: 24px;
    height: 24px;
    margin-right: 8px;
}
```

```

</style>

<div class="navbar">
    <a
        href="#section1">PEROJE
        K UAS AGUNG
        MAULANA</a>
    </div>
    <div class="sinyal-
    container">
        
        JEKPOTT BROO
    </div>
    """,
    unsafe_allow_html=True)

```

- **Tujuan:** Menambahkan navigasi dan indikator sinyal di halaman Streamlit.
- **Navbar:** Dihias dengan warna latar belakang gelap, teks berwarna putih, dan link yang berwarna hijau saat di-hover.
- **Sinyal:** Bagian bawah kiri menampilkan indikator sinyal dengan gambar ikon sinyal dan pesan "BARU IDUP ALHAMDULILAH".

Fungsi untuk Menambahkan Latar Belakang Gambar

```

def set_background(image_path, size="cover"):
    with open(image_path, "rb") as file:
        base64_image =
            base64.b64encode(file.read()).decode()

```

```

css = f"""
<style>
.stApp {{
    background-image:
url("data:image;base64,{base64_image}");
    background-size: {size}; /* Menentukan
ukuran gambar latar belakang */
    background-repeat: no-repeat;
    background-attachment: fixed;
}}
</style>
"""

```

```
st.markdown(css, unsafe_allow_html=True)
```

```
st.markdown(css, unsafe_allow_html=True)
```

- **Tujuan:** Menambahkan gambar latar belakang pada aplikasi Streamlit.
- **Proses:** Gambar yang diberikan (`image_path`) akan dibaca dan diubah menjadi format Base64. Kemudian, gambar tersebut digunakan sebagai latar belakang aplikasi dengan properti CSS.

## 11. Fungsi untuk Menampilkan GIF di Sidebar dengan Animasi

```

def display_animation(gif_path):
    with open(gif_path, "rb") as file:
        gif_data = file.read()
    base64_gif = base64.b64encode(gif_data).decode()
    st.sidebar.markdown(
f"""
<div style="text-align: center; height: 250px; overflow-y: auto;">
    >

</div>

"",

```
unsafe_allow_html=True  
)  
• Tujuan: Menampilkan animasi GIF di sidebar dalam bentuk lingkaran.  
• Proses: GIF yang diberikan dibaca, dikodekan ke Base64, dan kemudian ditampilkan di sidebar dengan ukuran tertentu dan efek lingkaran.
```

## 12. Memuat Model YOLO

```
@st.cache_resource  
  
def load_model(model_path):  
    return YOLO(model_path)  
  
• Tujuan: Memuat model YOLO yang digunakan untuk deteksi objek.  
• Proses: Fungsi ini akan memuat model YOLO dari file yang diberikan dan menyimpannya dalam cache agar tidak perlu memuat ulang setiap kali aplikasi dijalankan.
```

## 13. Fungsi untuk Menampilkan Hasil Deteksi

```
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 - 10,
```

```
cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
```

```
    return image, detected_objects
```

- **Tujuan:** Menampilkan hasil deteksi objek pada gambar.
- **Proses:**

- Mengambil kotak deteksi, skor kepercayaan, dan label objek dari hasil prediksi model YOLO.
- Jika skor kepercayaan lebih dari 0.5, gambar objek yang terdeteksi dengan kotak hijau dan teks label dengan skor.
- Mengembalikan gambar yang telah diberi anotasi dan daftar objek yang terdeteksi.

