

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. HASIL IMPLEMENTASI

Pada tahap ini dilakukan implementasi sistem deteksi pelanggaran menggunakan model YOLOv8 yang telah dilatih sebelumnya untuk memantau kepatuhan pengunjung perpustakaan terhadap larangan membawa makanan dan minuman. Implementasi sistem dilakukan dengan mengintegrasikan beberapa komponen utama, yaitu laptop sebagai perangkat pemrosesan, webcam sebagai input citra, serta NodeMCU ESP8266 dan *Buzzer* sebagai perangkat notifikasi peringatan. Sistem ini dirancang agar dapat bekerja secara *real-time* dalam mendeteksi objek yang termasuk ke dalam kategori makanan dan minuman.



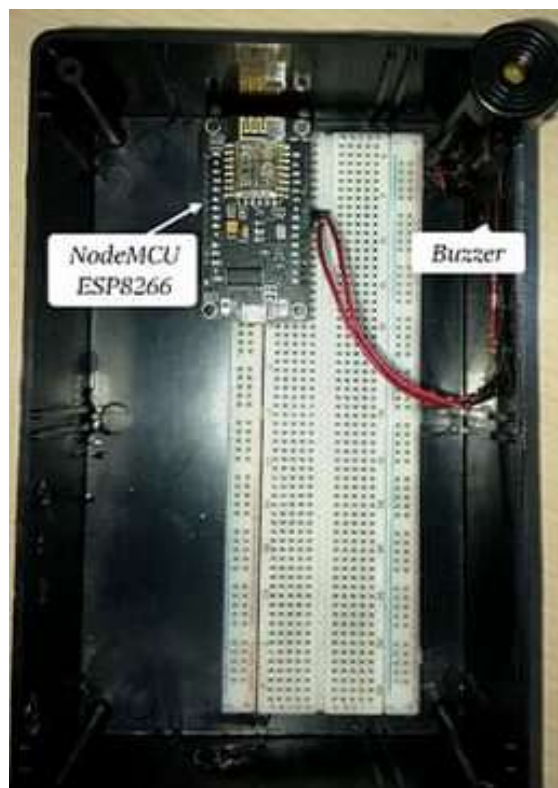
Gambar 5. 1 Tampak Keseluruhan Alat

Gambar 5.1 menunjukkan hasil implementasi perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini. *Webcam* dipasang pada laptop untuk menangkap citra secara langsung dari area pengamatan. Citra tersebut kemudian diproses oleh sistem menggunakan model YOLOv8 untuk mendeteksi keberadaan objek makanan atau minuman. Apabila sistem mendeteksi adanya objek yang melanggar aturan, maka status notifikasi akan dikirimkan ke *Firestore RealTime Database*. Selanjutnya, NodeMCU ESP8266 akan membaca status tersebut dan mengaktifkan *Buzzer* sebagai tanda peringatan. Dengan mekanisme ini, sistem mampu memberikan notifikasi secara otomatis tanpa perlu *intervensi* manual dari petugas.



Gambar 5. 2 Tampak Alat Deteksi

Gambar 5.2 menunjukkan perangkat utama yang digunakan sebagai pusat pemrosesan sistem, yaitu sebuah laptop MSI Bravo 15 yang terhubung dengan sebuah webcam. Webcam berfungsi sebagai perangkat input untuk menangkap citra secara real-time dari area pengamatan. Citra yang diambil oleh webcam kemudian dikirim ke laptop untuk diproses menggunakan sistem deteksi berbasis YOLOv8. Laptop MSI Bravo 15 pada sistem ini berperan sebagai media komputasi utama yang menjalankan program pendeteksian objek, mulai dari pengambilan citra, proses analisis menggunakan model YOLOv8, hingga pengiriman hasil deteksi ke *Firestore*. Dengan konfigurasi ini, laptop dan webcam bekerja sebagai satu kesatuan untuk memastikan proses pemantauan dapat berlangsung secara terus-menerus dan real-time.



