


Implementasi Algoritma Yolo untuk Monitoring Jumlah Mahasiswa dan Kontrol Otomatis pada Ruang Kelas

Abdillah Khalid¹, Dedi Ary Prasetya¹

¹Universitas Muhammadiyah Surakarta, Jl. A. Yani, Mendungan, Pabelan, Kec. Kartasura, Kabupaten Sukoharjo, Jawa Tengah 57162, Surakarta, Indonesia

 Email korespondensi: abdillah156010@gmail.com

Abstrak. Pengelolaan ruang kelas secara konvensional yang bergantung pada intervensi manual rentan terhadap *human error* dan menyebabkan inefisiensi penggunaan energi, serta menghasilkan data okupansi yang tidak akurat dan minimnya informasi status ruangan secara *real-time*. Untuk mengatasi masalah tersebut, penelitian ini mengembangkan sebuah sistem terintegrasi untuk melakukan monitoring jumlah mahasiswa dan otomasi kontrol perangkat berbasis *computer vision*. Metodologi yang diterapkan mengintegrasikan algoritma deteksi YOLOv10s (dengan *Confidence Score* 0.4) yang dieksekusi pada GPU RTX 3050 TI, unit kontrol berbasis *microcontroller* ESP32 dengan sensor suhu DHT22, serta platform website yang dibangun menggunakan *framework* Flask dan terhubung dengan Firebase (Firestore untuk data historis dan Realtime Database untuk data *real-time*). Hasil pengujian menunjukkan sistem ini sangat efektif: dari sisi akurasi, sistem mencapai tingkat keberhasilan di atas 90% pada skenario statis, meskipun mengalami fluktuasi pada skenario dinamis akibat oklusi dan instabilitas jaringan yang memengaruhi total latensi. Secara fungsional, sistem kontrol otomatis untuk aktuator juga terbukti berjalan responsif dan andal. Dengan demikian, disimpulkan bahwa sistem yang diimplementasikan ini fungsional, andal, dan menawarkan solusi konkret untuk meningkatkan efisiensi energi. Penelitian ini juga berhasil menyajikan bukti konsep (*proof of concept*) dan referensi implementasi yang solid untuk sistem sejenis, dengan potensi pengembangan lebih lanjut pada penanganan oklusi yang lebih kompleks dan optimasi kinerja jaringan.

Kata kunci: YOLOv10; Computer Vision; Monitoring; Otomasi Ruangan



PENDAHULUAN

Ruangan seperti ruangan kelas merupakan salah satu fasilitas penting dalam lingkungan pendidikan. Namun, terkadang dalam pengelolaannya kerap menghadapi berbagai kendala yang sering terjadi seperti ketidaksesuaian jadwal penggunaan, jumlah mahasiswa yang melebihi kapasitas, serta minimnya informasi mengenai status pemakaian ruangan secara *real-time* [1]. Dalam proses manajemen ruangan kebanyakan masih menggunakan metode manual seperti mencatat jumlah kehadiran ataupun pengawasan langsung, tetapi metode ini sering kali tidak efisien dan rawan kesalahan, terutama ketika banyak orang yang memasuki dan meninggalkan ruangan bersamaan [2].

Adapun beberapa aspek lain yang dikelola pada ruangan kelas seperti pencahayaan dan suhu ruangan, suhu ruangan biasanya bergantung pada alat elektronik seperti kipas angin dan pencahayaan ruangan bergantung pada lampu [3]. Dalam pengoperasian lampu dan kipas angin ini masih menggunakan sistem manual dimana masih menggunakan saklar *on-off*, hal ini memungkinkan adanya *human error*, salah satunya ialah perangkat elektronik masih terus menyala meskipun sudah tidak digunakan lagi [4].

Perkembangan teknologi pada era globalisasi ini sangatlah pesat dan masyarakat juga telah bergantung pada hal tersebut, sehingga membuat teknologi menjadi dasar dari segala kegiatan manusia [5]. Salah satu perkembangan teknologi yang ada pada bidang elektronika ini memberikan dampak positif di berbagai aspek dan juga sektor. Salah satunya terbentuk sistem teknologi *people counting*, dimana teknologi ini melibatkan algoritma pemrosesan gambar dan *Deep Learning* untuk menghitung jumlah orang secara benar dan tepat.

Dari permasalahan yang sudah disebutkan, serta gambaran teknologi yang akan digunakan, maka diperlukan sistem yang dapat menjadi solusi untuk permasalahan di atas dengan memanfaatkan dan mengimplementasikan teknologi yang telah dijabarkan, sistem penghitung orang berperan penting dalam berbagai hal terkait ruangan seperti keamanan ruangan, manajemen ruangan [6]. Dengan menerapkan sistem penghitung orang, pengguna dapat mengontrol aktuator pada ruangan tersebut, contohnya mengontrol lampu ataupun kipas angin ketika ada atau tidaknya orang dalam ruangan [7].

Maka dari itu, pada penelitian tugas akhir ini yang berjudul “Implementasi Algoritma YOLO untuk *Monitoring* Jumlah Mahasiswa dan Kontrol Otomatis pada Ruang Kelas” bertujuan untuk memecahkan permasalahan yang telah disebutkan. penelitian ini menggunakan *library* OpenCV untuk pemrosesan citra serta menggunakan YOLOv10 yang merupakan salah satu algoritma dari *Deep Learning* untuk *object detection* guna



menghitung dan mendeteksi manusia secara *real-time*. Adapun beberapa komponen elektronika tambahan seperti ESP32, *Relay*, dan sensor untuk mengontrol aktuator.

METODE

Persiapan Alat dan Bahan

Penelitian tugas akhir yang berjudul “Implementasi Algoritma YOLO untuk Monitoring Jumlah Mahasiswa dan Kontrol Otomatis pada Ruang Kelas” memiliki beberapa alat dan bahan yang yang harus dipenuhi dalam proses pembuatannya. Seluruh alat dan bahan ini merupakan fondasi utama yang menunjang proses perancangan, implementasi, dan pengujian agar sistem dapat berfungsi secara andal dan sesuai dengan tujuan yang diinginkan.

Tabel 1. Alat dan Bahan

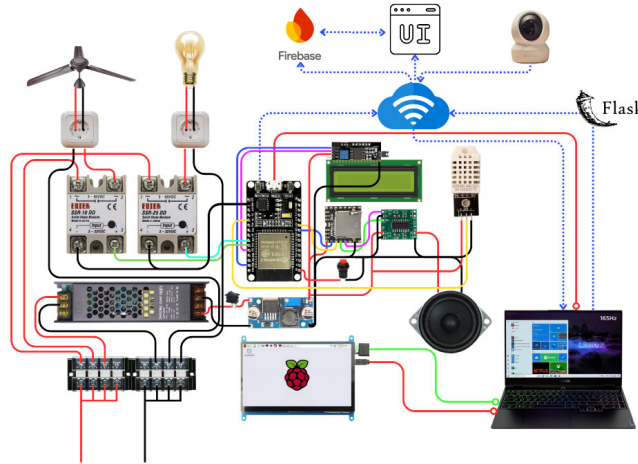
Alat dan Bahan	Jumlah
ESP32	1
SSR 10DA	2
LM2596	1
<i>Power supply</i>	1
Push Button	1
Switch	1
Layar LCD 7inch	1
LCD I2C 16x2	1
Dfplayer	1
Mini Amplifier HW-104	1
Speaker 4ohm	1
CCTV	1
Panel Box	1
Terminal 3X2	2
DHT22	1
Laptop	1
Stop Kontak 1L	2

