DESAIN MODEL KOMUNIKASI DATA MENGGUNAKAN HTTP DAN WEBSOCKET UNTUK SISTEM DETEKSI OTOMATIS BERBASIS ESP32-CAM PADA APLIKASI PENGAWASAN

Muhamad Nur Alvyangga Saputro, Kasiyanto
Teknik Elektronika Sistem Senjata, Politeknik Angkatan Darat
Jalan Anggrek No. 1 Pendem, Kec. Junrejo, Kota Batu, Jawa Timur, 65324, Indonesia
anggapoltekad2122@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem komunikasi data yang efisien antara perangkat ESP32-CAM, server web, dan browser pengguna dalam aplikasi pengawasan berbasis WebSocket. Sistem ini menggunakan protokol HTTP untuk mentransfer gambar dari ESP32-CAM ke server, sementara WebSocket digunakan untuk mengirimkan notifikasi deteksi objek secara langsung ke browser pengguna. Masalah utama yang dihadapi adalah keterbatasan pengiriman gambar melalui jaringan yang memengaruhi akurasi deteksi objek. Tujuan penelitian ini adalah untuk mengoptimalkan komunikasi data dan deteksi objek dengan memanfaatkan HTTP untuk pengiriman gambar dan WebSocket untuk notifikasi *real-time*. Metode yang digunakan dalam penelitian ini meliputi pengambilan gambar dengan ESP32-CAM, pengolahan gambar menggunakan YOLO untuk deteksi objek, serta pengiriman data melalui HTTP dan WebSocket. Hasil penelitian menunjukkan bahwa sistem yang dikembangkan berhasil mengatasi masalah latensi tinggi dengan memberikan notifikasi deteksi objek secara *real-time* melalui WebSocket, serta dapat mendeteksi objek dengan tingkat akurasi yang baik meskipun terdapat kendala pada pencahayaan yang kurang optimal. Sistem ini dapat beroperasi dengan baik dalam jaringan lokal tanpa ketergantungan pada internet, menjadikannya solusi yang efisien untuk aplikasi pengawasan.

Kata kunci : *ESP32-CAM, HTTP, WebSocket, YOLO*

1. PENDAHULUAN

Sistem pengawasan yang berbasis Internet of Things (IoT) saat ini semakin populer dalam berbagai aplikasi, salah satunya adalah dengan menggunakan perangkat ESP32-CAM [1]. Perangkat ini mampu menangkap gambar dan mengirimkannya melalui jaringan tanpa kabel [2] [3]. Untuk mendukung sistem pengawasan yang cepat dan efisien, diperlukan komunikasi data yang baik antara perangkat ESP32-CAM, server web, dan browser pengguna [4] [5]. Salah satu protokol yang digunakan untuk mentransfer gambar adalah HTTP. Konsep ini sederhana dan mudah diterapkan. Namun, pengiriman gambar melalui HTTP bisa terkadang lambat, terutama jika ukuran gambar besar atau jaringan tidak stabil [6].

Selain itu, untuk sistem pengawasan yang lebih responsif, diperlukan metode yang memungkinkan pengiriman notifikasi secara langsung saat objek terdeteksi. Di sini perlu adanya sebuah WebSocket, karena WebSocket memungkinkan komunikasi dua arah antara server dan browser tanpa perlu terus-menerus meminta data, sehingga proses pemberitahuan dapat lebih cepat. Dengan menggunakan WebSocket, notifikasi deteksi objek dapat langsung diterima oleh pengguna tanpa ketergantungan pada koneksi internet [4]. Fungsi komunikasi yang efisien ini sangat penting untuk pemantauan real-time, terutama pada sistem yang menggunakan jaringan lokal atau LAN [7].

Salah satu masalah utama dalam sistem pengawasan berbasis ESP32-CAM adalah keterbatasan dalam pengiriman gambar melalui jaringan lokal. Gambar yang diambil oleh ESP32-CAM cenderung memiliki ukuran besar, sehingga membutuhkan waktu lebih lama untuk dikirim melalui jaringan [8] [9]. Protokol HTTP yang digunakan untuk pengiriman gambar dapat menjadi lambat, terutama ketika jaringan lokal sedang sibuk atau banyak perangkat yang terhubung. Hal ini dapat menyebabkan keterlambatan dalam pengalaman *real-time* dengan kecepatan tinggi [10].

Masalah lainnya adalah dalam hal deteksi objek. Meskipun ESP32-CAM sudah dilengkapi dengan kemampuan untuk menangkap gambar, proses untuk mengenali objek dalam gambar tersebut membutuhkan pemrosesan yang cepat dan akurat. Deteksi objek dengan model seperti YOLO (You Only Look Once) perlu dipastikan dapat berjalan dengan baik meskipun kondisi pencahayaan kurang mendukung atau objek berada pada jarak jauh [11]. Selain itu, metode pengiriman informasi yang menggunakan sistem polling tradisional juga memperburuk masalah, karena informasi baru hanya dikirim setelah browser meminta data, yang dapat menyebabkan keterlambatan dalam pemberitahuan deteksi objek [12].

