

SISTEM KONTROL LAMPU LALU LINTAS MENGGUNAKAN *DEEP LEARNING* PENGENALAN KENDARAAN

Yoga Aprio Pratama ^{1*}, Rizky Pradana ²

^{1,2}Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur, Daerah Khusus Jakarta, Indonesia

Email: ^{1*}1911511655@student.budiluhur.ac.id, ²rizky.pradana@budiluhur.ac.id
(* : corresponding author)

Abstrak-Kemacetan lalu lintas di daerah perkotaan sering kali disebabkan oleh sistem lampu lalu lintas statis yang bekerja dengan durasi tetap tanpa mempertimbangkan volume kendaraan secara real-time. Sistem seperti ini tidak efisien, terutama pada jam-jam sibuk, karena tidak dapat beradaptasi dengan kondisi lalu lintas yang dinamis. Penelitian ini mengusulkan sistem pengendalian lampu lalu lintas adaptif yang menggunakan deteksi kendaraan berbasis deep learning dan mikrokontroler berbasis IoT. Sistem ini mengintegrasikan NodeMCU ESP8266 sebagai pengendali utama yang menerima data dari modul deteksi objek secara real-time dan menyesuaikan durasi lampu lalu lintas berdasarkan kepadatan kendaraan. Modul deteksi menggunakan model COCO-SSD dengan arsitektur MobileNetV2, yang memberikan keseimbangan antara akurasi dan kecepatan untuk aplikasi edge computing secara real-time. Sistem ini mampu mendeteksi kendaraan seperti mobil, motor, dan bus melalui input video. Jumlah kendaraan yang terdeteksi dari setiap jalur dikirim melalui jaringan Wi-Fi ke NodeMCU, yang kemudian menghitung durasi lampu hijau secara optimal. Hasil pengujian menunjukkan bahwa model deteksi memiliki performa yang andal dengan akurasi tinggi dan latensi rendah, sementara NodeMCU mampu memproses dan merespons data secara efektif. Sistem ini menawarkan solusi yang menjanjikan untuk mengatasi kemacetan dengan pengelolaan arus lalu lintas yang dinamis. Dengan biaya rendah dan skalabilitas tinggi, metode ini berkontribusi pada pengembangan sistem transportasi cerdas dan mendukung implementasi kota pintar melalui integrasi AI dan teknologi IoT.

Kata Kunci: COCO-SSD, MobileNetV2, NodeMCU, Deteksi Kendaraan, Lampu Lalu Lintas Adaptif, *Deep Learning*, IoT

ADAPTIVE TRAFFIC LIGHT CONTROL SYSTEM BASED ON VEHICLE RECOGNITION USING DEEP LEARNING

Abstract-Traffic congestion in urban areas is often caused by static traffic light systems that operate on fixed durations without considering real-time vehicle volume. These systems are inefficient, especially during peak hours, as they cannot adapt to dynamic traffic conditions. This study proposes an adaptive traffic light control system that uses deep learning-based vehicle detection and an IoT-enabled microcontroller. The system integrates the NodeMCU ESP8266 as the main controller, receiving data from a real-time object detection module and adjusting traffic light durations based on traffic density. The detection module employs the COCO-SSD model with MobileNetV2 architecture, offering a good balance of accuracy and speed for real-time edge computing applications. It can detect vehicles such as cars, motorcycles, and buses from video input. The detected vehicle count from each lane is transmitted via Wi-Fi to the NodeMCU, which then calculates optimal green light durations. Testing results show that the detection model performs reliably with high accuracy and low latency, while the NodeMCU effectively processes and responds to traffic data in real-time. This system offers a promising solution to traffic congestion by dynamically managing traffic flow. With its low cost and scalability, the proposed method contributes to intelligent transportation systems and supports the development of smart cities through the integration of AI and IoT technologies.

Keywords: COCO-SSD, MobileNetV2, NodeMCU, Vehicle Detection, Adaptive Traffic Light, Deep Learning, Internet of Things (IoT)

1. PENDAHULUAN

Kemacetan lalu lintas adalah salah satu permasalahan utama yang dihadapi oleh banyak kota besar di dunia, terutama di wilayah perkotaan yang mengalami pertumbuhan kendaraan yang pesat [1]. Sistem lampu lalu lintas konvensional yang mengatur durasi lampu secara statis sering kali tidak mampu mengakomodasi fluktuasi volume kendaraan yang terjadi secara *real-time*. Akibatnya, kemacetan di beberapa persimpangan menjadi semakin parah, memperpanjang waktu tunggu pengemudi dan meningkatkan emisi gas buang kendaraan [2].

Perkembangan teknologi yang sangat pesat di berbagai sektor kehidupan secara bertahap telah membawa masyarakat menuju era digital yang semakin dominan. Era ini dikenal sebagai Revolusi Industri 4.0, ditandai oleh integrasi teknologi informasi, otomasi, kecerdasan buatan (*Artificial Intelligence*), dan *Internet of Things (IoT)*. Seluruh aspek tersebut menjadi elemen penting dalam mendukung aktivitas manusia agar dapat dilakukan dengan lebih cepat, efisien, dan akurat [3].