Perancangan Sistem

Secara garis besar, sistem pendeteksi jumlah orang ini dirancang agar pengguna dapat mengetahui dan memantau jumlah orang di dalam ruangan, status ruangan, serta pola penggunaannya pada waktu tertentu. Selain itu, sistem ini juga memiliki fitur untuk mengontrol beberapa aktuator yang terhubung seperti lampu dan kipas angin. Pengoperasian perangkat tersebut secara manual sering kali tidak

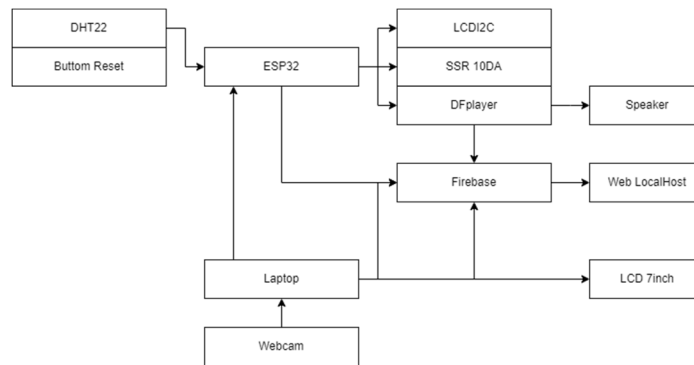


efisien akibat faktor *human error*, misalnya lupa mematikan perangkat saat ruangan tidak lagi digunakan, sehingga menyebabkan pemborosan energi maka dari itu, sistem ini tidak hanya memudahkan pemantauan, tetapi juga secara proaktif meningkatkan efisiensi penggunaan energi dengan meminimalisasi kelalaian manusia.



Gambar 1. Wiring Diagram

Gambar 1 merupakan diagram pengkabelan (*wiring diagram*) yang menggambarkan keseluruhan sistem dan rangkaian yang digunakan pada alat. Diagram ini disusun untuk memberikan gambaran menyeluruh mengenai hubungan antar komponen, serta protokol komunikasi yang digunakan, sehingga dapat mempermudah proses perancangan, pemrograman, dan pengujian alat di tahap selanjutnya.



Gambar 2. Blok Diagram Sistem

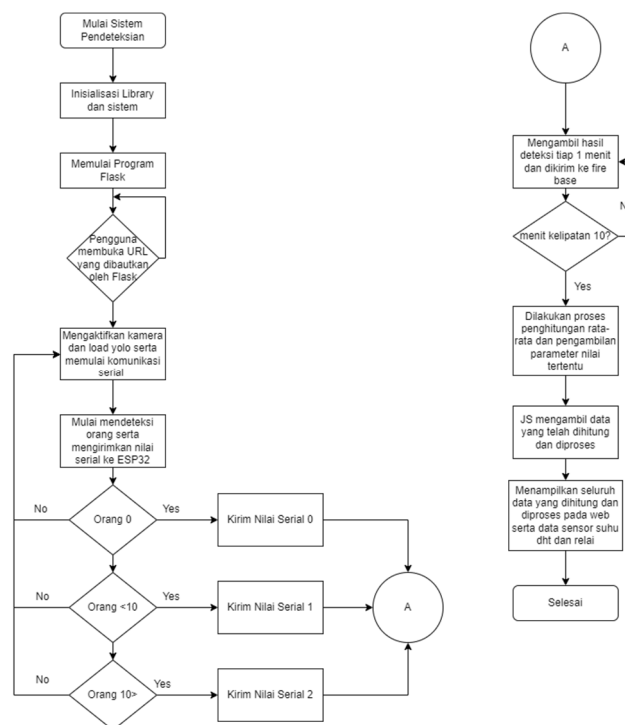
Diagram sistem yang ditunjukkan pada gambar 2 menggambarkan integrasi seluruh komponen yang digunakan. Proses pendeteksian orang dilakukan oleh CCTV dan diproses di laptop. Sementara itu, ESP32 bertugas sebagai unit kontrol utama yang terhubung dengan sensor (DHT22) dan aktuator (SSR, DFPlayer). Sebagai pusat data, Firebase menerima data dari dua sumber hasil pemrosesan jumlah orang dari laptop dan



data sensor dari ESP32. Seluruh data ini kemudian ditampilkan secara *real-time* pada antarmuka.

Perancangan Perangkat Lunak

Perancangan perangkat lunak sistem ini menggunakan beberapa teknologi utama yang dikembangkan di Visual Studio Code. Program inti dibuat dengan Python, di mana pustaka OpenCV digunakan untuk membaca input kamera dan Ultralytics (YOLO) untuk logika deteksi orang. *Framework* Flask kemudian berperan untuk membuat *web server* yang menyajikan *video stream* dan data ke pengguna, untuk membangun dasbor web, HTML, CSS, dan JavaScript digunakan, Unit kontrol ESP32 diprogram melalui Arduino IDE guna membaca sensor suhu DHT22 dan mengontrol *relay*. Seluruh komponen ini terhubung ke Firebase, untuk menyimpan dan menyajikan semua data secara *real-time*. Secara sederhana seluruh sistematisa dan cara kerja program dapat dilihat pada *flowchart* yang tertampil pada gambar 3.

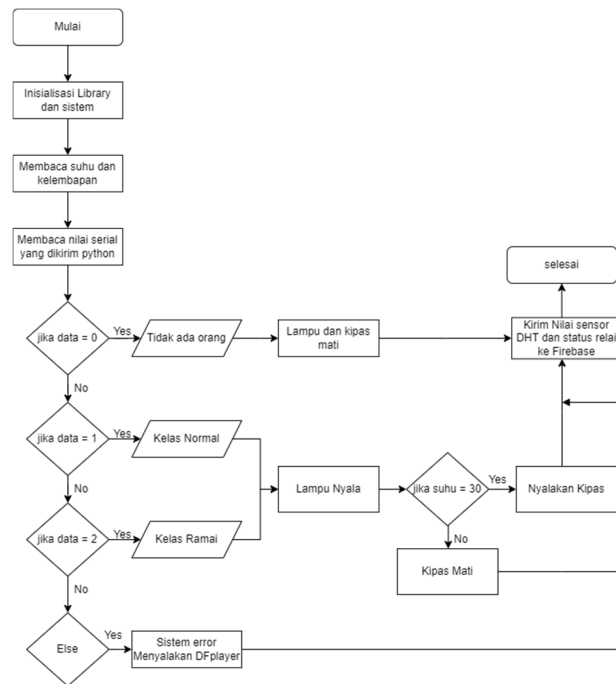


Gambar 3. Flowchart Sistem Pendeteksian

Gambar 3 merupakan *flowchart* sistem pendeteksian yang menggambarkan alur kerja sisi *server*, yang mencakup program Python dan website. Proses diawali dengan inisialisasi pustaka dan menjalankan *server* Flask. Saat pengguna mengakses alamat *web* yang disediakan, program akan secara otomatis menampilkan *video stream* hasil deteksi. Secara bersamaan, hasil deteksi tersebut juga diproses oleh algoritma untuk dicatat ke



Firestore dan ditampilkan pada website. Terakhir, sistem mengirimkan data hasil deteksi melalui komunikasi serial ke ESP32 untuk mengontrol *actuator*.