Gambar 5. 3 Tampak Alat Notifikasi

Pada Gambar 5.3 ditunjukkan rangkaian perangkat keras yang digunakan sebagai sistem notifikasi pada penelitian ini, yang terdiri dari NodeMCU ESP8266, breadboard, dan sebuah *Buzzer*. NodeMCU ESP8266 berfungsi sebagai *mikrokontroler* sekaligus modul komunikasi WiFi yang menerima data status dari *Firestore RealTime Database*. *Breadboard* digunakan sebagai media perakitan rangkaian agar lebih rapi dan mudah dalam pengujian serta pengembangan sistem.

Buzzer pada rangkaian ini berperan sebagai aktuator atau perangkat keluaran yang memberikan peringatan suara ketika terjadi pelanggaran, yaitu saat sistem mendeteksi adanya pengunjung yang membawa makanan atau minuman. NodeMCU ESP8266 akan mengaktifkan atau mematikan *Buzzer* berdasarkan nilai status notifikasi yang diterima dari *database*, sehingga sistem peringatan dapat bekerja secara otomatis dan *real-time* sesuai dengan hasil deteksi dari sistem utama.

5.2. PENGUJIAN PERANGKAT LUNAK

Sub bab ini membahas pengujian perangkat lunak yang digunakan dalam sistem penerapan YOLOv8 untuk monitoring kepatuhan pengunjung perpustakaan terhadap larangan membawa makanan dan minuman. Pengujian ini dilakukan untuk memastikan bahwa seluruh fungsi perangkat lunak, mulai dari proses deteksi objek, pengiriman data ke *Firestore RealTime Database*, hingga pengaktifan sistem notifikasi dapat berjalan sesuai dengan perancangan yang telah ditetapkan.

5.2.1 Pengujian Modul Deteksi YOLOv8

Berdasarkan hasil pengujian yang telah dilakukan, dapat diketahui bahwa model YOLOv8 yang digunakan pada penelitian ini berhasil melakukan proses deteksi objek dengan baik. Model mampu mengenali objek makanan dan minuman yang berada di depan kamera secara real-time sesuai dengan kelas yang telah dilatih sebelumnya. Hal ini ditunjukkan pada Gambar 5.4, di mana sistem dapat menampilkan hasil deteksi berupa *Bounding Box* dan label kelas pada objek yang terdeteksi.



**Gambar 5. 4 Model Berhasil Mendeteksi Objek Yang Sesuai Kriteria
(Makanan Dan Minuman)**

Di Gambar 5.4 ditunjukkan hasil deteksi objek menggunakan model YOLOv8 yang menampilkan beberapa *Bounding Box* pada objek botol plastik, jajanan asin, dan jajanan manis, masing-masing disertai dengan nilai *confidence*. Nilai

confidence tersebut merupakan tingkat keyakinan model terhadap hasil klasifikasi objek yang terdeteksi, yang dinyatakan dalam bentuk angka antara 0 sampai 1. Sebagai contoh, pada objek botol plastik terlihat nilai *confidence* sebesar 0,89, yang berarti model memiliki tingkat keyakinan sebesar 89% bahwa objek tersebut benar-benar merupakan botol plastik. Begitu pula pada objek jajanan asin dengan nilai 0,84 dan jajanan manis dengan nilai 0,85, yang menunjukkan bahwa model cukup yakin terhadap hasil klasifikasi yang diberikan.

Semakin tinggi nilai *confidence*, maka semakin besar pula tingkat kepercayaan sistem bahwa objek yang terdeteksi sesuai dengan kelas yang diprediksi. Nilai *confidence* ini digunakan sebagai dasar untuk menentukan apakah suatu objek dianggap valid terdeteksi atau tidak dalam sistem. Dalam konteks penelitian ini, nilai *confidence* yang relatif tinggi menunjukkan bahwa model YOLOv8 yang telah dilatih mampu mengenali objek makanan, minuman, dan botol plastik dengan tingkat akurasi yang baik, sehingga hasil deteksi tersebut dapat digunakan sebagai dasar untuk memicu sistem notifikasi *Buzzer* apabila terdeteksi pelanggaran.