Tujuan utama dari penelitian ini adalah merancang dan membangun sistem komunikasi data yang efisien antara perangkat ESP32-CAM, server web, dan browser pengguna. Sistem ini bertujuan untuk memastikan gambar yang diambil oleh ESP32-CAM dapat dikirim ke server secara cepat melalui protokol HTTP, sementara notifikasi deteksi objek dapat langsung diterima oleh pengguna melalui WebSocket. Dengan menggunakan WebSocket,

sistem ini mampu memberikan pemberitahuan secara *real-time* tanpa perlu menunggu [13]. Konsep ini menjadi dasar dalam memahami bagaimana perubahan dalam pendapatan mempengaruhi aktivitas ekonomi, menjadikan proses pengawasan lebih responsif dan efisien.

Selain itu, penelitian ini bertujuan untuk mengintegrasikan deteksi objek otomatis menggunakan model YOLO. Dengan YOLO, gambar yang dikirimkan oleh ESP32-CAM dapat diproses secara otomatis untuk mendeteksi objek seperti manusia atau benda lainnya [4]. Deteksi objek yang cepat dan akurat ini akan meningkatkan nilai tambah pada sistem pengawasan, sehingga pengguna dapat memperoleh informasi yang lebih cepat dan akurat. Penelitian ini juga bertujuan untuk menguji kinerja sistem dalam jaringan lokal (LAN), memastikan bahwa sistem dapat berjalan tanpa bergantung pada koneksi internet dan tetap memberikan pengalaman pengawasan yang efektif dan stabil.

Untuk mengatasi masalah yang ada, penelitian ini menyarankan solusi yang menggunakan dua protokol komunikasi, yaitu HTTP dan WebSocket. Protokol HTTP digunakan untuk mengirim gambar dari ESP32-CAM ke server [8] [14]. Meskipun ukuran gambar yang dikirim cukup besar, HTTP dipilih karena mudah digunakan dan sudah banyak diterapkan di berbagai sistem. Namun, untuk notifikasi yang lebih cepat dan efisien [10], WebSocket digunakan untuk mengirimkan informasi deteksi objek secara langsung ke browser pengguna. Dengan menggunakan WebSocket, notifikasi menjadi lebih responsif dan cepat, sehingga meningkatkan kinerja sistem secara keseluruhan.

Kemudian penelitian ini juga akan mengintegrasikan deteksi objek otomatis menggunakan model YOLO. Dengan YOLO, gambar yang dikirimkan oleh ESP32-CAM akan diproses untuk mendeteksi objek seperti manusia atau benda lainnya. Hal ini akan membuat sistem pengawasan lebih cerdas dan mampu memberikan informasi secara lebih cepat kepada pengguna [11] [15]. Penelitian ini juga akan menguji sistem dalam jaringan lokal (LAN) tanpa bergantung pada internet, sehingga memastikan komunikasi tetap lancar dan cepat. Dengan solusi ini, diharapkan masalah seperti latensi tinggi dan pengiriman gambar yang lambat dapat diatasi secara efektif.

2. TINJAUAN PUSTAKA

Penelitian ini berfokus pada optimasi komunikasi data end-to-end antara perangkat ESP32-CAM, web server, dan browser pengguna-khususnya pada arsitektur pemrosesan tepi (edge) yang menekan latensi dan ketergantungan pada cloud dalam skenario jaringan lokal [8].

2.1. Penelitian terdahulu

M. Nguyen, L. Trinh, dan N. H. Do mengembangkan sistem deteksi gerakan real-time menggunakan ESP32-CAM dan sensor PIR untuk aplikasi keamanan rumah. Sistem ini memungkinkan deteksi pergerakan manusia secara otomatis dengan memanfaatkan perangkat ESP32-CAM dan sensor PIR, yang kemudian mengirimkan data ke mikrokontroler untuk diproses. Hasil penelitian menunjukkan bahwa sistem ini dapat mendeteksi gerakan dengan akurasi yang baik dalam berbagai kondisi lingkungan, serta dapat mengirimkan notifikasi secara real-time ke perangkat pengguna, meningkatkan efisiensi dalam pengawasan rumah.

Murley, Ma, Mason, dan Bailey membahas adopsi WebSocket dan lanskap komunikasi real-time dalam sistem IoT. Penelitian ini mengeksplorasi kemampuan WebSocket dalam meningkatkan efisiensi komunikasi real-time dengan mengurangi latensi dan memungkinkan pertukaran data dua arah yang lebih cepat. Hasil penelitian menunjukkan bahwa WebSocket sangat berguna dalam aplikasi IoT yang memerlukan komunikasi real-time, seperti sistem pengawasan berbasis ESP32-CAM, karena memiliki latensi rendah dan efisiensi komunikasi yang lebih baik dibandingkan metode polling konvensional.

H. J. Ochoa, R. Peña, Y. L. Mezquita, E. Gonzalez, dan S. Camacho-Leon melakukan analisis perbandingan antara WebSocket dan HTTP untuk komunikasi real-time dalam sistem rumah pintar. Penelitian ini membandingkan kedua protokol komunikasi tersebut untuk menentukan keunggulannya dalam pengiriman data real-time di sistem rumah pintar. Hasil penelitian menunjukkan bahwa WebSocket lebih efisien dalam komunikasi real-time dibandingkan HTTP, yang menghasilkan respons yang lebih cepat dan efektif dalam pengiriman informasi.

Y. Chang, F. Wu, dan H. Lin) merancang dan mengimplementasikan sistem komputasi edge berbasis ESP32 untuk deteksi objek menggunakan model YOLO. Penelitian ini membahas bagaimana ESP32 yang dilengkapi dengan kemampuan komputasi edge dapat memproses data visual menggunakan model YOLO secara real-time, memungkinkan deteksi objek secara cepat dan efisien tanpa bergantung pada server cloud. Sistem ini memberikan solusi efisien dalam pengawasan real-time serta meningkatkan kinerja deteksi objek dengan memanfaatkan pengolahan data langsung di perangkat.