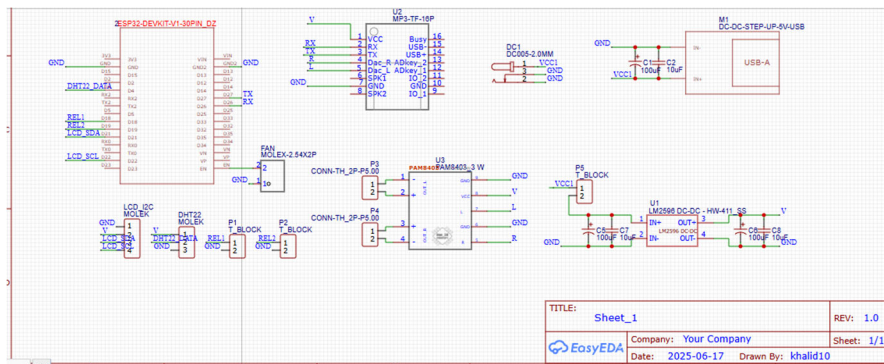
Gambar 4. *Flowchart* Sistem Kontrol

Untuk gambar 4 ini merupakan *flowchart* sistem kontrol, bagian ini menjelaskan alur kerja pada sistem mikrokontroler ESP32, yang bertugas menerima data serial dari Python untuk menerapkan logika kontrol. Aturannya adalah: jika ada orang, lampu akan menyala; dan jika ada orang sekaligus suhu melebihi ambang batas (*setpoint*), maka kipas akan menyala. Apabila tidak ada orang (data serial bernilai 0), semua aktuator akan dimatikan. Selain itu, ESP32 juga mengirimkan data suhu dari sensor DHT22 dan status *relay* terkini ke Firebase untuk ditampilkan pada website.

Perancangan Elektronika

Perancangan elektronika dimulai dengan merancang rangkaian PCB yang mengintegrasikan berbagai komponen seperti ESP32, sensor-sensor, serta aktuator ke dalam PCB (*Printed Circuit Board*). Proses perancangan ini dilakukan menggunakan perangkat lunak EasyEDA, seperti yang ditunjukkan pada Gambar 5.



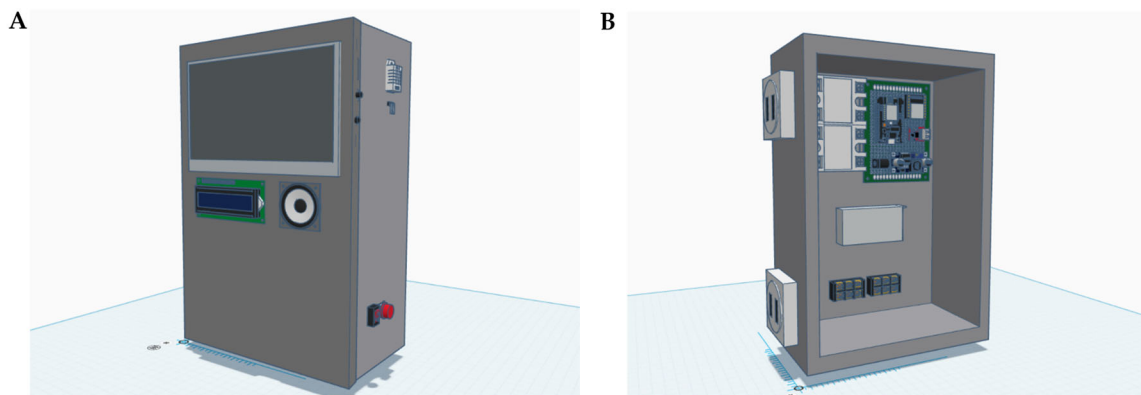


Gambar 5. Schematic PCB

Gambar 5 menunjukkan *Schematic* dari rangkaian PCB yang dirancang. Pada skematik ini, seluruh komponen saling terhubung sesuai dengan sistem yang telah ditentukan. Terdapat beberapa konektor seperti *Terminal Block* dan *Molex* yang berfungsi untuk menghubungkan sensor dan aktuator eksternal seperti *relay*, sensor DHT22, dan LCD ke dalam rangkaian PCB. Hal ini dikarenakan beberapa komponen tersebut tidak ditanam langsung di dalam PCB yang dibuat. Selain itu, terdapat juga dua modul *step-down*: satu modul digunakan untuk menyuplai daya ke berbagai sensor dan aktuator, sedangkan modul *step-down* lainnya adalah tipe DC-DC USB yang berfungsi khusus untuk menyuplai daya bagi layar LCD 7 inci.

Perancangan Hardware

Gambar 6 dan 7 menunjukkan desain 3D atau gambaran rancangan bagaimana bentuk alat akan dibuat serta bagaimana susunan komponen di dalam dan luar alat.



Gambar 6. (A) Rancangan 3D Tampak Luar, dan (B) Rancangan 3D Tampak Dalam

Dapat dilihat dari Gambar 6 (A) bahwa alat dikemas dalam sebuah kotak berukuran 28 cm × 19 cm × 10 cm. Pada bagian luar kotak ini, ditempatkan berbagai komponen antarmuka seperti sensor DHT22, layar LCD 7 inci, *speaker*, tombol *push button reset*, saklar



on-off sistem, serta layar LCD I2C 16×2. Selain itu, sebuah CCTV akan diletakan sesuai dengan kebutuhan sistem.

Selanjutnya, Gambar 6 (B) memperlihatkan tata letak komponen di dalam kotak. Di dalamnya terdapat dua modul SSR dan papan PCB yang telah dirancang dengan berbagai komponen tertanam. Terdapat pula sebuah *Power supply Unit* (PSU) yang berfungsi mengubah tegangan AC menjadi DC 12V 5A untuk menyuplai daya ke seluruh rangkaian pada PCB. Selain itu, digunakan sebuah blok terminal untuk mendistribusikan jalur daya masukan yang awalnya satu menjadi tiga keluaran, yaitu dua jalur untuk masing-masing modul SSR dan satu jalur untuk PSU.

Pengujian Sistem

Pada tahap pengujian, alat yang telah dirancang dan diprogram diuji coba secara menyeluruh untuk memastikan cara kerja dan sistematikanya telah sesuai dengan rancangan awal. Selama proses pengujian, dilakukan pengambilan data berdasarkan parameter-parameter yang sudah ditentukan untuk mengukur kinerja sistem. Apabila ditemukan kesalahan atau *error* yang signifikan, maka akan dilakukan analisis dan perbaikan pada rancangan maupun program agar alat dapat beroperasi secara optimal. Tahap ini bertujuan untuk memvalidasi bahwa seluruh fungsi dan cara kerja alat telah berjalan dengan semestinya.

Perhitungan Nilai Akurasi Pendeteksian Sistem

Akurasi deteksi dihitung dengan membandingkan hasil deteksi sistem terhadap kondisi sebenarnya di lapangan (*ground truth*) dalam berbagai skenario. Pengujian ini dipengaruhi oleh pengaturan ambang batas *Confidence Score*, yaitu nilai yang menunjukkan tingkat keyakinan model YOLO terhadap objek yang dideteksinya.