#### 14. Fungsi Utama Streamlit (Main Application)

```
# Main Streamlit app
```

```
def main():
```

```
    # Menambahkan CSS untuk kotak pada
```

```
    sidebar
```

```
    st.markdown("""
```

```
<style>
```

```
.sidebar-title {
```

```
    background-color: #70ff8f;
```

```
    padding: 6px;
```

```
    border-radius: 6px;
```

```
    font-size: 16px;
```

```
    font-weight: bold;
```

```
    text-align: center;
```

```
}
```

```
</style>
```

```
"""", unsafe_allow_html=True)
```

```
# Tampilkan animasi di sidebar
```

```
display_animation("planets-universe.gif") #
```

```
Path ke file GIF
```

```
# Menampilkan judul dalam kotak
st.sidebar.markdown('<div class="sidebar-
title">MENU</div>',
unsafe_allow_html=True)

st.markdown(
    '<h1 style="color: white;">HASIL
DETEKSI</h1>',
unsafe_allow_html=True
)
```

```
# Atur background
set_background("download (1).jpeg",
size="100% 100%") # Path ke gambar latar
belakang

model_path = "yolo11n.pt" # Path to your
YOLO model
model = load_model(model_path)
```

```
# Gunakan radio button sebagai kontrol
detection_status = st.sidebar.radio(
    "",

    options=["
        LOM MENJADI LAAH",
    "
        BARU IDUP ALHAMDULILAH"],
    index=0,
    key="detection_control"
)
```

```
# Jalankan deteksi hanya jika radio button
diatur ke "Aktif"
```

```
if detection_status == " BARU IDUP  
ALHAMDULILAH":  
    cap = cv2.VideoCapture(0)  
    st_frame = st.empty() # Placeholder  
    untuk tampilan video  
    st_detection_info = st.empty() #  
    Placeholder untuk informasi deteksi  
  
    while True:  
        ret, frame = cap.read()  
        if not ret:  
            st.warning("Gagal menangkap  
gambar dari kamera.")  
            break  
  
        # Jalankan deteksi dengan YOLO  
        frame = cv2.cvtColor(frame,  
cv2.COLOR_BGR2RGB) # Convert ke RGB  
        results = model.predict(frame,  
imgsz=460) # Lakukan prediksi  
  
        # Tampilkan hasil dan daftar objek  
        yang terdeteksi  
        frame, detected_objects =  
display_results(frame, results[0])  
  
        # Tampilkan video  
        st_frame.image(frame,  
channels="RGB", use_column_width=True)  
  
        # Tampilkan informasi objek yang
```

terdeteksi

```
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 = "Tidak ada objek  
terdeteksi."
```

```
st_detection_info.text(detection_info)  
# Perbarui teks informasi deteksi
```

```
cap.release()
```

- **Tujuan:** Menampilkan aplikasi utama untuk deteksi objek menggunakan YOLO.
- **Proses:**
  - Menambahkan elemen-elemen visual seperti animasi di sidebar dan latar belakang.
  - Menggunakan radio button untuk mengontrol status deteksi.

- Jika status deteksi aktif, aplikasi akan menangkap video dari webcam, menjalankan model YOLO, dan menampilkan hasil deteksi secara langsung.