2.2. ESP32-CAM dan Wifi pada Jaringan Lokal

ESP32-CAM banyak digunakan sebagai node akuisisi citra berbiaya rendah untuk pemantauan, tetapi keterbatasan komputasi dan memori membuat strategi sistem perlu diarahkan ke pembagian tugas: kamera fokus menangkap dan mengirim gambar, sementara komputasi berat (inferensi model) dilakukan di server lokal. Pendekatan ini umum dipilih

karena dapat mengurangi latensi dan beban bandwidth dibandingkan skenario yang bergantung pada layanan cloud [8], [7]. Dalam konteks penelitian ini, wifi dimaknai secara praktis antara lain dari seluruh alur (capture → transfer → inferensi → notifikasi) berjalan di Wi-Fi/LAN sehingga sistem tetap berfungsi walaupun tidak ada akses internet [7].

2.3. HTTP untuk Transfer Gambar dari ESP32-CAM ke Server Lokal

HTTP masih menjadi pilihan yang "langsung jadi" untuk pengiriman gambar karena kompatibel luas, sederhana diimplementasikan, dan mudah diintegrasikan dengan server aplikasi. Pada sistem IoT, pemilihan protokol aplikasi/transport biasanya ditentukan oleh tipe payload (gambar cenderung besar), pola komunikasi (unggah periodik/event-driven), dan biaya overhead [6]. Namun, HTTP memiliki overhead header dan mekanisme request–response yang dapat menambah konsumsi sumber daya pada perangkat terbatas. Studi pembandingan pada platform IoT menunjukkan bahwa HTTP dapat lebih boros energi dibanding protokol ringan tertentu pada skenario tertentu (misalnya dibanding MQTT), sehingga penelitian ini perlu menempatkan HTTP secara tepat: dipakai untuk transfer citra (payload besar), sementara kanal real-time dipisahkan agar tidak memaksa polling berulang [10].

2.4. WebSocket untuk Notifikasi Real-Time ke Browser

Kebutuhan real-time pada web muncul karena pola request–response tidak memungkinkan server "menyela" klien kapan saja. Alternatif seperti polling/long-polling menambah trafik dan permintaan yang sering sia-sia, sedangkan WebSocket menyediakan koneksi persisten dua arah dengan cara overhead pesan yang relatif kecil setelah handshake [4], [9]. Pada penelitian ini, WebSocket diposisikan bukan untuk memindahkan gambar (yang berat), tetapi untuk push notifikasi hasil deteksi (ringan), misalnya status "manusia terdeteksi/tidak", timestamp, nama file, dan skor keyakinan. Literatur juga menunjukkan pemanfaatan WebSockets untuk meningkatkan keandalan komunikasi dan mengatasi keterbatasan pola web service tradisional pada beban tertentu, sehingga relevan sebagai kanal notifikasi yang responsif di LAN [4].

2.5. YOLO sebagai Metode Deteksi Objek untuk Pengawasan

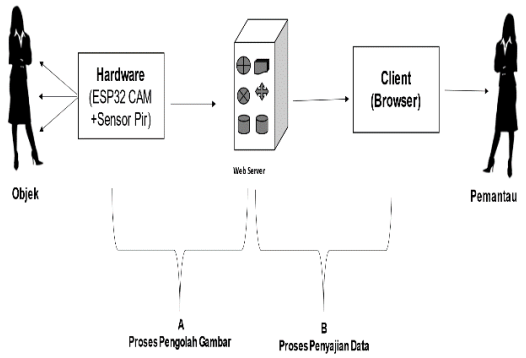
YOLO (You Only Look Once) dikenal luas karena kompromi yang kuat antara kecepatan dan akurasi untuk deteksi objek real-time. Tinjauan komprehensif YOLO menunjukkan evolusi berkelanjutan dari sisi arsitektur, efisiensi, dan performa yang membuat YOLO cocok untuk kebutuhan pengawasan [11]. Dalam ranah pengawasan video, variasi YOLO modern juga sering dipadukan dengan mekanisme perhatian (attention) atau transformer untuk meningkatkan ketahanan pada kondisi kompleks (misalnya occlusion atau clutter). Studi terbaru pada pengawasan real-time mengusulkan integrasi berbasis transformer-YOLOv8 dan melaporkan performa yang kompetitif, sehingga memperkuat relevansi pipeline deteksi "manusia vs bukan manusia" pada penelitian ini [12].

2.6. Model Komunikasi HTTP + WebSocket pada Sistem Tanpa Internet

Model yang paling masuk akal untuk masalah Anda adalah pemisahan kanal: HTTP untuk unggah citra (besar, tidak kontinyu) dan WebSocket untuk hasil deteksi (kecil, butuh cepat). Kajian protokol IoT menekankan bahwa pemetaan protokol ke kebutuhan aplikasi (payload size, latency target, skalabilitas) adalah kunci desain sistem komunikasi yang efisien [6]. Dengan batasan tanpa internet, desain LAN memberi konsekuensi metodologis penting, kemudian pengujian harus fokus pada latensi intra-jaringan, stabilitas Wi-Fi lokal, serta beban server saat inferensi [7], [8].