False Positive (FP) adalah kesalahan ketika sistem mendeteksi target (orang) yang sebenarnya tidak ada. Risiko kesalahan ini meningkat apabila ambang batas *confidence score* diatur terlalu rendah, sehingga membuat sistem menjadi terlalu sensitif. FP dihitung menggunakan rumus berikut:

$$FP = \text{Jumlah Terdeteksi} - \text{Jumlah Sebenarnya} \quad (\text{hanya jika hasilnya} > 0)$$

False Negative (FN) adalah kesalahan ketika sistem tidak mendeteksi target (orang) yang sebenarnya ada. Risiko kesalahan ini meningkat apabila ambang batas *confidence score* diatur terlalu tinggi, sehingga membuat sistem menjadi terlalu sensitif. FN dihitung menggunakan rumus berikut:

$$FN = \text{Jumlah Sebenarnya} - \text{Jumlah Terdeteksi} \quad (\text{hanya jika hasilnya} > 0)$$



Setelah nilai *false positive* (FP) dan *false negative* (FN) diperoleh dari setiap skenario pengujian, akurasi keseluruhan sistem deteksi dapat dihitung. Akurasi merepresentasikan tingkat keberhasilan sistem dalam mendeteksi target dengan mempertimbangkan kedua jenis kesalahan tersebut. Nilai akurasi dihitung menggunakan rumus berikut:

$$Akurasi(\%) = \left(1 - \frac{FP+FN}{Jumlah\ Sebenarnya}\right) \times 100\%$$

Perhitungan Nilai Akurasi Data Agregat

Pengujian ini dilakukan untuk memvalidasi fitur peringkasan data (rata-rata per 10 menit). Metode ini membandingkan hasil perhitungan rata-rata dari sistem otomatis dengan jumlah orang yang sebenarnya pada menit ke-10 berdasarkan pencatatan manual. Selisih nilai agregat menunjukkan deviasi atau perbedaan antara nilai rata-rata dari sistem otomatis dengan jumlah orang yang dicatat secara manual pada menit ke-10. Nilai ini dihitung menggunakan rumus berikut:

$$Selisih\ Rata-Rata = |Nilai\ Manual\ pada\ Menit\ ke\ -\ 10 - Rata - rata\ Sistem|$$

Akurasi agregat merepresentasikan persentase keandalan sistem dalam meringkas data selama periode waktu tertentu (10 menit), dengan membandingkan rata-rata hasil sistem terhadap nilai acuan manual yang dicatat pada menit ke-10. Nilai akurasi dihitung menggunakan rumus berikut:

$$Akurasi\ Agregat = \left(1 - \frac{Nilai\ Manual\ pada\ Menit\ ke-10}{Rata-Rata\ Manual}\right) \times 100\%$$

HASIL

Komparasi Antara Model YOLOv10

Tahap ini guna menentukan varian YOLO dari model YOLOv10 yang paling cocok dengan sistem yang digunakan, model YOLOv10 sendiri memiliki 5 versi yakni n (*nano*), s (*small*), m (*medium*), l (*large*) dan x (*extralarge*). Dalam alat ini, otak pemrosesan pendeteksian YOLO dieksekusi oleh laptop dengan mengandalkan GPU RTX3050TI dengan VRAM 4GB. Model yang akan dipakai ialah YOLOv10s, karena versi ini yang paling efisien dan memberikan hasil yang baik dalam pemrosesan. Pemilihan ini berdasarkan nilai mAP, efek versi terhadap GPU, serta kecocokannya dengan kamera yang digunakan, karena kamera yang digunakan ialah CCTV dengan resolusi 2MP, maka tipe s merupakan versi yang paling cocok agar tidak *overkill* dengan sistem. Dikarenakan model yang lebih tinggi membutuhkan resolusi kamera yang lebih tinggi pula, agar dapat memaksimalkan performa pendeteksinya, hasil penilaian dapat dilihat pada tabel 2.

Tabel 2. Komparasi Model YOLOv10

No	Model YOLO	mAP (Class Person)	FPS	Penggunaan GPU	Suhu GPU (Interval 2 jam)
----	------------	--------------------	-----	----------------	---------------------------



1	YOLOv10n	74,40%	42-70	21%-38%	51-53
2	YOLOv10s	80,00%	40-63	24%-40%	53-55
3	YOLOv10m	83,30%	26-43	42%-72%	60-63

Berdasarkan Tabel 2, nilai mAP dari YOLOv10s naik secara signifikan tanpa membebani GPU yang digunakan dan suhu pun stabil, berbeda dengan YOLOv10m dimana FPS turun secara drastis dengan penggunaan GPU yang cukup tinggi serta suhu yang juga tergolong cukup tinggi, sehingga diputuskan pada alat ini akan menggunakan YOLOv10s.

Penentuan *Confidence Score* yang Tepat pada Sistem

Pengujian bertujuan untuk membandingkan hasil deteksi sistem dengan hasil sesungguhnya yang ada dilapangan dengan menentukan beberapa nilai *Confidence Score* yang memang cocok untuk sistem.

Tabel 3. Komparasi *Confidence Score* dengan Beberapa Skenario

No. Uji	Confidence Threshold	Jumlah Orang Sebenarnya	Kondisi Skenario	Jumlah Terdeteksi	FP	FN	Akurasi (%)
1	0.0	5	Pencahayaan Baik	6	1	0	80%
2	0.2	5	Pencahayaan Baik	5	0	0	100%
3	0.4	5	Pencahayaan Baik	5	0	0	100%
4	0.6	5	Pencahayaan Baik	4	0	1	80%
5	0.8	5	Pencahayaan Baik	3	0	2	60%
6	0.0	5	Pencahayaan Redup	7	2	0	60%
7	0.2	5	Pencahayaan Redup	6	1	0	80%
8	0.4	5	Pencahayaan Redup	5	0	0	100%
9	0.6	5	Pencahayaan Redup	4	0	1	80%
10	0.8	5	Pencahayaan Redup	3	0	2	60%

FP = *False Positif* (Kesalahan sistem dimana mendeteksi object yang bukan kelasnya)

FN = *False Negatif* (Kesalahan sistem dimana tidak bisa mendeteksi object yang merupakan kelasnya)

Dalam tahap pengujian hasil deteksi YOLOv10s ini pengguna atur beberapa nilai *Confidence Score* yakni 0.0, 0.4, 0.6 dan 0.8 serta beberapa kondisi skenario yakni apabila cahaya redup dan apabila cahaya terang, setelah melakukan pengujian didapat bahwa

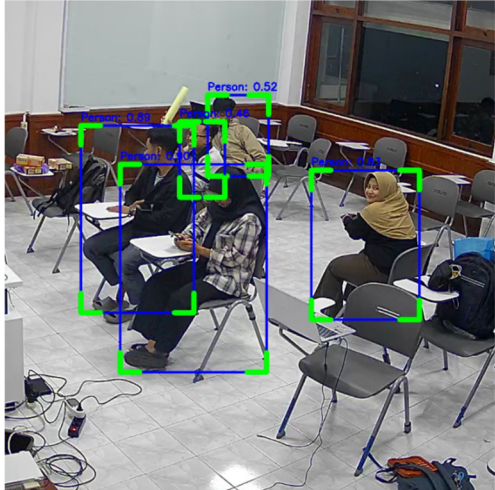
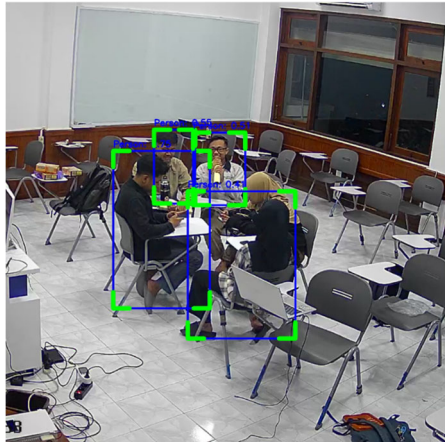


dengan pengatur *Confidence Score* 0.4 maka sistem akan lebih akurat dalam mendeteksi dan menghitung jumlah orang yang berada didalam ruangan.