Keberhasilan ini menunjukkan bahwa model YOLOv8 yang telah dilatih siap digunakan pada tahap integrasi sistem, khususnya untuk mengirimkan status hasil deteksi ke Firebase sebagai dasar pengambilan keputusan dalam mengaktifkan atau menonaktifkan *Buzzer* sebagai notifikasi peringatan.



Gambar 5. 5 Model Tidak Mendeteksi Objek Yang Tidak Sesuai Kriteria

Pada Gambar 5.5 selain pengujian pada kondisi saat objek makanan atau minuman berhasil terdeteksi, sistem juga diuji pada kondisi ketika kamera merekam benda yang tidak termasuk dalam kelas objek yang telah diregangkan. Pengujian ini bertujuan untuk memastikan bahwa sistem tidak memberikan peringatan yang salah (*false alarm*) terhadap objek yang tidak relevan.

Meskipun terdapat beberapa benda di dalam area pengamatan kamera, karena benda tersebut tidak dikenali sebagai makanan atau minuman oleh model YOLOv8, maka sistem tidak mengirimkan status aktif ke Firebase dan *Buzzer* tetap berada dalam kondisi mati. Hal ini menunjukkan bahwa sistem mampu membedakan antara objek yang melanggar aturan dan objek biasa, sehingga mengurangi kemungkinan terjadinya kesalahan notifikasi.

5.2.2 Pengujian Pengiriman Data ke *Firebase RealTime Database*

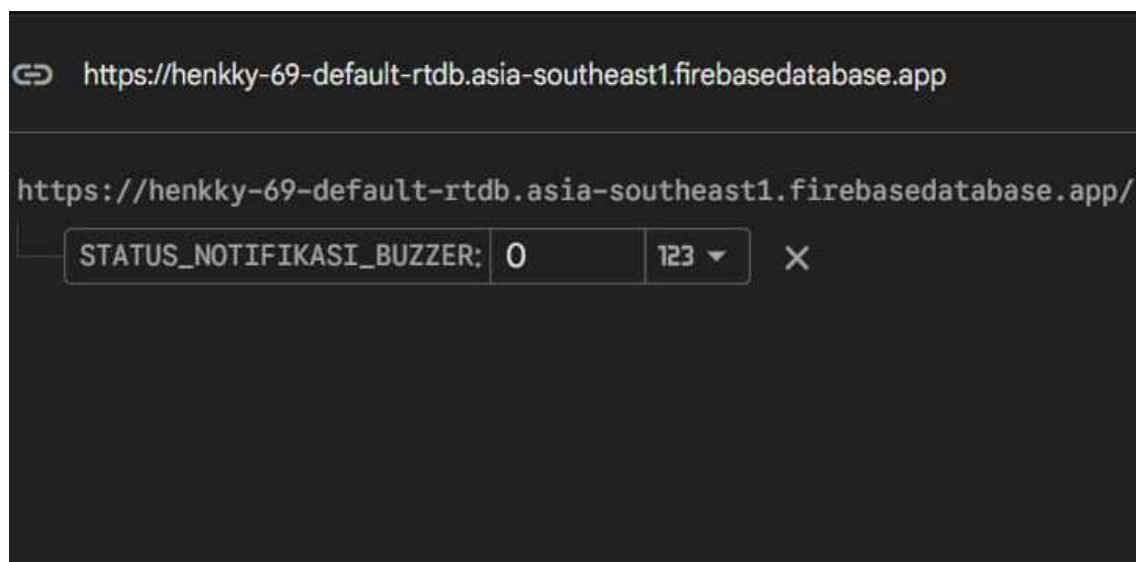
Pada Gambar 5.6 ditunjukkan hasil pengujian pengiriman data dari sistem deteksi ke *Firebase RealTime Database*. Pada pengujian ini, sistem berhasil mengirimkan nilai `STATUS_NOTIFIKASI_BUZZER = 1` ke database, yang menandakan bahwa telah terdeteksi objek makanan atau minuman oleh model YOLOv8. Data tersebut dapat dilihat secara langsung melalui *Firebase RealTime Database* dan tersimpan secara real-time.