## 15. Eksekusi Aplikasi

```
if __name__ == "__main__":  
    main()
```

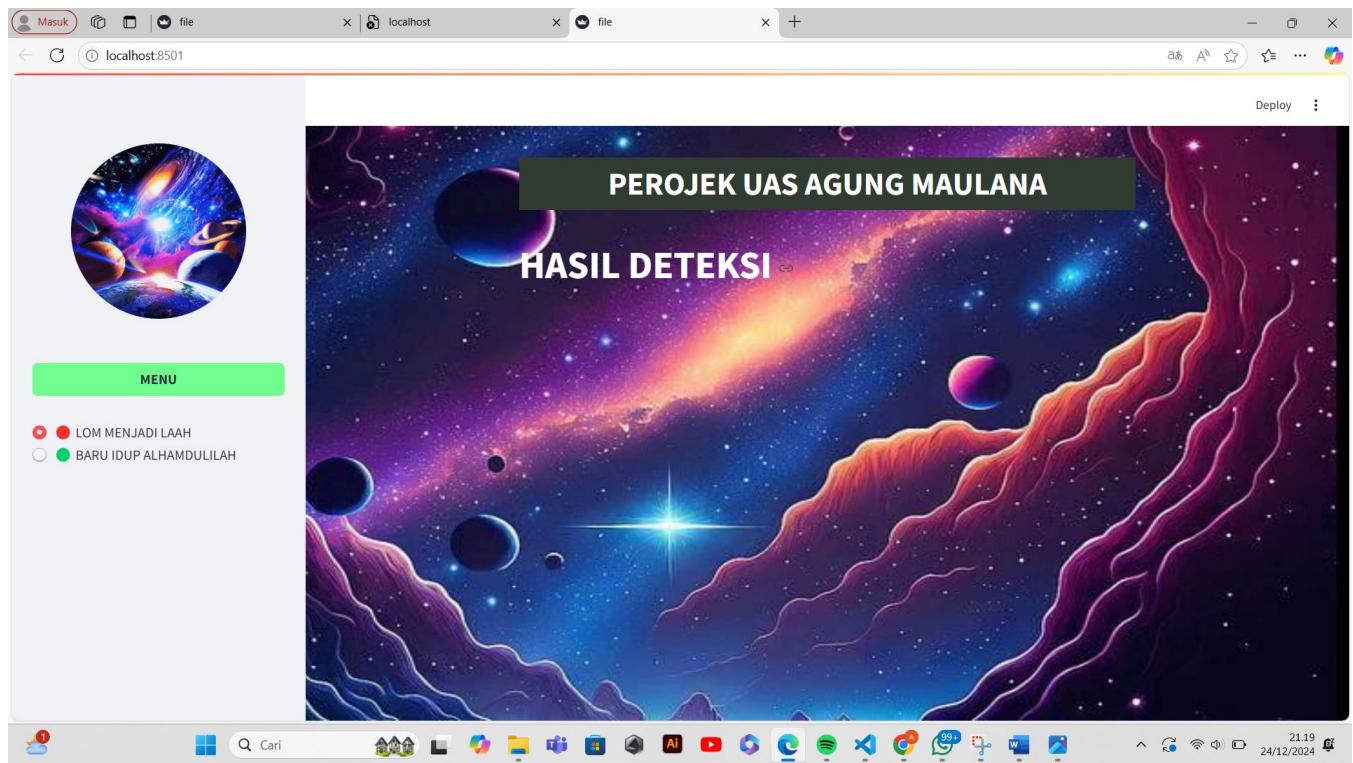
- **Tujuan:** Menjalankan fungsi utama main() saat file ini dijalankan.