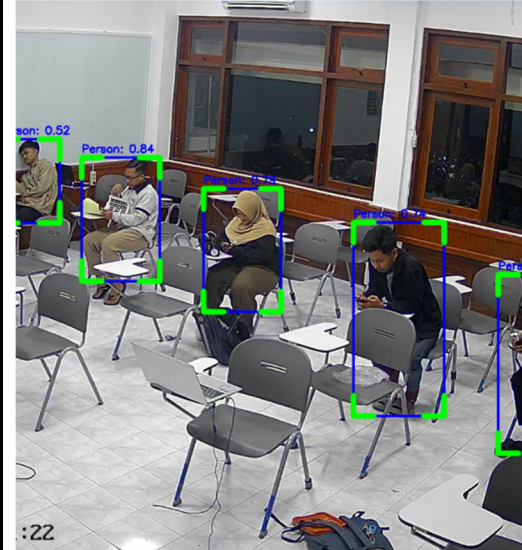
Uji Sistem dengan Beberapa Skenario Tertentu

Setelah menentukan model yang cocok dengan sistem serta *Confidence Score* yang sesuai maka sistem akan di uji dengan beberapa skenario yang akan terjadi, hasil yang dideteksi oleh sistem akan dibandingkan dengan hasil sebenarnya, skenario tersebut berjumlah 5 skenario dimana meliputi saat berdempetan, berjejer, menyamping, melingkar dan menghadap belakang.

Tabel 4. Uji Hasil Deteksi YOLO dengan Beberapa Skenario

No	Skenario	Jumlah Aktual	Jumlah Deteksi	Akurasi	Bukti Gambar
1	Berdempetan	5	5	100%	
2	Melingkar	5	4	80%	



3	Berjejer Menyamping	5	5	100%	
4	Berjejer Kebelakang	5	5	100%	
5	Menghadap Belakang	5	5	100%	



Sebagaimana ditunjukkan pada Tabel 4, hasil pengujian dari kelima skenario secara umum menunjukkan kinerja yang baik. Hasil deteksi sistem pada sebagian besar skenario telah sesuai dengan jumlah orang sebenarnya, yang mengindikasikan tingkat akurasi yang tinggi. Meskipun demikian, ditemukan satu anomali pada skenario "melingkar". Dalam skenario tersebut, sistem hanya mendeteksi empat orang dari total lima orang yang seharusnya terhitung. Kesalahan ini diduga terjadi karena sistem kesulitan mengidentifikasi salah satu objek akibat posisi yang tumpang tindih (oklusi) dalam formasi tersebut.

Perhitungan Nilai Akurasi Data Agregat

Pengujian ini bertujuan untuk mengukur seberapa akurat sistem dengan membandingkan hasil perhitungan sistem dengan pencatatan manual. Pengujian ini akan dilakukan dalam 2 kondisi. Pertama adalah kondisi di mana kelas memiliki subjek yang sebagian besar terus-menerus bergerak dan posisinya berubah-ubah, dalam kasus ini pengguna sebut sebagai skenario dinamis. Kondisi lainnya di mana kelas memiliki subjek yang sebagian besar tidak bergerak dan posisinya teratur, yang dalam kasus ini akan dikenal sebagai skenario statis

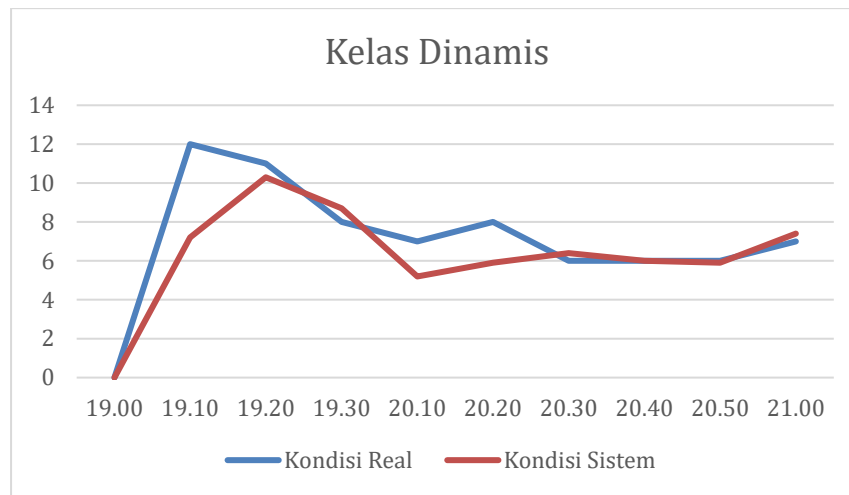
Tabel 5. Akurasi Data Agregat Kelas Dinamis

No	JAM	Kondisi Real	Kondisi Sistem	Selisih	Akurasi Agregat
1	19.00	0	0	0,00	100%
2	19.10	12	7,2	4,80	60,00%
3	19.20	11	10,3	0,70	93,64%
4	19.30	8	8,7	0,70	91,25%
5	20.10	7	5,2	1,80	74,29%
6	20.20	8	5,9	2,10	73,75%
7	20.30	6	6,4	0,40	93,33%
8	20.40	6	6	0,00	100%
9	20.50	6	5,9	0,10	98,33%
10	21.00	7	7,4	0,40	94,29%

Pada skenario ini, *sampling* data diambil mulai dari jam 19:00 hingga 21:00, meskipun sempat terjadi pemberhentian pengambilan data dari jam 19:31 hingga 19:59 akibat gangguan konektivitas yang membutuhkan waktu perbaikan. Dari data yang berhasil dikumpulkan, seperti yang terlihat pada Tabel 5, nilai akurasi pada skenario dinamis ini cenderung naik turun. Akurasi tertinggi yang dicatat sistem mencapai 100%, yang terjadi pada saat subjek tidak banyak keluar masuk ruangan ataupun berpindah posisi. Sementara itu, akurasi terendah tercatat sebesar 60,00% pada jam 19:10, yang diakibatkan oleh banyaknya subjek baru memasuki ruangan secara bersamaan sehingga



perubahan posisi dan jumlah dalam satu waktu mengganggu pencatatan data oleh sistem.



Gambar 7. Hubungan Antara Kondisi Real dengan Sistem pada Kelas Dinamis

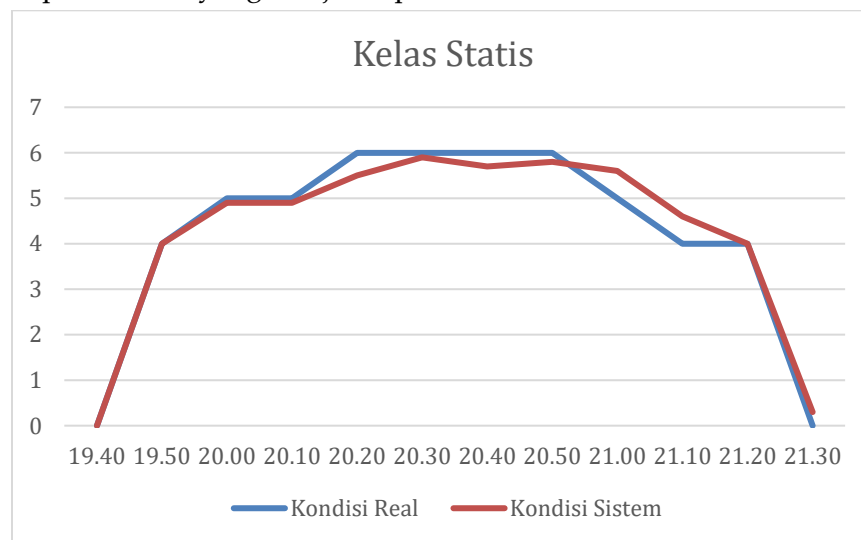
Gambar 11 menyajikan grafik perbandingan antara data yang dihasilkan oleh sistem dengan data pencatatan manual. Kurva yang dihasilkan oleh sistem terlihat lebih halus dan tidak memiliki puncak setajam kurva data *real*. Puncak pada data *real* mencapai 12 orang, sedangkan sistem hanya berada di rata-rata 7-8 orang karena sistem juga menghitung jumlah orang pada menit sebelumnya. Meskipun demikian, kurva sistem dapat mengikuti tren umum dari kurva data *real* dan memvalidasi perubahan okupansi secara keseluruhan.