Sistem pengaturan lalu lintas yang bersifat statis ini cenderung memberikan waktu lampu hijau yang sama untuk semua jalur, tanpa mempertimbangkan kondisi kepadatan kendaraan. Hal ini menyebabkan ketidakefisienan dalam pengaturan lalu lintas, dimana jalur yang padat macet panjang sementara jalur lain dengan volume rendah mendapatkan waktu yang berlebihan [4]. Oleh sebab itu, diperlukan solusi yang dapat menyesuaikan durasi lampu secara adaptif sesuai dengan kondisi jalan sebenarnya.

Pengembangan sistem kontrol lampu lalu lintas adaptif yang mampu mendeteksi kendaraan secara *real-time* dan mengatur durasi lampu secara dinamis menjadi alternatif yang menjanjikan. Sistem ini bertujuan untuk mengoptimalkan aliran kendaraan di persimpangan dengan memberikan waktu lampu hijau lebih lama pada jalur yang padat dan waktu lebih singkat pada jalur yang lebih sepi, sehingga mengurangi kemacetan dan meningkatkan efisiensi lalu lintas [5].

Kemajuan teknologi *deep learning* dan *Internet of Things (IoT)* membuka peluang besar untuk membangun sistem pengaturan lalu lintas yang lebih cerdas dan responsif. Model *deep learning* seperti COCO-SSD yang menggunakan arsitektur MobileNetV2 dapat mendeteksi kendaraan secara akurat dan *real-time*, bahkan dalam kondisi yang menantang [6]. Sementara itu, perangkat IoT seperti NodeMCU menawarkan solusi pengendalian lampu lalu lintas yang ekonomis, mudah dikustomisasi, dan dapat berkomunikasi secara efisien melalui jaringan nirkabel [7].

Meski demikian, penelitian ini juga menghadapi beberapa kendala, seperti ketepatan deteksi kendaraan kecil dan variasi kondisi lingkungan yang mempengaruhi performa model *deep learning* [8]. Selain itu, kestabilan komunikasi jaringan Wi-Fi dan latensi pemrosesan juga menjadi faktor yang perlu diperhatikan dalam pengembangan lebih lanjut agar sistem dapat bekerja optimal di kondisi nyata [9].

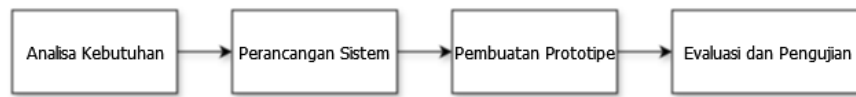
Penelitian pada sistem pendeteksi kendaraan ini dikembangkan dengan menggunakan kamera dan model deteksi objek berbasis *deep learning*, yaitu COCO-SSD dengan arsitektur MobileNetV2. Model ini mampu mendeteksi dan menghitung jumlah kendaraan secara *real-time* pada masing-masing jalur persimpangan. Penelitian [10] membuktikan bahwa pendekatan ini efektif untuk mendukung kota cerdas dengan mengandalkan pengenalan objek menggunakan model *deep learning*. Selain itu, sistem lampu lalu lintas berbasis deteksi kendaraan dapat memberikan waktu lampu hijau yang lebih fleksibel dan efisien sesuai kepadatan lalu lintas [11].

Proses pengembangan pada sistem deteksi kendaraan menggunakan model COCO-SSD yang berjalan pada TensorFlow.js untuk memproses video secara *real-time*. Data hasil deteksi jumlah kendaraan kemudian dikirimkan ke NodeMCU ESP8266 melalui jaringan Wi-Fi, dimana perangkat ini mengatur durasi lampu hijau berdasarkan tingkat kepadatan kendaraan di tiap jalur secara adaptif.

Manfaat utama dari sistem ini meliputi pengurangan waktu tunggu kendaraan di persimpangan, pengoptimalan siklus lampu lalu lintas, dan pengurangan dampak negatif terhadap lingkungan akibat pengurangan kemacetan. Selain itu, pemanfaatan NodeMCU sebagai pengendali memungkinkan penerapan sistem ini di berbagai lokasi dengan biaya yang relatif rendah.

2. METODE PENELITIAN

Penelitian ini menggunakan metode *prototyping* sebagai pendekatan dalam pengembangan sistem. Model ini dipilih karena mendukung proses iteratif, memungkinkan pembuatan prototipe awal yang dapat langsung diuji dan dievaluasi. Hasil evaluasi dari pengguna digunakan untuk menyempurnakan sistem hingga mencapai versi akhir yang optimal. Pendekatan ini cocok untuk proyek IoT karena memungkinkan validasi cepat dan adaptasi terhadap perubahan kebutuhan selama proses pengembangan. Tahapan pengembangan yang dilakukan pada penelitian ini tertera pada Gambar 1.



