

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Hasil pengujian menunjukkan semua hardware dan software bekerja dengan baik dan sesuai dengan fungsi yang dibutuhkan.

Berikut adalah jawaban untuk pertanyaan dalam rumusan masalah:

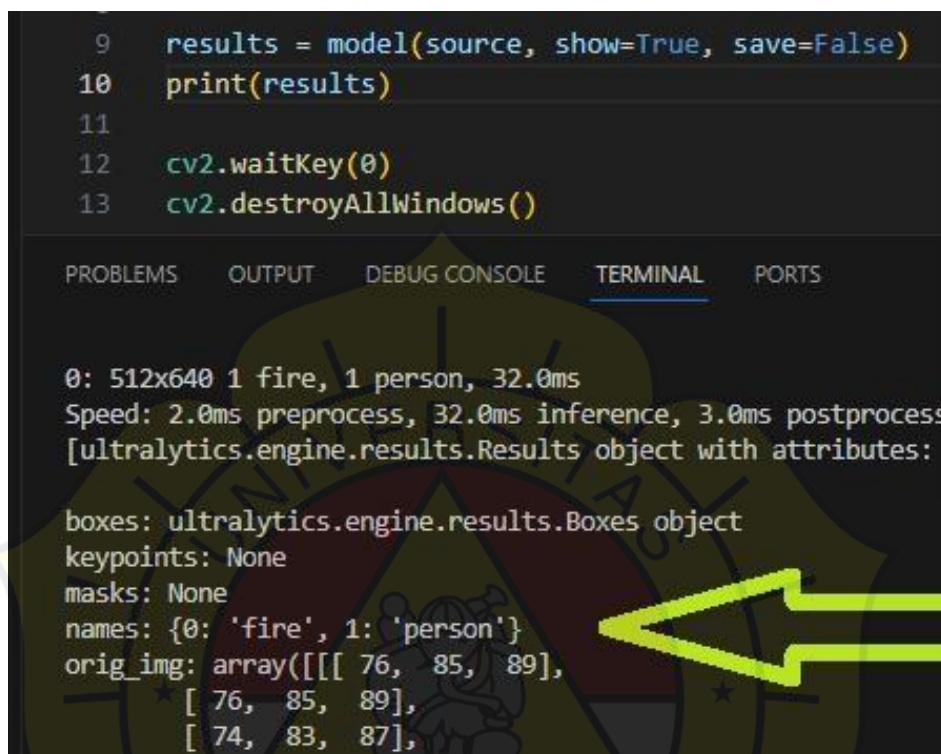
1. Cara agar sistem dapat mendeteksi kebakaran berdasarkan image api kamera adalah dengan mendeteksi nomor *index* pada kelas objek api tersebut, yaitu 0.
2. Untuk sistem dapat menghitung jumlah orang yang terdeteksi, hal yang sama juga dilakukan dengan nomor *index* pada kelas objek orang yaitu 1. Yang hanya perlu dilakukan adalah menghitung jumlah *index* 1 yang ditemukan dalam sebuah list setiap kali *frame* kamera dibaca.
3. Agar petugas sekolah dapat cepat mengetahui lokasi kebakaran, sistem akan memberikan notifikasi kalimat berdasarkan lokasi ditemukannya api dalam kamera, sesuai dengan nomor 1.

Untuk mengetahui indeks mana yang harus dideteksi, lakukan `'print (results)'` agar *YOLOv8* dapat menampilkan nama dari masing-masing *index* yang ditemukan di dalam model yang dilatih.

```
9 results = model(source, show=True, save=False)
10 print(results)
11
12 cv2.waitKey(0)
13 cv2.destroyAllWindows()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
0: 512x640 1 fire, 1 person, 32.0ms
Speed: 2.0ms preprocess, 32.0ms inference, 3.0ms postprocess
[ultralytics.engine.results.Results object with attributes:
boxes: ultralytics.engine.results.Boxes object
keypoints: None
masks: None
names: {0: 'fire', 1: 'person'}
orig_img: array([[ 76,  85,  89],
                 [ 76,  85,  89],
                 [ 74,  83,  87],
```



