



**INTEGRASI SENSOR ULTRASONIK DAN ESP32-CAM PADA SISTEM
DETEKSI KENDARAAN BERBASIS IOT DENGAN SERVER FLASK DAN
NOTIFIKASI *REAL-TIME* VIA TELEGRAM DI CV INDAH JAYA
SENTOSA**

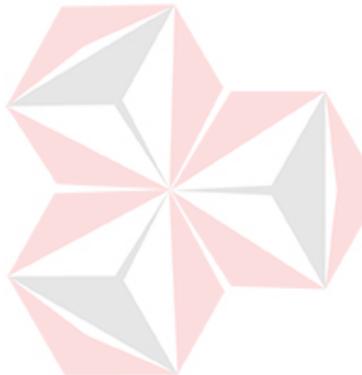


Oleh:
WAYAN VEMAS KAKA DANUARTA
22410200007

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2025

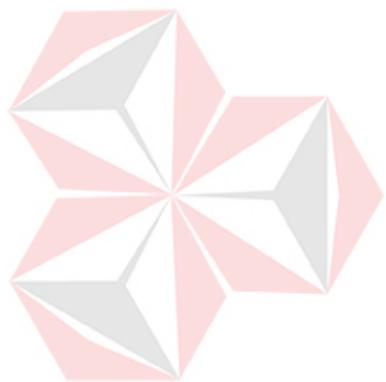
**Integrasi Sensor Ultrasonik dan Esp32-Cam Pada Sistem Deteksi Kendaraan
Berbasis IoT dengan Server Flask dan Notifikasi *Real-Time* Via Telegram di
CV Indah Jaya Sentosa**

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Strata Satu (S1)



Disusun Oleh :
Nama : Wayan Vemas Kaka Danuarta
Nim : 22410200007
Program : S1 (Strata Satu)
Jurusan : Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2025**



UNIVERSITAS
Dinamika
“Kaum Enjoy Aman”

LEMBAR PENGESAHAN

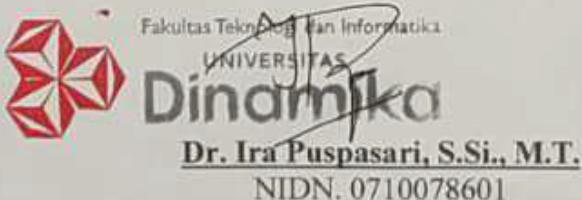
INTEGRASI SENSOR ULTRASONIK DAN ESP32-CAM PADA SISTEM DETEKSI KENDARAAN BERBASIS IOT DENGAN SERVER FLASK DAN NOTIFIKASI *REAL-TIME VIA TELEGRAM DI CV INDAH JAYA SENTOSA*

Laporan Kerja Praktik oleh
Wayan Vemas Kaka Danuarta
NIM : 22410200007
Telah diperiksa, diuji dan disetujui

Surabaya, 19 Desember 2025



Mengetahui,
Ketua Program Studi S1 Teknik Komputer



SURAT PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Universitas Dinamika, saya:

Nama : Wayan Vemas Kaka Danuarta
NIM : 22410200007
Program Studi : S1 Teknik Komputer
Fakultas : Fakultas Teknologi dan Informatika
Jenis Karya : Laporan Kerja Praktik
Judul Karya : INTEGRASI SENSOR ULTRASONIK DAN ESP32-CAM PADA SISTEM DETEKSI KENDARAAN BERBASIS IOT DENGAN SERVER FLASK DAN NOTIFIKASI *REAL-TIME* VIA TELEGRAM DI CV INDAH JAYA SENTOSA

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 8 Desember

2025 Vem
atakan



Wayan Vemas Kaka Danuarta
NIM : 22410200007

ABSTRAK

Kerja Praktik ini bertujuan untuk merancang dan mengembangkan sistem pengawasan aktivitas keluar-masuk kendaraan di lingkungan industri seperti CV Indah Jaya Sentosa masih sering dilakukan secara konvensional, yang berpotensi menimbulkan inefisiensi waktu dan celah keamanan. Kerja Praktik ini bertujuan merancang bangun sistem deteksi kendaraan otomatis berbasis *Internet of Things* (IoT) untuk meningkatkan efektivitas pemantauan gerbang. Sistem yang dikembangkan mengintegrasikan sensor ultrasonik HC-SR04 dan modul kamera ESP32-CAM. Mekanisme kerja alat dirancang menggunakan logika pemicu, dimana kamera hanya akan mengakuisisi citra visual apabila sensor mendeteksi objek fisik dalam radius kurang dari 100 cm. Data citra tersebut kemudian ditransmisikan melalui protokol HTTP POST menuju server lokal berbasis *framework* Flask. Pada sisi *backend*, server memproses data gambar menggunakan model *Deep Learning* untuk klasifikasi jenis kendaraan dan mendistribusikan hasil deteksi tersebut secara *real-time* kepada petugas keamanan melalui API Bot Telegram. Berdasarkan hasil pengujian lapangan, sistem mampu mendeteksi keberadaan kendaraan dengan akurasi pemicu yang presisi dan mengirimkan notifikasi visual dengan rata-rata latensi sebesar 5,4 detik. Implementasi sistem ini terbukti efektif sebagai solusi *early warning system* yang responsif tanpa memerlukan intervensi manual secara terus-menerus.