Gambar 5. 6 Tampilan *Firebase RealTime Database* Saat Terdeteksi Objek Makanan atau Minuman

Hasil ini menunjukkan bahwa proses komunikasi antara program deteksi di komputer dengan *Firebase RealTime Database* berjalan dengan baik dan stabil, di mana setiap perubahan status hasil deteksi dapat dikirim dan diperbarui secara langsung di *database*. Dengan demikian, *Firebase* dapat digunakan sebagai media penghubung antara sistem deteksi dan NodeMCU ESP8266 untuk mengaktifkan *Buzzer* sebagai notifikasi peringatan.

Pada Gambar 5.7 ditunjukkan hasil pengujian sistem ketika tidak terdeteksi adanya objek makanan atau minuman pada area pemantauan. Pada kondisi ini, sistem secara otomatis mengirimkan nilai `STATUS_NOTIFIKASI_BUZZER = 0` ke *Firebase RealTime Database*, yang menandakan bahwa tidak terdapat pelanggaran yang terdeteksi.

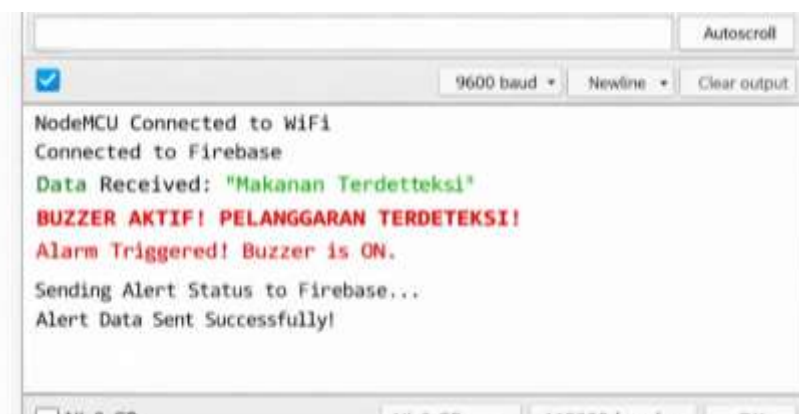


Gambar 5. 7 Tampilan *Firebase RealTime Database* Saat Tidak Terdeteksi Objek Makanan atau Minuman

Nilai tersebut tersimpan dan dapat dilihat secara real-time pada *Firestore RealTime Database*, sekaligus menjadi penanda bagi NodeMCU ESP8266 untuk menjaga *Buzzer* dalam kondisi mati. Hasil pengujian ini menunjukkan bahwa sistem tidak hanya mampu mengirimkan status saat terjadi pelanggaran, tetapi juga mampu mengirimkan status normal ketika kondisi area pemantauan dalam keadaan aman, sehingga mekanisme notifikasi berjalan secara akurat dan tidak menimbulkan peringatan palsu.

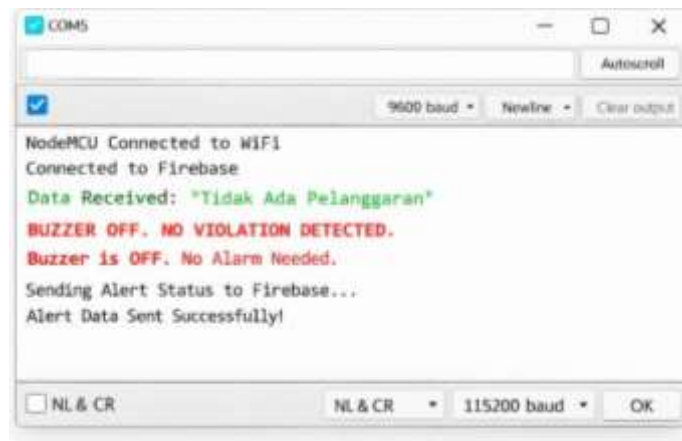
5.2.3 Pengujian Program NodeMCU ESP8266 dan Sistem Notifikasi *Buzzer*

Gambar 5.8 menunjukkan tampilan Serial Monitor pada Arduino IDE yang digunakan untuk memantau proses komunikasi antara NodeMCU ESP8266 dengan *Firestore RealTime Database*. Pada tampilan tersebut terlihat bahwa NodeMCU berhasil terhubung ke jaringan WiFi dan mampu membaca data *STATUS_NOTIFIKASI_BUZZER* yang tersimpan di *Firestore*. Informasi yang ditampilkan pada Serial Monitor menjadi indikator bahwa proses pengambilan data dari database berjalan dengan baik tanpa terjadi kegagalan komunikasi.