Gambar 5.1. Deteksi Api dan Orang 1

Berdasarkan Gambar 5.1 diatas, dapat dilihat kelas api memiliki *index* 0 dan kelas orang memiliki *index* 1. Setelah mengetahui hal tersebut, yang perlu dilakukan hanyalah Penelitian sintaks kondisi apakah *index* api dan indeks orang ditemukan. Jika benar maka program hanya perlu merangkai kalimat yang bisa dibaca oleh Pengguna untuk dapat dimasukkan ke dalam *database* dan di konversikan ke dalam *audio* mp3.

```

results = model(frame, conf=0.1, stream=True, save=False, show=False)
print(results)

for r in results:
    class_id = r.boxes.cls
    print(class_id)

    fire_found = 0 in class_id # memberi nama tipe data api
    person_found = 1 in class_id # memberi nama tipe data orang

    class_id_numpy = class_id.cpu().numpy() # konversi list class id tensor ke list numpy
    print(class_id_numpy)
    count_person = np.sum(class_id_numpy == 1) # hitung jumlah index 1 (orang) dalam list

    if fire_found:
        cv2.imwrite('saved_frame.jpg', frame)
        cap.release()
        cv2.destroyAllWindows()

        pixmap = QPixmap['saved_frame.jpg']
        videolabel.setPixmap(pixmap)
        videolabel.setScaledContents(True)

        fire_text = 'kebakaran di ' + input_sensor + '.'
        people_text = ' masih ada ' + str(count_person) + ' orang di dalam' if person_found else ''
        text = fire_text + people_text
        print(text)

```

Gambar 5.2. Deteksi Api dan Orang 2

Hasil kode di Gambar 5.2 atas yang nantinya akan menghasilkan teks “kebakaran di ruang 1 lantai 1. masih ada 1 orang di dalam.” jika output sensornya mengatakan “ruang 1 lantai 1”

```

language = 'id'
speech = gTTS(text=text, lang=language, slow=False)

speech.save("output.mp3")

with open('saved_frame.jpg', 'rb') as photo:
    bot.send_photo('-1002132512911', photo)
    bot.send_message('-1002132512911', bot_text)

for i in range(2):
    pygame.mixer.init()
    pygame.mixer.music.load("output.mp3")
    pygame.mixer.music.play()

    while pygame.mixer.music.get_busy():
        pygame.time.Clock().tick(10)

pygame.quit()

```

Gambar 5.3. Deteksi Api dan Orang 3

Teks pada Gambar 5.3 di atas yang nantinya akan dikonversi menggunakan library *gtts* (*google text to speech*) lalu diputar menggunakan modul *mixer* dari paket *pygame* sebanyak 2 kali.

5.2. Saran

- list tensor harus dikonversikan menjadi list numpy. Hal ini dilakukan karena list tensor tidak bisa dihitung secara langsung menggunakan sintaks `numpy.sum()`.
- Disarankan untuk tidak menghubungkan adapter 12v dengan router yang hanya mendukung 9v meskipun mereka memiliki ukuran adapter yang sama. hal ini bisa menimbulkan kabel yang terbakar.
- *Yolov8 Ultralytics* umumnya terinstall hanya untuk CPU. Jika ingin menggunakan GPU agar proses Deteksi berjalan lebih cepat, uninstall *torch* dan *torchvision* yang terinstall sebelumnya lalu lakukan pip install torch dari situs resmi pytorch seperti:

```
pip3 install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu117
```
- Hasil deteksi gambar tidak memiliki bounding box. Jika ingin menggunakan bounding box, bisa gunakan `annotated_frame = results[0].plot()` di bawah sintaks `results = model(source, ...)` lalu lakukan `cv2.imwrite('saved_frame.jpg', annotated_frame)` di dalam kondisi ditemukannya api.



TEKNOLOGI INFORMASI
UNIVERSITAS DARMA PERSADA