3. METODE PENELITIAN

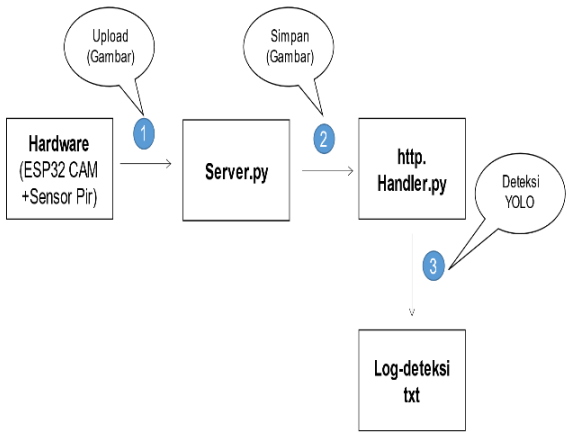
Penelitian ini bertujuan untuk menciptakan dan mengembangkan sistem komunikasi data yang efisien antar perangkat ESP32-CAM, server web, dan browser pengguna. Sistem yang dibuat akan mengelola pengambilan gambar, deteksi objek secara otomatis, serta pengiriman hasil deteksi kepada pengguna dalam waktu nyata. Secara keseluruhan, metode penelitian ini dibagi menjadi dua bagian utama yaitu proses pemrosesan gambar dan proses penyajian data.



Gambar 1. Desain alur keseluruhan

3.1. Proses pengolahan gambar

Proses pengolahan citra dalam sistem ini dimulai dengan pengambilan gambar oleh perangkat ESP32-CAM yang memiliki sensor PIR, kemudian dilanjutkan dengan pengolahan citra untuk identifikasi objek menggunakan model YOLO, serta penyimpanan hasil dari deteksi yang sudah dilakukan. Seluruh proses dapat dijelaskan sebagaimana tertuang dalam Gambar 2.



Gambar 2. Proses pengolahan gambar

Proses untuk memproses gambar diawali dengan penggunaan ESP32-CAM yang dilengkapi sensor PIR (Sensor Inframerah Pasif). Alat ini memiliki tugas utama untuk menangkap gambar objek yang terdeteksi dalam wilayah pemantauan. Sensor PIR berfungsi mendeteksi pergerakan objek, seperti manusia atau hewan, yang memasuki area yang dapat dijangkau oleh sensor tersebut. Ketika pergerakan terdeteksi oleh sensor PIR, ESP32-CAM secara otomatis merekam gambar objek yang ada. Setelah gambar diambil, foto tersebut lalu dikirim ke server melalui protokol HTTP. Pemilihan HTTP untuk mengirim gambar ini karena protokol tersebut sederhana untuk diterapkan dan bisa digunakan dengan banyak sistem lainnya. Protokol ini memungkinkan gambar yang ditangkap oleh ESP32-CAM dikirim dengan cepat ke server agar bisa diproses lebih lanjut.

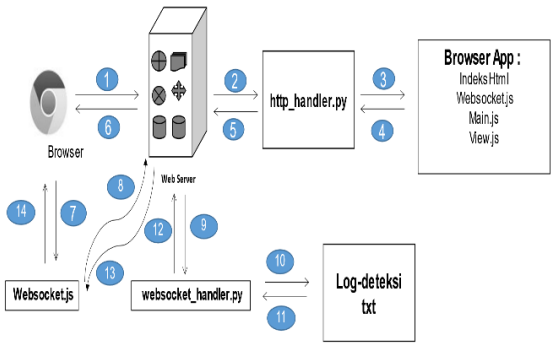
Setelah gambar diterima oleh server, langkah berikutnya adalah menyimpan gambar tersebut agar bisa diakses dan diproses oleh sistem. Server.py memiliki peran untuk mengelola proses penyimpanan ini, menjamin bahwa gambar yang diunggah dapat tersimpan dengan baik di folder yang telah ditentukan. Proses penyimpanan gambar ini sangat krusial untuk memastikan bahwa data gambar yang dikirimkan tetap ada dan dapat dimanfaatkan dalam proses pengolahan selanjutnya. Nama berkas gambar umumnya mencakup informasi penting seperti waktu pengambilan gambar atau ID dari perangkat yang mengirim, yang akan memudahkan dalam pelacakan serta pengolahan gambar di masa mendatang.

Gambar yang disimpan di server akan diproses lebih lanjut menggunakan skrip http. Handler.py. Skrip ini bertugas untuk mengambil gambar yang telah disimpan dan mengirimkannya ke model deteksi objek YOLO (You Only Look Once). YOLO dipilih karena kemampuannya mendeteksi objek secara *real-time* dengan cepat dan dengan akurasi yang memuaskan. Cara kerja model YOLO adalah dengan membagi gambar menjadi beberapa grid, lalu melakukan prediksi pada setiap grid guna mendeteksi objek tertentu, seperti manusia atau benda lainnya. Proses deteksi ini menghasilkan label untuk objek yang terdeteksi, seperti "manusia", serta koordinat bounding box yang menunjukkan posisi objek dalam gambar. YOLO juga memberikan skor keyakinan yang menilai seberapa pasti model terhadap objek yang terdeteksi.