Tabel 6. Akurasi Data Agregat Kelas Statis

No	JAM	Kondisi <i>Real</i>	Kondisi Sistem	Selisih	Akurasi Agregat
1	19.40	0	0	0,00	100%
2	19.50	4	4	0,00	100%
3	20.00	5	4,9	0,10	98,00%
4	20.10	5	4,9	0,10	98,00%
5	20.20	6	5,5	0,50	91,67%
6	20.30	6	5,9	0,10	98,33%
7	20.40	6	5,7	0,30	95,00%
8	20.50	6	5,8	0,20	96,67%
9	21.00	5	5,6	0,60	88,00%
10	21.10	4	4,6	0,60	85,00%
11	21.20	4	4	0,00	100%
12	21.30	0	0,3	0,30	0%



Pada skenario statis, pengambilan data dilakukan dari pukul 19.40 hingga 21.30, sebagaimana ditampilkan pada Tabel 6. Hasil pengujian menunjukkan bahwa sistem memiliki tingkat akurasi yang sangat tinggi. Hampir seluruh data uji menghasilkan akurasi yang mendekati 100%, dengan pengecualian pada data terakhir yang menunjukkan selisih kecil namun berdampak signifikan terhadap nilai akurasi agregat. Dapat disimpulkan bahwa pencatatan sistem pada skenario ini sangat presisi. Hal ini disebabkan oleh kondisi subjek yang cenderung diam, sehingga memudahkan proses deteksi oleh sistem. Berbeda dengan skenario dinamis, akurasi pada skenario statis ini cenderung stabil dan konsisten. Hasil yang lebih lengkap mengenai tren akurasi agregat dapat dilihat pada kurva yang disajikan pada Gambar 12.



Gambar 8. Hubungan Antara Kondisi Real dengan Sistem pada Kelas Statis

Hal ini terjadi karena pada skenario statis seperti yang ditampilkan pada gambar 12, jumlah orang di dalam ruangan cenderung tidak banyak berubah dari menit ke menit. Akibatnya, nilai rata-rata selama 10 menit yang dihitung oleh sistem akan sangat mendekati atau bahkan sama dengan nilai instan yang dicatat secara manual pada akhir interval tersebut. Hasil ini membuktikan bahwa pada kondisi yang stabil, akurasi sistem dalam merepresentasikan jumlah orang yang sebenarnya sangat tinggi, serta tren umum dari kurva kondisi sistem dapat mengikuti kondisi *real* jauh lebih baik dibandingkan pada skenario kelas dinamis.

Pengujian Sistem Kontrol Terintegrasi

Pengujian ini bertujuan untuk memverifikasi respons fungsional dari dua aktuator yang dikontrol oleh sistem, yaitu lampu dan kipas. Verifikasi dilakukan untuk



memastikan kedua aktuator beroperasi sesuai dengan algoritma yang telah ditentukan berdasarkan input dari sensor.

Tabel 7. Verifikasi Aksi Sistem

Skenario Uji	Jumlah Orang	Suhu (°C)	Status Lampu (Seharusnya)	Status Kipas (Seharusnya)	Hasil
Tidak Ada Orang, Suhu Normal	0	28.5°C	MATI	MATI	Sesuai
Tidak Ada Orang, Suhu Panas	0	32.0°C	MATI	MATI	Sesuai
Ada Orang, Suhu Normal	>0	28.5°C	NYALA	MATI	Sesuai
Ada Orang, Suhu Panas	>0	32.0°C	NYALA	NYALA	Sesuai

Hasil pengujian yang disajikan pada Tabel 7 menunjukkan bahwa logika kontrol sistem telah berjalan dengan benar. Sebagai contoh, sistem berhasil menjaga kipas tetap nonaktif saat suhu ruangan melebihi ambang batas namun tidak ada orang terdeteksi. Demikian pula, aktuator lampu dan kipas beroperasi secara akurat sesuai dengan kondisi kehadiran orang dan suhu yang ada. Keberhasilan pada seluruh skenario uji ini membuktikan bahwa sistem dapat mengontrol aktuator secara andal.

Pengujian Pengaruh Kualitas Jaringan terhadap *Latency* Sistem

Tabel 8. Hasil Pengujian Latency Sistem pada Berbagai Kualitas Jaringan

No	Kondisi Jaringan	Ping	Estimasi Latency Total Sistem
1	Wifi Sinyal Kuat	98ms	2.0-3.5 detik
2	Wifi Sinyal Sedang	200ms	4.5-6.2 detik
3	Wifi Sinyal Lemah	400ms	10.5-15.4 detik

Hasil pengujian pada Tabel 8 menunjukkan korelasi positif antara ping dengan estimasi total latensi sistem. Pada kondisi jaringan bersinyal kuat dengan ping 98 ms, total latensi sistem berkisar antara 1,2–2,5 detik. Performa ini tergolong cukup baik dan andal. Sebaliknya, saat sinyal melemah dan ping mencapai 400 ms, total latensi meningkat drastis hingga 10,5–15,4 detik. Hal ini menyebabkan sistem menjadi sangat lambat, tidak responsif, dan tidak efektif.

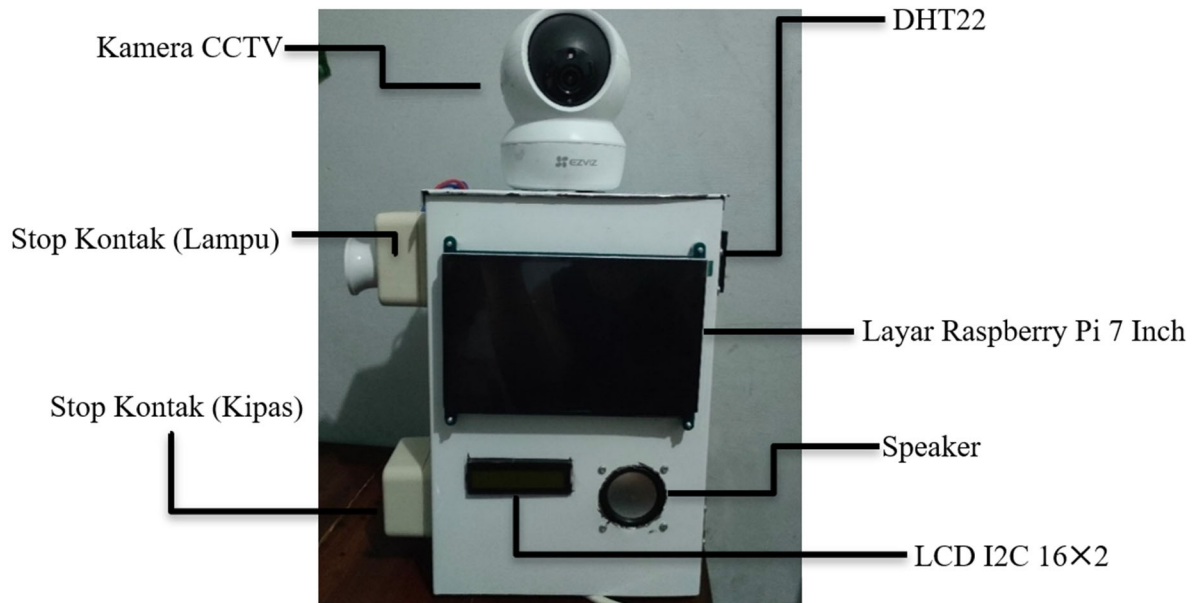
PEMBAHASAN

Pengujian sistem Monitoring jumlah mahasiswa di dalam ruang kelas dilakukan dengan menjalankan perangkat pada ruang kelas secara langsung, saat dilakukan



pengujian alat terdapat data manual yang dicatat persepuluh menitnya, yang kemudian hasil data tersebut dibandingkan dengan hasil data yang dihasilkan oleh sistem. Adapun beberapa parameter seperti sensor suhu DHT22 serta *relay*. *User Interface* dari sistem sendiri berupa website yang akan menampilkan data pendeteksian serta status sensor dan aktuator yang diberikan secara *real time*.