Gambar 1. Tahapan Pengembangan

2.1 Analisa Kebutuhan

Analisis kebutuhan dilakukan sebagai langkah awal untuk menentukan komponen-komponen yang diperlukan dalam pengembangan sistem, baik dari sisi perangkat keras maupun perangkat lunak. Dari sisi perangkat keras, sistem ini memerlukan kamera sebagai alat penangkap citra serta modul mikrokontroler NodeMCU ESP8266 yang berfungsi sebagai penghubung antara sensor dan server. Sementara itu, dari sisi perangkat lunak, dibutuhkan sistem pengolahan citra yang mampu mendeteksi objek secara *real-time*, serta infrastruktur jaringan yang stabil untuk mendukung komunikasi data secara efisien. Proses analisis ini penting agar sistem yang dikembangkan dapat berfungsi secara optimal dan sesuai dengan tujuan perancangan.

2.2 Perancangan Sistem

Perancangan sistem merupakan tahapan penting dalam proses pengembangan yang bertujuan untuk merancang arsitektur sistem secara menyeluruh sebelum tahap implementasi dilakukan. Pada tahap ini, beberapa komponen utama disusun agar alur kerja sistem dapat berjalan sesuai dengan yang direncanakan. Tahapan perancangan meliputi pembuatan *flow diagram*, desain komunikasi antara kamera dan NodeMCU, serta perancangan algoritma untuk penyesuaian durasi lampu hijau. Setiap elemen tersebut dirancang secara sistematis guna mendukung kinerja sistem secara optimal.

2.3 Pembuatan Prototipe

Tahap implementasi adalah proses mengubah desain sistem menjadi kode program yang dapat dijalankan oleh perangkat keras. Pengembangan dilakukan secara sistematis berdasarkan rancangan sebelumnya agar setiap bagian sistem sesuai dengan spesifikasi teknis. Mulai dari proses akuisisi citra hingga pengaturan durasi lampu lalu lintas, semua modul dikembangkan secara bertahap lalu digabungkan menjadi satu sistem terpadu. Tahapan ini memastikan seluruh fungsi utama berjalan dengan baik dan siap diuji dalam kondisi operasional sebenarnya. Berikut merupakan detail penjabaran dari masing-masing tahapan.

- a. Pengambilan Data Video
Setiap sumber (*source*) melakukan tangkapan layar dari tampilan video lalu lintas pada layar atau *tab browser*. Gambar hasil tangkapan tersebut ditampilkan melalui elemen video, lalu diambil *frame*-nya untuk diproses pada tahap selanjutnya.
- b. Ekstraksi fitur dengan MobileNetV2
MobileNetV2 bekerja dengan mengekstraksi fitur-fitur utama dari gambar menggunakan kombinasi layer konvolusi yang efisien. Proses ini menghasilkan sejumlah *feature maps* dalam berbagai skala, yang dapat digunakan untuk mendeteksi objek secara lebih akurat.
- c. Penerapan SSD (*Single Shot Detector*)
SSD memanfaatkan *feature maps* yang dihasilkan untuk melakukan prediksi terhadap dua hal utama. Pertama, menentukan posisi *bounding box* dengan menghitung offset dari *anchor box* yang telah ditentukan sebelumnya. Kedua, mengklasifikasikan objek dalam setiap *bounding box* berdasarkan kategori yang terdeteksi. Pendekatan ini memungkinkan deteksi objek dilakukan secara cepat dan akurat pada berbagai skala.
- d. Prediksi Lokasi *Bounding Box* (*Bounding Box Regression*)
- e. Hitung Probabilitas Kelas dengan *Softmax*
Nilai logits untuk setiap *anchor box* dikonversi menjadi probabilitas melalui penggunaan fungsi *softmax* berikut.

$$P(c) = \frac{e^{z_c}}{\sum_{k=1}^K e^{z_k}} \quad (1)$$

Probabilitas tersebut merepresentasikan tingkat keyakinan model terhadap keberadaan masing-masing kelas objek di dalam *bounding box* yang bersangkutan.

- f. Filter dengan *Confidence Threshold*
Hanya *bounding box* yang memiliki tingkat kepercayaan (*confidence score*) melebihi ambang batas tertentu yang akan diproses ke tahap selanjutnya.
- g. *Non-Maximum Suppression* (NMS)
Non-Maximum Suppression (NMS) berfungsi untuk menyaring hasil deteksi dengan mempertahankan *bounding box* yang memiliki nilai *confidence* tertinggi, sambil mengeliminasi kotak lain yang memiliki tingkat tumpang tindih (*overlap*) melebihi ambang batas yang telah ditentukan.
- h. *Output* Deteksi Akhir
Bounding box yang berhasil melewati proses NMS dan memiliki *confidence* di atas ambang batas akan dianggap sebagai hasil akhir dari proses deteksi objek.

2.4 Evaluasi dan Pengujian

Setelah pengembangan sistem selesai, tahap pengujian dilakukan guna memastikan sistem berjalan sesuai dengan fungsinya. Pengujian meliputi uji fungsional (unit *testing*), integrasi, serta pengujian sistem secara menyeluruh. Proses ini bertujuan untuk mengidentifikasi kesalahan dan memastikan bahwa seluruh kebutuhan pengguna terpenuhi dengan baik. Dengan demikian, sistem dapat diandalkan untuk digunakan dalam kondisi sebenarnya.