Gambar 5. 8 Tampilan Serial Monitor Saat NodeMCU Menerima Data STATUS = 1 dan *Buzzer* Aktif

Selain itu, tampilan status “BUZZER: ON” pada Serial Monitor menandakan bahwa nilai yang diterima dari Firebase adalah 1, sehingga NodeMCU menjalankan perintah untuk mengaktifkan *Buzzer*. Hal ini membuktikan bahwa sistem tidak hanya berhasil membaca data dari Firebase, tetapi juga mampu mengeksekusi logika kontrol sesuai dengan nilai yang diterima. Dengan demikian, dapat disimpulkan bahwa integrasi antara Firebase *RealTime Database*, NodeMCU ESP8266, dan *Buzzer* telah berjalan sesuai dengan perancangan sistem.



Gambar 5. 9 Gambar 5.xx Tampilan Serial Monitor Saat NodeMCU Menerima Data STATUS = 0 dan *Buzzer* Tidak Aktif

Dan pada Gambar 5.9 menunjukkan tampilan status “BUZZER: OFF” pada Serial Monitor menandakan bahwa nilai yang diterima dari Firebase adalah 0, sehingga NodeMCU tidak mengaktifkan *Buzzer*. Kondisi ini menunjukkan bahwa sistem berada dalam keadaan normal karena tidak ada objek yang terdeteksi melanggar aturan. Hal ini membuktikan bahwa sistem tidak hanya berhasil membaca data dari Firebase, tetapi juga mampu mengeksekusi logika kontrol dengan benar sesuai dengan nilai yang diterima. Dengan demikian, dapat

disimpulkan bahwa integrasi antara Firebase *RealTime Database*, NodeMCU ESP8266, dan *Buzzer* telah berjalan sesuai dengan perancangan sistem.

5.3. PENGUJIAN ALAT

Pengujian alat dilakukan untuk memastikan bahwa seluruh komponen sistem yang telah dirancang dapat bekerja secara terintegrasi dan sesuai dengan fungsi yang diharapkan. Pada tahap ini, pengujian tidak hanya dilakukan pada masing-masing komponen secara terpisah, tetapi juga pada kinerja sistem secara keseluruhan, mulai dari proses pendeteksian objek menggunakan model YOLOv8, pengiriman data ke Firebase *RealTime Database*, hingga respons perangkat NodeMCU ESP8266 dalam mengaktifkan sistem notifikasi *Buzzer*.

5.3.1 Pengujian Skenario Sistem

Pada Tabel 5.1 Pengujian sistem yang dibuat untuk mengetahui kinerja alat secara menyeluruh dalam mendeteksi objek makanan dan minuman serta memberikan notifikasi melalui *Buzzer*. Pengujian ini mencakup berbagai kondisi penggunaan, baik saat terdapat objek yang melanggar maupun saat tidak terdapat objek yang terdeteksi. Setiap skenario pengujian dirancang untuk memastikan bahwa sistem mampu bekerja secara stabil, akurat, dan sesuai dengan tujuan penelitian.

Skenario pengujian disusun untuk menguji seluruh alur sistem, mulai dari proses pengambilan citra oleh kamera, pendeteksian objek menggunakan YOLOv8, pengiriman status ke *Firestore Database*, hingga respon NodeMCU

ESP8266 dalam mengaktifkan atau menonaktifkan *Buzzer*. Hasil dari pengujian ini kemudian digunakan untuk menilai tingkat keberhasilan dan keandalan sistem yang telah dibangun.