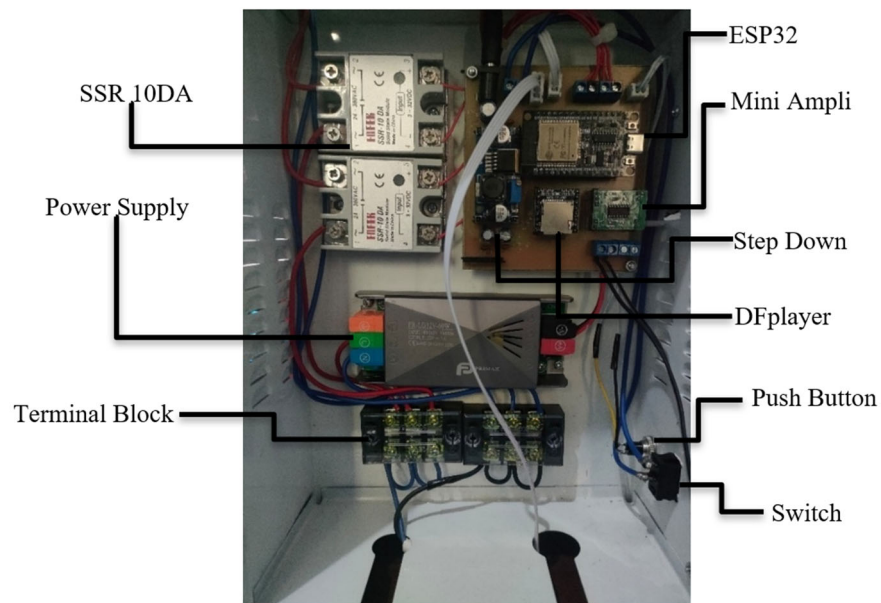
Perancangan Hardware dan Elektronika



Gambar 9. Realisasi Alat Berdasarkan Desain

Gambar 8 merupakan realisasi dari desain yang telah dibuat sebelumnya. Alat ini menggunakan kotak panel berbahan paduan aluminium dengan dimensi 28 x 19 x 10 cm. Selain itu, terdapat CCTV yang berfungsi sebagai kamera untuk menangkap gambar. Proses pemrosesan gambar selanjutnya dilakukan oleh laptop dengan memanfaatkan GPU untuk mengeksekusi program pendeteksian. Laptop akan diletakkan berdekatan dengan kotak panel, sedangkan CCTV dapat diposisikan sesuai kebutuhan ruangan. Peletakan CCTV ini fleksibel karena tidak terhubung secara langsung ke laptop maupun kotak panel.





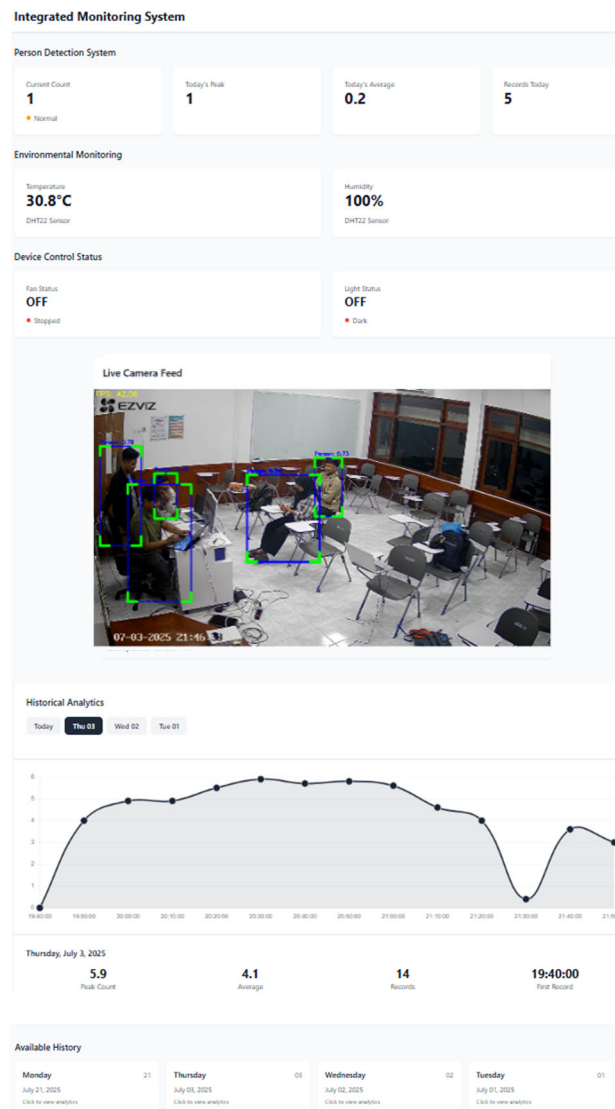
Gambar 10. Sistem Elektronika Alat

Gambar 9 menunjukkan diagram sistem elektronika yang terpasang di dalam kotak panel. Sistem ini menggunakan sebuah *terminal block* untuk mendistribusikan jalur tegangan AC ke dua unit *Solid State Relay* (SSR) dan satu *Power Supply Unit* (PSU). Selanjutnya, PSU menyuplai daya ke rangkaian PCB utama. Sebelum daya diterima oleh PCB, sebuah modul *step-down* digunakan untuk menurunkan tegangan dari 12V menjadi 5V. Rangkaian ini juga dilengkapi beberapa konektor T-Block dan Molex untuk menghubungkan berbagai sensor dan aktuator. Untuk komunikasi data, ESP32 terhubung dengan laptop melalui koneksi serial. Koneksi ini juga berfungsi sebagai sumber daya untuk ESP32, sehingga dayanya disuplai langsung dari laptop, bukan dari PSU utama.

Perancangan Software dan *User Interface*

Pengembangan perangkat lunak untuk alat ini melibatkan dua lingkungan pengembangan utama: Arduino IDE dan Visual Studio Code. Arduino IDE digunakan untuk menulis, memverifikasi, dan mengunggah program ke *single-board microcontroller* ESP32. Kode yang ditulis di dalamnya berfungsi untuk mengimplementasikan algoritma pembacaan sensor, mengontrol aktuator, serta menerima data serial dari program Python. Program pada ESP32 ini juga terintegrasi dengan Firebase, di mana nilai sensor DHT22 dan status *Solid State Relay* (SSR) dikirim ke *Realtime Database* agar dapat dipantau melalui situs website.