Setelah penyelesaian proses identifikasi objek, hasil dari deteksi akan disimpan dalam sebuah file log yang dinamakan Log-deteksi. txt. File log ini berfungsi untuk merekam hasil identifikasi objek dengan cara yang terorganisir, mencakup informasi tentang objek yang dikenali, waktu pengenalan, dan tingkat kepercayaan yang dihasilkan oleh YOLO. Penyimpanan hasil identifikasi dalam bentuk log sangat krusial untuk memungkinkan analisis lebih dalam serta pendataan hasil yang telah diperoleh. Dengan adanya log ini, pengembang atau pengawas sistem dapat mengawasi efektivitas deteksi objek, menemukan masalah yang mungkin ada, dan melakukan perbaikan pada sistem jika diperlukan. Setiap kali gambar dianalisis dan objek berhasil dikenali, informasi tersebut akan dicatat dalam Log-deteksi. txt, yang akan terus diperbarui selama sistem masih berjalan.

3.2. Proses penyajian data

Proses penyajian informasi adalah langkah berikutnya setelah gambar diolah dan objek terdeteksi di sisi server. Tahap ini menitikberatkan pada cara komunikasi data antara server dan browser pengguna, sehingga hasil deteksi bisa disampaikan dengan cepat, informatif, dan mudah untuk diawasi. Dalam penelitian ini, penyajian data dirancang dengan memanfaatkan gabungan protokol HTTP dan WebSocket, di mana keduanya memiliki fungsi yang berbeda namun saling mendukung satu sama lain.



Gambar 3. Proses penyajian data

Secara umum, penyajian data dimulai saat pengguna membuka aplikasi pengawasan melalui peramban. Peramban akan mengirimkan permintaan awal kepada server web menggunakan protokol HTTP. Tujuan dari permintaan ini adalah untuk memuat halaman antarmuka aplikasi yang terdiri dari beberapa elemen utama, yaitu file index. html sebagai struktur tampilan, main. js sebagai pengontrol logika aplikasi, view. js untuk pengelolaan tampilan data,

serta websocket. js yang berfungsi mengelola komunikasi real-time antara peramban dan server.

Setelah halaman website selesai dimuat, browser akan membuat hubungan WebSocket dengan server. Hubungan ini bersifat permanen, sehingga memungkinkan terjadinya pertukaran data dua arah secara terus-menerus tanpa harus mengajukan permintaan baru seperti yang dilakukan pada HTTP biasa. Penggunaan WebSocket di tahap ini dipilih karena dapat mengurangi penundaan dan beban komunikasi, membuatnya sangat cocok untuk sistem pemantauan yang memerlukan pembaruan data secara langsung.

Pada bagian server, modul websocket_handler. py berfungsi untuk mengelola koneksi WebSocket yang berasal dari browser. Modul ini akan mengawasi setiap perubahan atau pembaharuan data yang datang dari hasil mendeteksi objek, khususnya informasi yang tersimpan dalam file Log-deteksi. txt. Saat ada data deteksi terbaru, server akan membaca informasi itu dan mengemasnya ke dalam format data yang terstruktur sehingga lebih mudah untuk diolah dan ditampilkan di browser.

Data yang terdeteksi dan dikirim melalui WebSocket mencakup detail penting, seperti jenis objek yang teridentifikasi (contohnya manusia), waktu ketika deteksi berlangsung, serta skor kepercayaan dari model YOLO. Setelah itu, data ini diterima oleh browser dan diproses oleh websocket. js, yang kemudian mengirimkan informasi itu ke view. js untuk ditampilkan di antarmuka pengguna. Penyajian informasi dapat berupa teks pemberitahuan, daftar riwayat deteksi, atau indikator visual yang menunjukkan adanya aktivitas di area yang diawasi.

Selain komunikasi langsung menggunakan WebSocket, protokol HTTP masih diperlukan untuk kebutuhan komunikasi yang tidak bersifat real-time, seperti permintaan data awal, memuat kembali halaman, atau mengambil berkas statis. Dengan memisahkan fungsi-fungsi HTTP dan WebSocket berdasarkan perannya, sistem yang dibuat menjadi lebih efisien dan teratur, serta mampu memberikan pengalaman pemantauan yang responsif kepada pengguna.

Melalui sistem ini, pengguna bisa melihat hasil deteksi objek secara instan melalui browser tanpa perlu menyegarkan halaman secara manual. Setiap pergerakan yang ditangkap oleh ESP32-CAM akan diproses oleh server dan disampaikan kepada pengguna dalam waktu yang cukup cepat. Dengan cara ini, penyajian data yang dibuat dalam penelitian ini dapat memenuhi kebutuhan sistem pengawasan berbasis ESP32-CAM yang bersifat real-time, efektif, dan mudah diakses.

4. HASIL DAN PEMBAHASAN

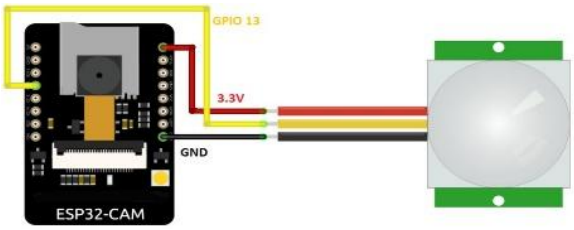
Dalam studi ini, akan dianalisis dan dibagi menjadi empat bagian, yaitu (1) penjelasan umum tentang penerapan sistem, (2) hasil dari penerapan model komunikasi data, (3) hasil identifikasi objek dengan menggunakan YOLO, dan (4) penilaian kinerja sistem.