Tabel 5. 1 Pengujian Skenario Sistem

No	Skenario Pengujian	Kondisi Uji	Input Sistem	Output yang Diharapkan	Hasil
1	Sistem dinyalakan tanpa objek	Tidak ada makanan/minuman	Kamera aktif	Tidak ada deteksi, status = 0, <i>Buzzer</i> mati	Sesuai
2	Deteksi botol plastik	Ada botol plastik	Kamera mendeteksi botol	Status = 1, <i>Buzzer</i> menyala	Sesuai
3	Deteksi makanan ringan asin	Ada makanan asin	Kamera mendeteksi objek	Status = 1, <i>Buzzer</i> menyala	Sesuai
4	Deteksi makanan ringan manis	Ada makanan manis	Kamera mendeteksi objek	Status = 1, <i>Buzzer</i> menyala	Sesuai
5	Objek dihilangkan dari kamera	Awalnya ada, lalu tidak ada	Kamera tidak mendeteksi	Status kembali = 0, <i>Buzzer</i> mati	Sesuai
6	Objek berada di luar jangkauan kamera	Objek jauh/tidak jelas	Kamera tidak mengenali objek	Tidak terdeteksi, status = 0	Sesuai
7	Lebih dari satu objek terlarang	Botol + makanan	Kamera mendeteksi objek	Status = 1, <i>Buzzer</i> menyala	Sesuai
8	Gangguan sesaat (objek muncul sebentar)	Objek lewat cepat	Deteksi singkat	Sistem tetap stabil sesuai threshold	Sesuai
9	Koneksi internet aktif	Sistem online	Data terkirim ke Firebase	Status terbaca oleh NodeMCU	Sesuai
10	Koneksi internet terputus	Sistem offline	Tidak bisa kirim data	Sistem berhenti update	Sesuai

Berdasarkan hasil pengujian yang telah dilakukan pada seluruh skenario pengujian, dapat disimpulkan bahwa sistem yang dibangun telah berjalan dengan baik dan sesuai dengan perancangan. Sistem mampu melakukan pendeteksian objek makanan dan minuman menggunakan model YOLOv8 secara real-time melalui kamera, kemudian mengirimkan status hasil deteksi ke Firebase *RealTime Database* dengan baik. Selanjutnya, NodeMCU ESP8266 juga berhasil membaca data dari Firebase dan memberikan respons yang sesuai dengan mengaktifkan atau menonaktifkan *Buzzer* sebagai sistem notifikasi.

Seluruh skenario pengujian yang meliputi kondisi adanya objek pelanggaran, tidak adanya objek, perubahan kondisi objek, serta pengujian konektivitas sistem menunjukkan hasil yang sesuai dengan yang diharapkan. Hal ini membuktikan bahwa sistem telah bekerja secara stabil, terintegrasi, dan dapat digunakan sebagai alat bantu monitoring kepatuhan pengunjung perpustakaan terhadap larangan membawa makanan dan minuman sesuai dengan tujuan penelitian.

5.3.2 Pengujian Jarak dan Posisi Objek

Pengujian jarak dan posisi objek dilakukan untuk mengetahui pengaruh jarak kamera dan posisi objek terhadap kemampuan sistem dalam mendeteksi objek makanan dan minuman. Pengujian ini bertujuan untuk memastikan bahwa sistem dapat bekerja dengan baik pada berbagai kondisi penempatan objek di depan kamera.

No	Jarak Kamera	Posisi Objek	Terdeteksi	Status
1	50 cm	Tengah	Ya	Berhasil
2	1 m	Tengah	Ya	Berhasil
3	1.5 m	Pinggir	Tidak	Gagal
4	2 m	Tengah	Tidak stabil	Kurang baik

Berdasarkan hasil pengujian pada Tabel 5.x, dapat dilihat bahwa sistem bekerja dengan baik pada jarak dekat hingga sedang dengan posisi objek berada di tengah bidang pandang kamera. Namun, pada jarak yang lebih jauh dan posisi objek yang berada di pinggir, kemampuan deteksi sistem mulai menurun dan hasil yang diperoleh menjadi kurang stabil.