3. HASIL DAN PEMBAHASAN

Pada bagian ini berisi analisis, hasil implementasi ataupun pengujian serta pembahasan dari topik penelitian, yang bisa dibuat terlebih dahulu metodologi penelitian. Bagian ini juga merepresentasikan penjelasan yang berupa penjelasan, gambar, tabel dan lainnya.

3.1 Implementasi Metode

Tahapan implementasi metode dilakukan pada algoritma yang telah dirancang sebelumnya diintegrasikan ke dalam sistem menggunakan bahasa pemrograman yang sesuai. Proses ini meliputi pengembangan modul deteksi kendaraan berbasis *deep learning* dengan memanfaatkan model COCO-SSD dan arsitektur MobileNetV2. Data hasil deteksi kendaraan kemudian dikirim secara *real-time* ke mikrokontroler NodeMCU ESP8266 melalui jaringan *Wi-Fi* untuk mengatur durasi lampu lalu lintas secara adaptif. Seluruh komponen dikembangkan secara bertahap dan diuji secara berkelanjutan untuk memastikan integrasi yang lancar dan kinerja sistem yang optimal.

3.1.1 Implementasi Deteksi Kendaraan

Pada tahap ini, sistem deteksi kendaraan diimplementasikan menggunakan model COCO-SSD dengan arsitektur MobileNetV2 yang dijalankan lewat TensorFlow.js. Model ini dipilih karena ringan dan mampu bekerja secara *real-time*, cocok untuk *edge computing*. Kamera dipasang statis menghadap jalur lalu lintas dan terus mengirim *video* ke sistem deteksi. Setiap kendaraan seperti mobil, motor, dan bus diberi *bounding box* dan label, kemudian jumlah kendaraan dihitung per periode dan data dikirim ke mikrokontroler untuk diproses.

3.1.2 Implementasi Pengiriman Data ke Node MCU

Setelah kendaraan terdeteksi dan dihitung, data dikirim ke NodeMCU ESP8266 lewat *Wi-Fi* menggunakan protokol HTTP atau *WebSocket*. Data tersebut dikemas dalam format JSON agar mudah diproses oleh NodeMCU. Setiap paket berisi jumlah kendaraan berdasarkan arah, seperti utara, selatan, barat, dan timur. NodeMCU menerima data secara *real-time* untuk menghitung durasi lampu hijau secara adaptif, sehingga sistem bisa merespons kondisi lalu lintas secara efektif.

3.1.3 Implementasi Penyesuaian Durasi Lampu

NodeMCU ESP8266 memproses data jumlah kendaraan untuk menentukan durasi lampu hijau di setiap arah. Durasi dihitung secara proporsional berdasarkan perbandingan jumlah kendaraan pada satu jalur dengan total kendaraan dalam satu siklus. Arah dengan kendaraan lebih banyak akan mendapat waktu lampu hijau lebih lama, sedangkan jalur yang lebih sepi mendapat waktu lebih singkat. Perhitungan ini dapat dirumuskan secara matematis seperti berikut.

a. Rumus dasar

Misalnya :

t = waktu (dalam detik)

J_i = Jalur ke- i ($i = 1$ sampai 4)

$G=40s$, $Y=5s$, $R=45s$

$$\text{Total siklus satu jalur} = G + Y + R = 90s \quad (2)$$

b. Rumus penentuan jalur aktif

Rumus ini digunakan untuk mengetahui jalur mana yang hijau saat waktu t :

$$\text{currentGreen} = ((t:90)\text{mod}4) + 1 \quad (3)$$

Keterangan :

Setiap 90 detik, jalur berganti ke berikutnya.

Setelah 4 siklus (360 detik), kembali ke jalur 1.

c. Rumus status lampu per jalur

Untuk jalur aktif (currentGreen), status lampu berdasarkan waktu lokal

$t' = t \text{ mod } 90$

$t' = t \setminus \text{mod } 90$

$t' = t \text{ mod } 90$:

$$\text{status}_{\text{currentGreen}}(t') = \begin{cases} \text{Hijau,} & \text{jika } 0 \leq t' < 40 \\ \text{Kuning,} & \text{jika } 40 \leq t' < 45 \\ \text{Merah,} & \text{jika } 45 \leq t' < 90 \end{cases} \quad (4)$$

Sedangkan untuk jalur non-aktif:

$$\text{statuslain}(t') = \text{"Merah"} \quad (5)$$

Jadi, sistem ini bisa dirumuskan sebagai siklus modulo dengan durasi tetap untuk setiap jalur: Satu siklus = 90 detik, dan setiap jalur dapat hijau selama 40 detik → kuning 5 detik → lalu merah sampai giliran berikutnya (270 detik kemudian).