4.1. Gambaran Umum Implementasi Sistem

Implementasi dari sistem komunikasi data yang dijelaskan dalam bab metode telah berhasil dilaksanakan di dalam jaringan lokal (LAN). Sistem ini terdiri dari tiga elemen utama, yaitu ESP32-CAM, server web, dan browser yang bekerja bersama dalam satu sistem pengawasan yang efektif. Sistem ini dirancang agar dapat berfungsi sepenuhnya tanpa memerlukan koneksi internet, menggunakan komunikasi data melalui jaringan lokal (Wi-Fi) untuk menjamin kecepatan dan kestabilan yang terbaik.

4.2. Peran ESP32-CAM dalam Sistem

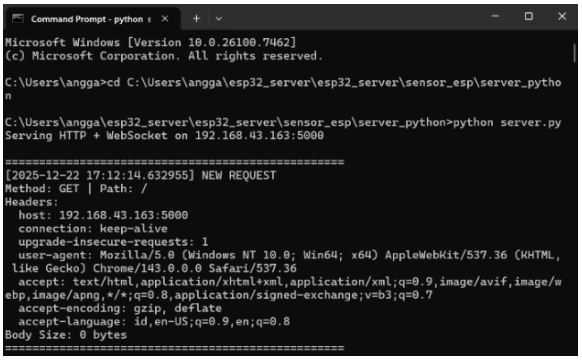
ESP32-CAM bertindak sebagai alat utama untuk mengambil gambar dalam sistem ini. Dengan adanya sensor PIR (Sensor Inframerah Pasif), ESP32-CAM memiliki kemampuan untuk mendeteksi gerakan di dalam area yang dipantau. Saat sensor PIR menangkap adanya gerakan, ESP32-CAM secara otomatis akan mengambil foto dan mengirimkannya ke server web melalui jaringan lokal dengan menggunakan protokol HTTP. Pemanfaatan jaringan lokal (LAN) memungkinkan perangkat melakukan transfer gambar dengan cepat tanpa harus bergantung pada koneksi internet, yang sangat penting untuk pengawasan secara langsung.



Gambar 4. Rangkaian ESP32-Cam dan sensor PIR

4.3. Peran Web Server dalam sistem

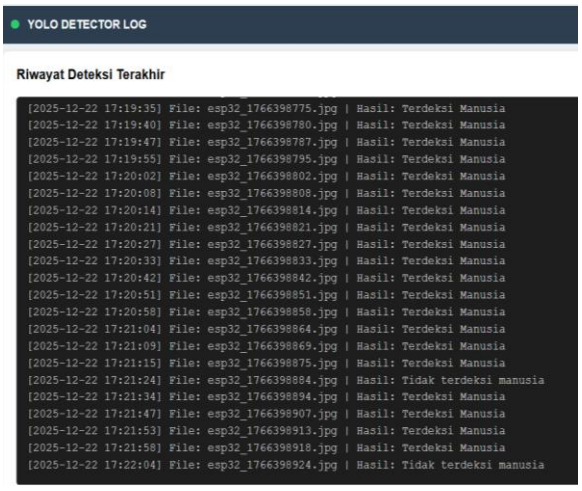
Server web berfungsi sebagai penghubung utama antara ESP32-CAM dan pengguna yang menggunakan browser. Server ini menerima foto yang dikirim oleh ESP32-CAM, menyimpannya di folder yang telah ditentukan, dan kemudian memproses gambar tersebut untuk mendeteksi objek. Menggunakan model YOLO (You Only Look Once), server menganalisis gambar dan mengidentifikasi objek, seperti manusia, dalam foto yang diambil. Hasil deteksi tersebut kemudian dikirimkan ke pengguna melalui browser secara langsung dan real-time melalui koneksi WebSocket, yang memungkinkan notifikasi diperbarui secara langsung tanpa harus menyegarkan halaman web secara manual.



Gambar 5. Hasil server berjalan

4.4. Peran Browser dalam Sistem

Browser berperan sebagai antarmuka bagi pengguna yang memungkinkan mereka untuk melihat hasil deteksi objek secara langsung. Ketika halaman aplikasi terbuka, browser membuat sambungan WebSocket dengan server untuk mendapatkan pemberitahuan deteksi objek secara langsung. Dengan koneksi yang terus berlangsung ini, informasi yang dikirimkan oleh server dapat diterima dan ditampilkan langsung di browser pengguna setiap kali objek terdeteksi. Pengguna tidak perlu menyegarkan halaman, karena data akan diperbarui secara otomatis melalui WebSocket.



Gambar 6. Hasil tampilan browser

Secara keseluruhan, penerapan sistem pemantauan yang menggunakan ESP32-CAM, server web, serta peramban yang berjalan di jaringan lokal berjalan dengan baik. Selanjutnya, sistem ini berhasil beroperasi tanpa bergantung pada akses internet, dengan pengiriman data yang efisien dan stabil.