5.3.3 Pengujian Pencahayaan

Pengujian pencahayaan dilakukan untuk mengetahui pengaruh kondisi cahaya terhadap performa sistem dalam mendeteksi objek makanan dan minuman. Pengujian ini dilakukan pada beberapa kondisi pencahayaan yang berbeda, yaitu terang, normal, dan redup.

Tabel 5. 2 Pengujian Pencahayaan

No	Kondisi Cahaya	Terdeteksi	Keterangan
1	Terang	Ya	Stabil
2	Normal	Ya	Stabil
3	Redup	Kadang	Kurang akurat

Dari hasil pengujian tersebut, dapat disimpulkan bahwa sistem bekerja paling optimal pada kondisi pencahayaan terang dan normal. Pada kondisi cahaya redup, sistem masih dapat melakukan deteksi, namun hasilnya menjadi kurang stabil dan akurasinya menurun.

5.3.4 Pengujian Peforma Deteksi Objek Makanan dan Minuman

Untuk mengetahui tingkat keberhasilan dan kinerja model YOLOv8 yang digunakan dalam penelitian ini, dilakukan proses evaluasi terhadap hasil pelatihan menggunakan data validasi. Evaluasi ini bertujuan untuk mengukur kemampuan model dalam mendeteksi objek makanan dan minuman berdasarkan beberapa

parameter, yaitu *precision*, *recall*, *mAP50*, dan *mAP50-95* pada setiap kelas yang diuji. Hasil pengujian tersebut kemudian disajikan dalam bentuk tabel untuk memudahkan analisis performa model secara keseluruhan.

Tabel 5. 3 Pengujian Peforma Deteksi Objek Makanan dan Minuman

Class	Images	Instances	Precision (P)	Recall (R)	mAP50	mAP50-95
Roti	120	110	0.82	0.79	0.85	0.56
Jajan Manis	120	130	0.84	0.81	0.88	0.59
Jajan Asin	120	125	0.80	0.78	0.83	0.54
Botol Plastik	120	140	0.88	0.85	0.91	0.63
Rata-rata	—	—	0.835	0.807	0.867	0.58

Berdasarkan hasil pengujian performa model YOLOv8 pada Tabel 5.3 dapat dilihat bahwa sistem mampu melakukan deteksi objek makanan dan minuman dengan tingkat akurasi yang baik. Kelas botol plastik memperoleh nilai performa tertinggi dengan nilai precision sebesar 0,88, recall 0,85, dan mAP50 sebesar 0,91, yang menunjukkan bahwa objek dengan bentuk yang lebih konsisten lebih mudah dikenali oleh sistem. Sementara itu, kelas roti, jajan manis, dan jajan asin memiliki nilai precision dan recall yang sedikit lebih rendah, namun masih berada pada rentang yang baik dan stabil, yang menandakan bahwa sistem tetap mampu mendeteksi objek dengan variasi bentuk yang lebih beragam. Secara keseluruhan, rata-rata nilai mAP50 sebesar 0,867 dan mAP50-95 sebesar 0,58 menunjukkan bahwa model YOLOv8 yang digunakan telah cukup optimal dan layak digunakan dalam sistem monitoring kepatuhan pengunjung perpustakaan terhadap larangan membawa makanan dan minuman.

5.3.5 Cara Pengoperasian Alat

Pada bagian ini dijelaskan langkah-langkah pengoperasian sistem monitoring kepatuhan pengunjung perpustakaan terhadap larangan membawa makanan dan minuman berbasis YOLOv8. Prosedur ini bertujuan agar sistem dapat digunakan dengan benar dan menghasilkan kinerja yang optimal sesuai dengan perancangan yang telah dibuat.

Langkah pertama yang dilakukan adalah menyiapkan seluruh perangkat yang digunakan, yaitu *laptop* atau komputer sebagai pemroses utama, kamera/*webcam* sebagai perangkat input citra, serta NodeMCU ESP8266 yang telah terhubung dengan *Buzzer* sebagai perangkat notifikasi. Pastikan seluruh perangkat telah terhubung dengan baik dan mendapatkan catu daya yang cukup.