Gambar 11. Tampilan UI Web

Sementara itu, Visual Studio Code (VS Code) digunakan sebagai *editor* untuk program utama berbasis Python. Dalam program ini, deteksi objek dikembangkan dengan memanfaatkan pustaka (*library*) Ultralytics (YOLO). Model deteksi yang digunakan adalah model standar yang telah dilatih pada *dataset* COCO (*Common Objects in Context*) dengan fokus hanya pada kelas *person*. Untuk menampilkan hasil deteksi pada situs website, digunakan kerangka kerja (*framework*) Flask. Flask berfungsi sebagai *web server* yang menjembatani program Python dengan *browser*, sehingga tangkapan kamera dan parameter deteksi dapat dipantau secara langsung. Selain itu, program Python ini juga terhubung ke Firebase untuk mencatat data historis, dengan mengirimkan jumlah deteksi setiap menit serta nilai rata-ratanya setiap sepuluh menit. alat ini akan menggunakan 2 fitur yang ditawarkan oleh Firebase yakni firestore untuk menyimpan



data pendeteksian secara *historical* lalu *Realtime Database* untuk menampilkan nilai sensor dan status rilai secara *real-time* pada website, jadi website nantinya akan mengambil data yang disimpan pada Firebase kemudian akan menampilkannya pada *UI* website.

Gambar 10 merupakan tampilan website, website ini dijalankan pada localhost sehingga orang yang dapat mengakses website ini hanyalah user yang terkoneksi dengan wifi yang sama pada sistem, pada website sendiri tertampil beberapa parameter seperti jumlah orang sekarang, rata-rata harian, orang terbanyak hari ini serta *record* data yang telah diambil, lalu adapun parameter seperti sensor DHT22 serta status *relay* apakah menyalakan atau mematikan perangkat, kemudian terdapat live camera feed dimana disini window kamera akan tertampil pada website dan adapun grafik yang menunjukkan jumlah rata rata pengunjung tiap 10 menit.

KESIMPULAN

Berdasarkan serangkaian pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem Monitoring jumlah orang berbasis model YOLOv10s dengan *Confidence Score* 0.4 dan pemrosesan GPU RTX 3050 TI telah berhasil diimplementasikan dengan kinerja yang sangat baik. Dalam pengujian akurasi, sistem menunjukkan performa yang berbeda pada dua skenario utama. Pada Skenario Statis, sistem mencapai tingkat akurasi yang sangat tinggi dan stabil, di mana hasilnya konsisten di atas 90%. Sebaliknya, pada Skenario Dinamis terdapat lonjakan naik-turun yang tidak stabil. Hal ini disebabkan oleh faktor tantangan seperti pergerakan subjek yang konstan dan potensi oklusi (saling menutupi). Meskipun demikian, hasil deteksi pada kedua skenario tetap menunjukkan korelasi yang kuat dengan pencatatan manual. Dari sisi fungsionalitas kontrol, sistem terbukti responsif dan andal. Aktuator lampu dan kipas angin berhasil beroperasi secara otomatis sesuai dengan logika yang ditetapkan, yang mengintegrasikan data kehadiran orang dengan data suhu dari sensor DHT. Secara keseluruhan, sistem telah bekerja secara efektif dan *real-time*. Walaupun demikian, kinerja sistem dipengaruhi oleh beberapa keterbatasan, yaitu adanya kesalahan deteksi minor serta instabilitas jaringan. Instabilitas ini menyebabkan fluktuasi *ping* yang berdampak langsung pada peningkatan total latensi sistem. Meskipun menghadapi tantangan tersebut, performa sistem secara keseluruhan telah bekerja secara *real-time* dan memenuhi tujuan utama penelitian.

UCAPAN TERIMA KASIH

Alhamdulillahirrabbi'lalamin. Puji syukur penulis panjatkan kehadirat Allah SWT yang senantiasa melimpahkan rahmat, karunia, dan hidayah-Nya, sehingga penulis dapat menyelesaikan penelitian dan penyusunan laporan ini dengan baik. Shalawat serta salam semoga senantiasa tercurahkan kepada junjungan kita, Nabi Muhammad SAW, beserta



keluarga, sahabat, dan para pengikutnya hingga akhir zaman. Dalam proses penyelesaian penelitian ini, penulis menyadari telah menerima banyak bimbingan, bantuan, dan dukungan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati, penulis ingin menyampaikan ucapan terima kasih yang tulus kepada:

1. Kedua orang tua tercinta serta saudara dan saudari saya, yang senantiasa memberikan doa yang tak pernah putus, kasih sayang, dan dukungan selama proses pengerjaan tugas akhir ini.
2. Bapak Dedi Ary Prasetya, S.T., M.Eng., selaku Dosen Pembimbing, saya berterima kasih yang sebesar-besarnya atas bimbingan, arahan, kesabaran, dan masukan kritis yang tak ternilai selama proses pengerjaan tugas akhir ini.
3. Seluruh Bapak dan Ibu Dosen Program Studi Teknik Elektro Universitas Muhammadiyah Surakarta, terima kasih atas bekal ilmu pengetahuan yang telah dibagikan dan menjadi dasar yang kokoh selama masa perkuliahan.
4. Tim Robotika (KRAI) Universitas Muhammadiyah Surakarta, terima kasih atas kesempatan dan bantuannya dalam memfasilitasi proses pengujian fungsionalitas alat penelitian ini.
5. Teman-teman saya terkhusus Ahmad, Catur, Fakhir, Harry, Aziz yang telah membantu dan menemani saya selama proses penelitian tugas akhir ini.
6. Seluruh pihak lain yang tidak dapat saya sebutkan satu per satu, yang telah turut serta memberikan bantuan dan mendukung kelancaran penyusunan tugas akhir ini.

DAFTAR PUSTAKA

- [1] D. Nurlaili, J. Maulindar, and D. Hartanti, "Monitoring Jumlah Orang Di Ruangan Menggunakan Teknologi IOT," vol. 12, no. 1, pp. 119–129, 2025.
- [2] A. R. Kurniawan, Z. Burdiarso, E. Ardhianto, and T. Informatika, "Implementasi Algoritma YOLO Untuk Deteksi Jumlah Pengunjung Didalam Gedung," *JATI (Jurnal Mhs. Tek. Inform.,* vol. 9, no. 2, pp. 2944–2948, 2025.
- [3] A. Rachman, Z. Arifin, and M. Septya, "Sistem Pengendali Suhu Ruangan Berbasis Internet of Things (IoT) Menggunakan Air Conditioner (AC) Dan NodeMCU V3 ESP82," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.,* vol. 5, no. 1, pp. 19–23, 2020.
- [4] T. Setiadi, L. Haidar, and M. Fadlan, "Prototype Rancang Bangun Sistem Lampu Dan Kipas Otomatis Menggunakan Sensor Pir Gerak Suhu Dan Cahaya Berbasis Arduino Uno," *J. Comput. Sci. Technol.,* vol. 2, no. 2, pp. 31–39, 2022, doi: 10.54840/jcstech.v2i2.40.
- [5] B. Putra, G. Pamungkas, B. Nugroho, and F. Anggraeny, "Deteksi dan Menghitung Manusia Menggunakan YOLO-CNN," *J. Inform. dan Sist. Inf.,* vol. 02, no. 1, pp. 67–76, 2021.



- [6] F. M. S. Nursuwars, R. A. Alauddin, and I. Usrah, "Sistem Penghitung Otomatis Jumlah Orang Dalam Ruangan Berbasis Internet of Things," *E-JOINT (Electronica Electr. J. Innov. Technol.*, vol. 5, no. 1, pp. 1–5, 2024, doi: 10.35970/e-joint.v5i1.2364.
- [7] I. P. S. Yoga, G. Sukadarmika, R. S. Hartati, and Y. Divayana, "Pendeteksi Jumlah Orang pada Sistem Bangunan Pintar Menggunakan Algoritma You Only Look Once," *Maj. Ilm. Teknol. Elektro*, vol. 22, no. 1, p. 11, 2023, doi: 10.24843/mite.2023.v22i01.p02.