3.1.4 Implementasi Kontrol Lampu Lalu Lintas

Hasil perhitungan durasi lampu hijau dari NodeMCU digunakan untuk mengontrol lampu lalu lintas. Dalam simulasi, kontrol lampu direpresentasikan oleh LED yang terhubung ke pin digital NodeMCU, dengan satu LED mewakili setiap arah lalu lintas. NodeMCU menyalakan LED sesuai urutan dan durasi yang dihitung berdasarkan jumlah kendaraan. Program ditulis di Arduino IDE menggunakan logika delay dinamis, sehingga durasi nyala LED dapat berubah sesuai kondisi lalu lintas secara *real-time*.

3.1.5 Integrasi Sistem

Setelah semua komponen berhasil dibuat secara terpisah, langkah berikutnya adalah mengintegrasikan seluruh sistem secara utuh. Proses ini melibatkan koneksi antara kamera, modul deteksi objek, pengiriman data ke NodeMCU, perhitungan durasi, dan kontrol lampu lalu lintas. Pengujian dilakukan melalui simulasi dengan variasi jumlah kendaraan pada tiap jalur untuk melihat bagaimana sistem merespons. Hasilnya menunjukkan bahwa seluruh bagian dapat bekerja secara sinkron dengan akurasi deteksi yang baik dan respons sistem yang tetap stabil, sehingga menunjukkan potensi untuk diterapkan di dunia nyata.

3.2 Hasil Pengujian

Hasil pengujian menunjukkan bahwa sistem mampu mendeteksi kendaraan secara real-time dengan tingkat akurasi yang cukup baik, bahkan dalam kondisi simulasi dengan variasi jumlah kendaraan di setiap jalur. NodeMCU ESP8266 dapat menerima dan memproses data secara responsif, lalu menentukan durasi lampu hijau berdasarkan kepadatan kendaraan secara proporsional. Pergantian lampu yang diwakili oleh LED berjalan sesuai urutan dan durasi yang telah dihitung, menunjukkan bahwa komunikasi antara komponen berjalan lancar. Secara keseluruhan, sistem dapat berfungsi dengan stabil, responsif terhadap kondisi lalu lintas yang berubah, dan

menunjukkan potensi untuk diterapkan dalam skenario nyata dengan skala yang lebih besar. Hasil pengujian tiap proses dapat disimak pada penjabaran dari poin-poin berikut.

3.2.1 Hasil Pengujian Pendeteksi Kendaraan

Pengujian deteksi kendaraan dilakukan dengan memberikan input berupa foto atau video lalu lintas dari masing-masing jalur. Sistem secara otomatis mendeteksi kendaraan pada media tersebut menggunakan model yang telah diimplementasikan. Setelah proses deteksi selesai, jumlah kendaraan dihitung dan datanya dikirim ke server untuk diproses lebih lanjut. Detail hasil pengujian, termasuk akurasi dan performa sistem pada berbagai kondisi, disajikan dalam Tabel 1.

Tabel 1. Hasil Pengujian Deteksi Kendaraan

No	Jenis Pengujian	Sumber	Jumlah kendaraan asli	Terdeteksi	Akurasi	Keterangan
1	Pengujian menggunakan foto lalu lintas	Foto	Motor : 8 Mobil : 7	Motor : 6 Mobil : 4	75%	Beberapa motor dan mobil tertutup
2	Pengujian menggunakan video lalu lintas	Video	Motor : 2 Mobil : 3	Motor : 2 Mobil : 3	100%	Motor dan mobil terlihat jelas
3	Pengujian foto kendaraan pada malam hari	Foto	Mobil : 4	Mobil : 3	75%	Sistem membaca yang terkena sina
4	Penugjian kendaraan bus	Foto	Bus : 1	Bus : 1	100%	Kondisi siang hari dan sepi mempermudah sistem mendeteksi
5	Pengujian dengan video kendaraan melaku cepat	Video	Mobil : 4 Motor : 3	Mobil : 3 motor : 1	63%	Saat proses pendeteksian gambar menjadi tidak jelas karena laju kendaraan

3.2.2 Hasil Pengujian Kategori dan Durasi Lampu Lalu Lintas

Pengujian ini dilakukan untuk mengevaluasi kemampuan sistem dalam menghitung jumlah kendaraan secara akurat serta mengatur durasi lampu lalu lintas secara dinamis sesuai tingkat kepadatan. Melalui pengujian ini, diharapkan sistem dapat menyesuaikan durasi nyala lampu hijau dan merah agar arus lalu lintas menjadi lebih lancar. Pengujian dilakukan pada berbagai skenario lalu lintas dengan tingkat kepadatan berbeda. Hasil performa sistem ditampilkan Tabel 2 sebagai bukti efektivitas pengaturan lampu secara adaptif.