4.5. Hasil implementasi model komunikasi data

Komunikasi HTTP dan WebSocket memainkan fungsi penting dalam transmisi gambar dan pemberitahuan secara langsung, serta memperkuat kecepatan dan efektivitas sistem secara keseluruhan. Ini terbukti melalui pengujian sebagai berikut :

4.6. Pengiriman Gambar dengan HTTP

Proses pengiriman foto dari ESP32-CAM menuju server web yang menggunakan protokol HTTP berlangsung dengan lancar dan efektif dalam jaringan lokal. Setelah mendeteksi gerakan, foto yang ditangkap oleh ESP32-CAM akan dikirim ke server melalui HTTP. Keberhasilan dalam mengirim foto tersebut tercatat dengan jelas dalam tabel di bawah ini :

Tabel 1. Hasil pengiriman gambar dengan HTTP

Metrik	Deskripsi	Rentang Nilai
Ukuran File	Ukuran umum gambar yang dikirim, dipengaruhi oleh tingkat ketajaman dan keadaan gambar.	50 KB - 150 KB
Waktu Kirim	Durasi yang diperlukan untuk mentransfer gambar dari ESP32-CAM ke server	300 ms - 500 ms
Konsistensi Data	Keandalan dalam pengiriman gambar, yaitu apakah data diterima dengan tepat tanpa adanya kehilangan atau kerusakan.	100% keandalan (Tanpa kehilangan atau kerusakan data)

4.7. Notifikasi Real-Time dengan WebSocket

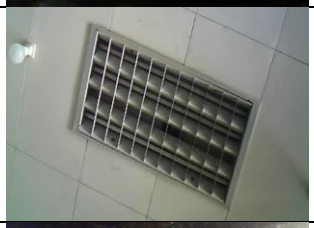
WebSocket telah diterapkan untuk mentransmisikan hasil deteksi objek secara langsung ke browser. Dengan WebSocket, terjalin koneksi yang terus-menerus antara server dan browser, sehingga data dapat dikirim secara langsung tanpa harus me-refresh halaman. Sistem ini sangat berguna untuk memberikan informasi deteksi objek segera setelah proses deteksi selesai. Hasil dari penerapan WebSocket menunjukkan bahwa pemberitahuan bisa diterima di browser dalam waktu yang sangat singkat setelah deteksi objek selesai. Salah satu manfaat utama dari WebSocket adalah kemampuannya untuk mempertahankan koneksi tetap aktif dan memungkinkan pengiriman data dengan cara push. Ini berbeda dengan metode HTTP polling, yang memerlukan browser untuk secara rutin mengajukan permintaan ke server guna memeriksa adanya pembaruan.

4.8. Hasil Deteksi Objek Menggunakan YOLO

Pada percobaan ini, gambar yang dihasilkan oleh ESP32-CAM lewat sensor PIR dikirimkan ke server untuk diproses menggunakan model YOLO. Hasil dari pendeteksian objek manusia menunjukkan bahwa model YOLO berhasil mengenali objek manusia dengan cukup baik, terutama ketika kondisi pencahayaan mendukung dan objek berada dalam jarak yang dekat. Namun, dalam situasi dengan pencahayaan yang kurang atau saat objek berada lebih jauh, tingkat akurasi dalam pendeteksian mengalami penurunan. Ini menandakan bahwa pencahayaan serta

jarak objek merupakan elemen penting yang memengaruhi kemampuan model dalam mendeteksi objek manusia dengan tepat.

Tabel 2. Sampel hasil pembacaan YOLO

No	Objek Gambar	Hasil Pembacaan
1.		Tidak terdeteksi manusia
2.		Terdeteksi Manusia
3.		Tidak terdeteksi manusia
4.		Terdeteksi Manusia
5.		Terdeteksi Manusia

Secara keseluruhan, walaupun terdapat beberapa kesulitan teknis berkenaan dengan pencahayaan dan jarak objek, YOLO memperlihatkan performa deteksi yang sangat baik dalam situasi yang lebih optimal, dengan tingkat konsistensi yang cukup tinggi pada pengujian yang telah dilakukan.

4.9. Evaluasi Kinerja Sistem

Evaluasi performa sistem menunjukkan bahwa pertukaran data antara perangkat ESP32-CAM, server, dan browser pengguna berlangsung dengan baik. Pengujian yang dilakukan menggunakan protokol HTTP untuk mengirimkan gambar dari ESP32-CAM ke server menunjukkan adanya sedikit latensi, tetapi tetap operasional dan tidak mengganggu keseluruhan kinerja sistem. WebSocket, yang diterapkan untuk mengirimkan notifikasi hasil deteksi objek, terbukti sangat efisien dengan latensi yang sangat minim, yang memungkinkan pengguna menerima pemberitahuan tentang deteksi objek hampir secara seketika. Pengujian akurasi deteksi objek menggunakan model YOLO menunjukkan hasil yang sangat baik dalam kondisi pencahayaan yang ideal, dengan rata-rata skor keyakinan di atas 90%. Meski demikian, pencahayaan yang buruk dan objek yang berada jauh mengurangi keakuratan deteksi, dengan skor keyakinan yang lebih rendah dalam situasi tersebut.

Dalam aspek kecepatan respons, sistem ini mengeluarkan pemberitahuan dalam waktu nyata dengan sangat cepat, yaitu dalam waktu kurang dari satu detik setelah objek terdeteksi, sehingga menjamin pengalaman pengawasan yang responsif. Secara keseluruhan, sistem ini berhasil mencapai tujuan penelitian dengan menyediakan pengawasan yang efisien, tepat, dan responsif dalam waktu nyata, meskipun terdapat beberapa kendala terkait dengan pencahayaan dan jarak objek.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil merancang serta mengembangkan sistem komunikasi data yang efektif antara perangkat ESP32-CAM, server, dan browser pengguna dalam aplikasi pengawasan yang berbasiskan WebSocket. Dengan memadukan protokol HTTP untuk pengiriman gambar dan WebSocket untuk notifikasi secara langsung, sistem ini mampu mengatasi masalah latensi dan meningkatkan responsivitas dalam proses pengawasan. Deteksi objek melalui model YOLO menunjukkan performa yang baik dalam mengidentifikasi manusia, meskipun terdapat tantangan saat menghadapi kondisi pencahayaan yang kurang baik atau objek yang berada jauh. Sistem ini berfungsi dengan baik dalam jaringan lokal (LAN), tanpa bergantung pada koneksi internet, sehingga menjadikannya solusi yang efisien dan praktis untuk pengawasan secara real-time.