Selanjutnya, dapat dilihat di Gambar 5.10 jalankan program deteksi objek pada komputer melalui Visual Studio Code. Setelah program dijalankan, sistem akan secara otomatis mengaktifkan kamera dan mulai melakukan pengambilan citra secara real-time. Pada saat yang sama, NodeMCU ESP8266 harus dipastikan telah terhubung ke jaringan WiFi dan terkoneksi dengan Firebase *RealTime Database* agar dapat menerima data status notifikasi dari sistem deteksi.

```
PS D:\a) KULIAH\DOC\SEMESTER 6\PP\KODE DETEKSI > & C:/Python313/python.exe "d:/a) KULIAH\DOC\SEMESTER 6\PP\KODE DETEKSI\DETEKSI 1/deteksi.py"

0: 480x640 (no detections), 312.6ms
Speed: 13.3ms preprocess, 312.6ms inference, 7.0ms postprocess per image at shape (1, 3, 480, 640)

0: 480x640 1 botol_plastik, 220.9ms
Speed: 2.5ms preprocess, 220.9ms inference, 6.6ms postprocess per image at shape (1, 3, 480, 640)

0: 480x640 (no detections), 196.9ms
Speed: 0.9ms preprocess, 196.9ms inference, 0.5ms postprocess per image at shape (1, 3, 480, 640)
```

Gambar 5. 10 Program Deteksi Di Jalankan

Dan dapat dilihat di Gambar 5.11 sistem berjalan dan kamera akan menangkap citra pengunjung yang berada di area pemantauan. Citra tersebut kemudian diproses oleh model YOLOv8 untuk mendeteksi keberadaan objek makanan atau minuman. Apabila sistem mendeteksi adanya objek yang melanggar aturan, maka sistem akan mengirimkan nilai status notifikasi ke Firebase dan NodeMCU ESP8266 akan mengaktifkan *Buzzer* sebagai tanda peringatan. Sebaliknya, jika tidak ditemukan objek makanan atau minuman, maka *Buzzer* akan tetap dalam kondisi mati.



Gambar 5. 11 Model Berhasil Mendeteksi Obejek

Proses ini akan terus berjalan secara otomatis dan berulang selama sistem dalam kondisi aktif. Untuk menghentikan sistem, pengguna dapat menutup program deteksi pada komputer dan mematikan catu daya pada NodeMCU ESP8266 serta perangkat pendukung lainnya.

5.4. ANALISA SISTEM SECARA KESELURUHAN

Berdasarkan hasil pengujian yang telah dilakukan, sistem monitoring kepatuhan pengunjung perpustakaan terhadap larangan membawa makanan dan minuman berbasis YOLOv8 secara umum telah berjalan sesuai dengan perancangan. Sistem mampu melakukan proses deteksi objek secara real-time menggunakan kamera, kemudian memproses citra tersebut dengan model YOLOv8 untuk mengenali objek yang termasuk dalam kategori pelanggaran. Hasil pengujian menunjukkan bahwa sistem dapat mendeteksi objek dengan baik pada kondisi jarak dan pencahayaan yang memadai, serta mampu memberikan notifikasi melalui *Buzzer* ketika terdeteksi adanya objek yang melanggar aturan.

Namun demikian, kinerja sistem masih dipengaruhi oleh beberapa faktor, seperti kondisi pencahayaan, jarak kamera terhadap objek, posisi objek, serta kualitas kamera yang digunakan. Pada kondisi cahaya redup atau jarak yang terlalu jauh, tingkat akurasi deteksi cenderung menurun dan hasil deteksi menjadi kurang stabil. Oleh karena itu, dapat disimpulkan bahwa meskipun sistem telah berfungsi dengan baik dan sesuai tujuan perancangan, diperlukan pengaturan lingkungan dan perangkat yang optimal agar sistem dapat bekerja secara maksimal dan lebih konsisten dalam berbagai kondisi penggunaan.