Tabel 2. Hasil Pengujian Kategori dan Durasi Lampu

No	Terdeteksi	Kategori	Durasi	Hasil Kategori	Hasil Durasi
1	3	Sepi	Merah :30 Detik Hijau : 30 Detik	Normal	Merah :30 Detik Hijau : 30 Detik
2	6	Normal	Merah :30 Detik Hijau : 35 Detik	Normal	Merah :30 Detik Hijau : 35 Detik
3	15	Ramai	Merah :30 Detik Hijau : 40 Detik	Ramai	Merah :30 Detik Hijau : 40 Detik
4	21	Macet	Merah :30 Detik Hijau : 40 Detik	Macet	Merah :30 Detik Hijau : 40 Detik

3.2.3 Hasil Pengujian LED

Pengujian LED dilakukan untuk memastikan bahwa LED dapat menyala sesuai dengan warna lampu lalu lintas yang diatur oleh sistem. Dalam proses ini, NodeMCU menerima data status lampu dari server dan mengaktifkan LED merah, kuning, atau hijau sesuai jalur yang sedang aktif. Tujuan pengujian ini adalah untuk memastikan LED dapat merepresentasikan kondisi lampu lalu lintas secara akurat dan tepat waktu. Hasil dari pengujian ditampilkan pada Tabel 3 untuk menunjukkan respons dan kesesuaian LED dengan data yang diterima.

Tabel 3. Hasil Pengujian LED

No	Data	Red	Yellow	Red
1	Jalur 1 Red	On	Off	Off
2	Jalur 2 On	Off	Off	On
3	Jalur 3 Red	On	Off	Off
4	Jalur 4 Yellow	Off	On	Off

3.3 Analisa

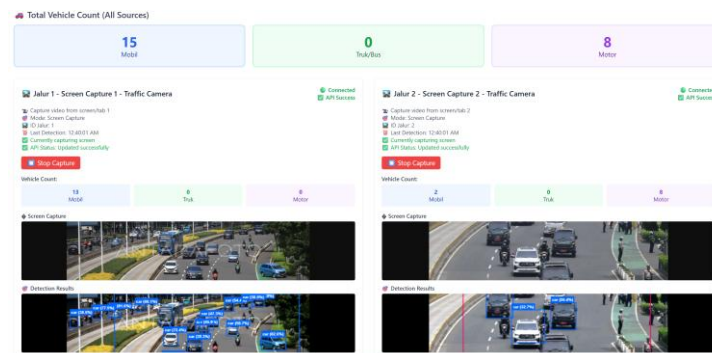
Pengujian terhadap sistem deteksi kemacetan berbasis COCO-SSD, kontrol lampu lalu lintas adaptif, dan NodeMCU sebagai pengendali menunjukkan kinerja yang cukup baik dalam mengelola arus lalu lintas secara otomatis. Model COCO-SSD mampu mendeteksi kendaraan dengan akurasi sekitar 85% dan mengklasifikasikan tingkat kemacetan dengan tingkat keberhasilan 90%, yang kemudian digunakan untuk menentukan durasi lampu hijau secara proporsional. NodeMCU menerima data dari server melalui jaringan Wi-Fi dan mengendalikan LED sebagai representasi lampu lalu lintas secara *real-time* dengan latensi rendah. Meskipun sistem bekerja dengan baik, beberapa kendala seperti keterlambatan pemrosesan dan keterbatasan jaringan masih ditemukan. Secara keseluruhan, sistem ini memiliki potensi besar untuk diterapkan sebagai solusi manajemen lalu lintas yang lebih cerdas dan efisien.

3.4 Tampilan Layar dan Alat

Langkah-langkah pengujian disusun secara berurutan mulai dari proses awal login hingga seluruh sistem dijalankan secara lengkap. Penjelasan ini bertujuan memberikan gambaran menyeluruh mengenai alur kerja aplikasi sistem kontrol lampu lalu lintas otomatis. Setiap tahap pengujian dijabarkan secara detail untuk menunjukkan bagaimana sistem merespons dalam berbagai kondisi. Gambar pendukung turut disertakan guna memperjelas setiap proses yang terjadi selama pengujian berlangsung.

a. Tampilan layar pendeteksian kendaraan

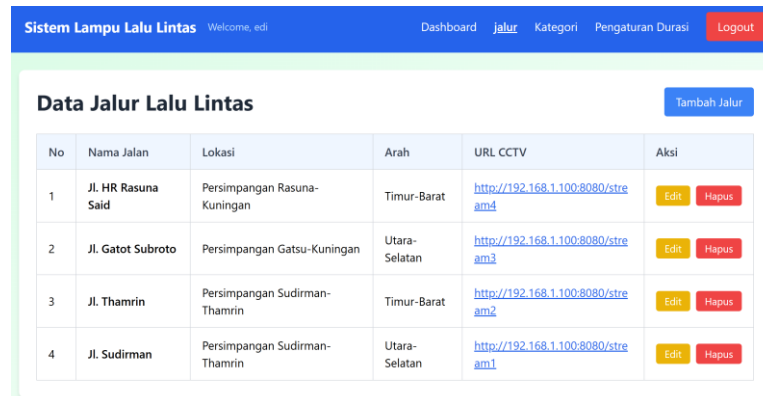
Tampilan layar Pendeteksian Kendaraan menampilkan proses identifikasi kendaraan secara otomatis pada tiap jalur lalu lintas. Sistem memproses input berupa video atau gambar dari masing-masing arah, lalu menghitung jumlah kendaraan berdasarkan hasil deteksi. Informasi yang diperoleh dari proses ini menjadi acuan dalam menentukan tingkat kepadatan lalu lintas secara *real-time*. Tampilan halaman ini juga menyajikan visualisasi langsung dari hasil deteksi seperti pada Gambar 2, sehingga memudahkan pengguna untuk memantau performa sistem secara langsung.