Untuk meningkatkan ketepatan deteksi dalam situasi pencahayaan yang kurang baik, dianjurkan untuk memperbaiki model YOLO atau memanfaatkan sistem pencahayaan tambahan di sekitar area pengawasan. Selain itu, demi meningkatkan performa deteksi pada objek yang berada pada jarak lebih jauh, diperlukan penelitian lebih lanjut terkait perbaikan kualitas gambar atau penggunaan lensa dengan jangkauan yang lebih luas pada perangkat ESP32-CAM. Integrasi sistem keamanan, seperti enkripsi data, juga penting untuk memperkuat keamanan dalam proses pengiriman gambar dan notifikasi pada sistem pengawasan yang berbasis media tersebut. Terakhir, evaluasi lebih mendalam mengenai pengaruh

gangguan sinyal atau masalah jaringan yang tidak diprediksi terhadap kinerja sistem di lingkungan nyata dapat memberikan pemahaman yang lebih menyeluruh mengenai ketahanan dan kehandalan sistem dalam situasi yang lebih dinamis. Dengan perbaikan-perbaikan ini, sistem pengawasan tersebut berpotensi menjadi lebih efisien dan terpercaya untuk berbagai aplikasi pengawasan yang lebih canggih.

DAFTAR PUSTAKA

[1] L. Cakrayuda, M. R. Arhieadhie, A. S. Putra, U. Pertahanan, and R. Indonesia, "SICEMOT : SISTEM KEAMANAN CERDAS BERBASIS ESP32-CAM , SENSOR GERAK , DAN NOTIFIKASI," vol. 13, no. 2, 2025.

[2] J. Syahpita, "Sistem Alarm Deteksi Gerak Berbasis IoT Menggunakan Sensor PIR dan ESP32 dengan Notifikasi Telegram Real-Time," vol. 4, no. 2, pp. 1469–1476, 2025.

[3] I. Of *et al.*, "IMPLEMENTATION OF IOT IN IMPROVING THE EFFICIENCY OF," pp. 197–205, 2024.

[4] P. Murley, Z. Ma, J. Mason, and M. Bailey, "WebSocket Adoption and the Landscape of the Real-Time Web".

[5] A. Lukman, N. Rokhim, M. Syafaat, and D. Widiatmoko, "Web-Based Pistol Storage Security Monitoring System Optimization for Database Effectiveness," vol. 6, no. 2, pp. 177–182, 2024, doi: 10.30812/bite.v6i2.4570.

[6] I. Petrescu, E. Niculae, V. Vulturescu, A. Dimitrescu, and L. M. Ungureanu, "Transport and Application Layer Protocols for IoT : Comprehensive Review," 2025.

[7] N. Surantha and N. Sutisna, "Key Considerations for Real-Time Object Recognition on Edge Computing Devices," pp. 1–26, 2025.

[8] Y. Chang, F. Wu, and H. Lin, "Design and Implementation of ESP32-Based Edge Computing for Object Detection," 2025.

[9] I. Farhan Santoso, H. Setiawan, A. Sridaryono, D. Widiatmoko, and Kasiyanto, "Monitoring the security system at pistol storage based on a web server in army units," *TEKNOSAINS J. Sains, Teknol. dan Inform.*, vol. 11, no. 2, pp. 275–279, 2024, doi: 10.37373/tekno.v11i2.1056.

[10] H. J. J. Ochoa, R. Peña, Y. L. Mezquita, E. Gonzalez, and S. Camacho-leon, "Comparative Analysis of Power Consumption between MQTT and HTTP Protocols in an IoT Platform Designed and Cold Chain Transport Operations," 2023.

[11] J. Terven, "A Comprehensive Review of YOLO Architectures in Computer Vision : From YOLOv1 to YOLOv8 and YOLO-NAS," pp. 1680–1716, 2023.

[12] D. Nimma *et al.*, "Object detection in real-time video surveillance using attention based transformer-YOLOv8 model," vol. 118, no. January, pp. 482–495, 2025.

[13] J. T. Informatika and J. T. Informatika, "Jurnal Teknik Informatika, Vol. 16, No. 2, April 2024," vol. 16, no. 2, pp. 22–26, 2024.

[14] K. Kasiyanto, A. Apriharta, D. Widiatmoko, D. Irmanto, and M. Cahyo Bagaskoro, "Hostage Liberation Operations using Wheeled Robots Based on LIDAR (Light Detection and Ranging) Sensors," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 23, no. 2, pp. 243–258, 2024, doi: 10.30812/matrik.v23i2.3493.

[15] R. Gustiawan, F. Kholid, A. Yaqin, and D. Widiatmoko, "Infus liquid monitoring system using web-based loadcell sensor," vol. 11, no. 2, pp. 268–274, 2024, doi: 10.37373/tekno.v11i2.1015.