## SS HASIL

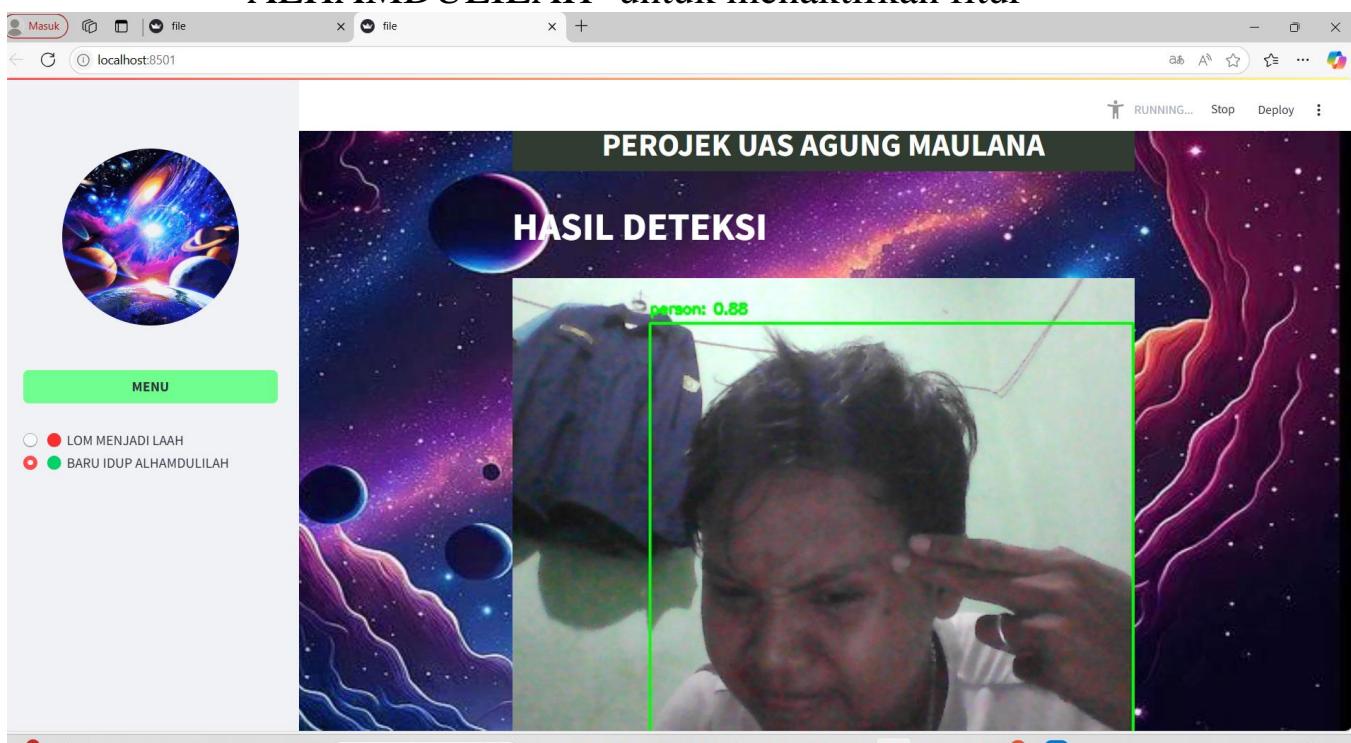
```
File Edit Selection View Go Run Terminal Help ↻ → 🔍 uas
EXPLORER .venv uas
  .venv
  download (1).jpeg
  file.py
  planets-universe.gif
  yolov5npt
  ...
  file.py x planets-universe.gif
  file.py > main
  63 def set_background(image_path, size="cover"):
  64     background-size: (size); /* Menentukan ukuran gambar latar belakang */
  65     background-repeat: no-repeat;
  66     background-attachment: fixed;
  67   }
  68   </style>
  69   """
  70
  71   st.markdown(css, unsafe_allow_html=True)
  72
  73   # Fungsi untuk menampilkan GIF di sidebar dengan bentuk lingkaran dan digeser atas-bawah
  74   def display_animation(gif_path):
  75       with open(gif_path, "rb") as file:
  76           gif_data = file.read()
  77           base64_gif = base64.b64encode(gif_data).decode()
  78           st.sidebar.markdown(
  79               f"""
  80               <div style="text-align: center; height: 250px; overflow-y: auto;">
  81                   
  82               </div>
  83               """
  84           )
  85           unsafe_allow_html=True
  86       )
  87
  88   # Load YOLO model
  89   @st.cache_resource
  90   ...
  91
  92   ...
  93
  94
  95   # Load YOLO model
  96   @st.cache_resource
  97   ...
  98
  99
  PROBLEMS OUTPUT TERMINAL PORTS
  PS D:\uas> streamlit run file.py
  powershell +v 🌐 ...
```

Ln 191, Col 22 (2572 selected) Spaces: 4 UTF-8 CRLF {} Python 3.12.8 (.venv: venv)

- 1) Nah kalau kalian sudah menerapkan code diatas tadi, kalian tinggal run, dengan cara :
- 2) Streamlit run app.py
- 3) Nanti kalian ada diminta masukkan gmail
- 4) Nanti tinggal masukkin saja



- 5) Nah nanti bakalan langsung dipindahkan ke web
- 6) Disini ada 2 opsi pencet “**LOM MENJADI LAAH**” untuk mematikan fitur, dan pencet “**BARU IDUP ALHAMDULILAH**” untuk menaktifkan fitur



- 7) Nah ini hasil fitur tadi. Tunggu apalagi ayok coba coding ini