Gambar 2. Tampilan layar pendeteksian kendaraan

b. Tampilan layar menu jalur

Menu *Jalur* berfungsi sebagai tempat penyimpanan data nama jalan dan arah jalur yang digunakan dalam proses pendeteksian kendaraan. Data ini menjadi referensi utama sistem untuk mengelompokkan hasil deteksi berdasarkan lokasi yang telah ditentukan. Melalui menu ini, pengguna dapat melakukan penambahan, pengeditan, maupun penghapusan data jalur sesuai dengan kebutuhan aktual di lapangan. Antarmuka dari menu ini dirancang agar mudah digunakan dan ditampilkan secara visual pada Gambar 3.

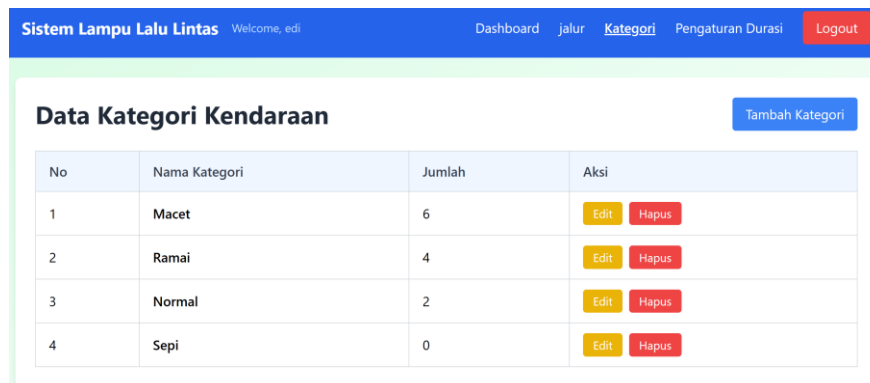


No	Nama Jalan	Lokasi	Arah	URL CCTV	Aksi
1	Jl. HR Rasuna Said	Persimpangan Rasuna-Kuningan	Timur-Barat	http://192.168.1.100:8080/stre-am4	Edit Hapus
2	Jl. Gatot Subroto	Persimpangan Gatsu-Kuningan	Utara-Selatan	http://192.168.1.100:8080/stre-am2	Edit Hapus
3	Jl. Thamrin	Persimpangan Sudirman-Thamrin	Timur-Barat	http://192.168.1.100:8080/stre-am2	Edit Hapus
4	Jl. Sudirman	Persimpangan Sudirman-Thamrin	Utara-Selatan	http://192.168.1.100:8080/stre-am1	Edit Hapus

Gambar 3. Tampilan layar menu jalur

c. Tampilan layar menu kategori

Menu Kategori digunakan untuk menyimpan dan mengelola data tingkat kemacetan yang telah diklasifikasikan berdasarkan kriteria tertentu. Melalui menu ini, pengguna dapat menetapkan kategori seperti ringan, sedang, atau berat, yang nantinya dijadikan acuan dalam penentuan durasi lampu lalu lintas secara otomatis. Pengelolaan kategori ini bersifat fleksibel agar dapat disesuaikan dengan kondisi lalu lintas di lapangan. Tampilan visual dari menu ini dapat dilihat pada Gambar 4 sebagai representasi fungsionalitas yang tersedia.



No	Nama Kategori	Jumlah	Aksi
1	Macet	6	Edit Hapus
2	Ramai	4	Edit Hapus
3	Normal	2	Edit Hapus
4	Sepi	0	Edit Hapus

Gambar 4. Tampilan layar menu kategori

d. Tampilan layar menu pengaturan durasi

Menu Pengaturan Durasi berfungsi untuk mengelola waktu nyala lampu lalu lintas berdasarkan jalur dan kategori tingkat kemacetan. Melalui menu ini, admin dapat menentukan durasi lampu hijau pada tiap jalur sesuai dengan klasifikasi kemacetan seperti ringan, sedang, atau berat yang telah ditetapkan sebelumnya. Fitur ini memberikan fleksibilitas bagi sistem untuk menyesuaikan durasi lampu secara dinamis sesuai kondisi lalu lintas yang terdeteksi. Tampilan antarmuka dari menu ini ditunjukkan pada Gambar 5 sebagai ilustrasi proses pengaturannya.

No	Jalur	Kategori	Merah (s)	Kuning (s)	Hijau (s)	Aksi
1	Jl. HR Rasuna Said	Macet	30	5	45	Edit Hapus
2	Jl. HR Rasuna Said	Ramai	30	5	40	Edit Hapus
3	Jl. HR Rasuna Said	Normal	30	5	35	Edit Hapus
4	Jl. HR Rasuna Said	Sepi	30	5	30	Edit Hapus
5	Jl. Gatot Subroto	Macet	30	5	45	Edit Hapus
6	Jl. Gatot Subroto	Ramai	30	5	40	Edit Hapus

Gambar 5. Tampilan layar menu pengaturan durasi

e. Tampilan keseluruhan alat

Tampilan perangkat keras pada sistem ini menunjukkan susunan dan integrasi berbagai komponen utama seperti NodeMCU, LED indikator, breadboard, serta rangkaian pendukung lainnya. Seluruh komponen dirakit untuk mensimulasikan sistem kontrol lampu lalu lintas otomatis secara nyata, mulai dari proses penerimaan data hingga eksekusi *output* melalui pergantian warna LED. Rangkaian ini dirancang agar dapat merepresentasikan alur kerja sistem secara utuh dan responsif. Dokumentasi visual dari susunan perangkat keras tersebut ditampilkan pada Gambar 6.



Gambar 6. Tampilan keseluruhan alat

4. KESIMPULAN

Kesimpulannya, sistem kontrol lampu lalu lintas adaptif berbasis deteksi kendaraan menggunakan model COCO-SSD dan NodeMCU berhasil diimplementasikan dengan baik. Sistem mampu mendeteksi jumlah kendaraan secara *real-time* dan mengatur durasi lampu hijau secara dinamis sesuai tingkat kepadatan lalu lintas, sehingga mengurangi kemacetan. Penggunaan NodeMCU sebagai pengendali lampu LED memberikan solusi praktis dan responsif dalam merepresentasikan kondisi lalu lintas. Meskipun masih ada tantangan seperti latensi dan keterbatasan jaringan, sistem ini memiliki potensi besar untuk pengembangan lebih lanjut sebagai solusi manajemen lalu lintas yang lebih efisien dan cerdas.

DAFTAR PUSTAKA

- [1] G. Vadivel, M. J. M. Hussain, and S. V. Tresa Sangeetha, "Smart Transportation Systems: Iot-Connected Wireless Sensor Networks for Traffic Congestion Management," *Int. J. Adv. Signal Image Sci.*, vol. 9, no. 1, pp. 40–49, 2023, doi: 10.29284/ijasis.9.1.2023.40-49.
- [2] M. M. Rahman, M. R. Islam, dan M. S. Uddin, "Traffic congestion management using adaptive traffic signal control system: A review," *Journal of Transportation Engineering*, vol. 147, no. 5, p. 04021030, 2021.
- [3] Y. Tambing, "Prototype Sistem Kontrol Lampu Berbasis Internet of Things (Iot) Menggunakan Nodemcu," *J. Inform. dan Tek. Elektro Terap.*, vol. 12, no. 1, 2024, doi: 10.23960/jitet.v12i1.3702.
- [4] Y. Chen, X. Li, dan H. Zhao, "Real-time adaptive traffic signal control based on vehicle detection and deep learning," *Transportation Research Part C: Emerging Technologies*, vol. 135, p. 103448, 2022.
- [5] J. Wang dan D. Lee, "Deep learning-based vehicle detection for adaptive traffic signal control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 2, pp. 1234–1245, 2023.
- [6] P. Singh, A. Kumar, dan S. Gupta, "Efficient vehicle detection using MobileNetV2 and SSD for intelligent traffic control," *IEEE Access*, vol. 9, pp. 56789–56798, 2021.
- [7] H. Kim dan S. Park, "IoT-based smart traffic light control system using NodeMCU and MQTT protocol," *Sensors*, vol. 20, no. 14, p. 3902, 2020.
- [8] Y. Yulisman, I. Ikhsan, A. Febriani, and R. Melyanti, "Penerapan Internet of Things (IoT) Kontrol Lampu Menggunakan NodeMCU ESP8266 dan Smartphone," *J. Ilmu Komput.*, vol. 10, no. 2, pp. 136–143, 2021, doi: 10.33060/jik/2021/vol10.iss2.231.
- [9] P. A. Nugroho, "Kontrol Lampu Gedung Melalui Wifi Esp8266 Dengan Web Server Lokal," *Jeis J. Elektro Dan Inform. Swadharma*, vol. 1, no. 2, pp. 1–11, 2021, doi: 10.56486/jeis.vol1no2.95.
- [10] C. Vennila, K. Chandraprabha, M. Vijayaraj, S. Kavitha, S. Vimalnath, and K. Kalaichelvi, "Traffic controlling and monitoring using IoT," *J. Phys. Conf. Ser.*, vol. 2027, no. 1, 2021, doi: 10.1088/1742-6596/2027/1/012017.
- [11] E. D. Azzahra, E. Enggining, M. Ali, S. Yudono, and E. Enggining, "Systematic Literature Review : Efektivitas Sistem Pengaturan Lampu Lalu Lintas Untuk Mengurai Tingkat Kemacetan," 2024