Kata kunci: IoT, ESP32-CAM, Sensor Ultrasonik, Flask, CNN, Deteksi Kendaraan, Telegram Bot.

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT atas segala rahmat, karunia, dan hidayahnya, sehingga penulis dapat menyelesaikan laporan kerja praktik ini dengan baik. Laporan yang berjudul *“Integrasi Sensor Ultrasonik dan Esp32-Cam Berbasis IoT dengan Server Flask dan Notifikasi Real-Time Via Telegram di CV Indah Jaya Sentosa.”* Ini disusun sebagai salah satu syarat untuk menyelesaikan program studi S1 Teknik Komputer. Laporan ini merupakan pengembangan sistem deteksi kendaraan berbasis IoT yang memanfaatkan *ESP32-CAM*, sensor Ultrasonik, dan notifikasi Telegram untuk efisiensi pemantauan.

Ucapan terima kasih disampaikan penulis kepada semua pihak yang memberikan dukungan dalam penyusunan laporan ini, yaitu:

1. Sang Hyang Widhi Wasa atas segala rahmat, berkat, dan hidayah yang telah diberikan.
2. Orang tua dan keluarga yang senantiasa memberikan doa, dukungan, dan motivasi.
3. Tan Amelia, S.Kom., M.MT. selaku Dekan Fakultas Teknologi dan Informatika
4. Dr. Ira Puspasari, S.Si., M.MT. selaku Ketua Program Studi S1 Teknik Komputer.
5. Harianto, S.Kom., M.Eng. selaku dosen pembimbing kerja praktik.
6. Yayuk Mariana selaku penyelia di CV Indah Jaya Sentosa.
7. Ahmad Yani sebagai ketua divisi IT dan Teknologi.

Penulis menyadari bahwa laporan ini masih kekurangan. Oleh karena itu, kritik dan saran yang membangun sangat diharapkan untuk perbaikan di masa mendatang.

Surabaya, 11 November 2025

Wayan Vemas Kaka Danuarta

DAFTAR ISI

ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN	xiii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Manfaat.....	3
BAB II GAMBARAN UMUM PERUSAHAAN	4
2.1 Sejarah Singkat Perusahaan	4
2.2 Visi, Misi, Dan Tujuan Perusahaan	5
2.2.1 Visi	5
2.2.2 Misi	5
2.3 Struktur Perusahaan	6
2.4 Informasi Kontak Perusahaan.....	7
BAB III LANDASAN TEORI.....	8
3.1 Internet Of Things.....	8
3.2 Esp32-Cam Development	9
3.2.1 Spesifikasi dan Karakteristik Throughput Esp32-Cam.....	10
3.2.2 Implementasi pada Deteksi Objek dengan Esp32-Cam	10



UNIVERSITAS
Dinamika



3.3 Konsep Dasar Pengolahan Citra Digital Pada Deteksi Kendaraan	11
3.3.1 Tahapan Pra-Pemrosesan (<i>Preprocessing</i>).....	12
3.3.2 Segmentasi Objek dengan Metode Background Subtraction	12
3.3.3 Perbaikan Kualitas Citra Menggunakan Operasi Morfologi.....	14
3.3.4 Klasifikasi Objek dan Ekstraksi Fitur Dengan BLOB	15
3.4 Server Flask (Web Framework).....	16
3.4.1 Arsitektur dan Fleksibilitas Sistem	16
3.4.2 Implementasi Pada Pengolahan Citra dan Kecerdasan Buatan.....	17
3.4.3 Peran Dalam Komunikasi IoT.....	17
3.4.4 Keamanan dan Efisien Layanan Web	18
3.5 Definisi Telegram Bot API.....	18
3.5.1 Mekanisme Kerja API dan Token Akses	19
3.6 Definisi Sensor Ultrasonik HC-SR04.....	19
3.6.1 Implementasi Pada Sistem Deteksi Kendaraan.....	20
3.7 USB TTL	21
BAB IV PEMBAHASAN	23
4.1 Pembahasan Kerja Praktik	23
4.2 Uraian Pembahasan.....	24
4.2.1 Perancangan Alat Hardware.....	25
4.2.2 Perancangan Alat Software	27
4.2.3 Kalibrasi Logika Program	28
4.2.4 Pengujian Sistem.....	33
BAB V PENUTUP	35
5.1 Kesimpulan	35
5.2 Saran	35
DAFTAR PUSTAKA	37

DAFTAR GAMBAR

Gambar 2.1 Logo Perusahaan.....	4
Gambar 2.2 Struktur Organisasi	6
Gambar 3.1 Esp32-Cam	9
Gambar 3.2 Sensor Ultrasonik	20
Gambar 4.1 Skematik Rangkaian Alat.	24
Gambar 4.2 Diagram Deployment	25
Gambar 4.3 Perancangan Alat.	26
Gambar 4.4 Diagram Server Flask.	28



DAFTAR TABEL

Tabel 4.1 Rekapitulasi Hasil Pengujian..... 34



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi di bidang Internet of Things (IoT) dan kecerdasan buatan (Artificial Intelligence) telah memberikan kontribusi besar dalam menciptakan sistem otomatis yang mampu melakukan pengawasan dan pengambilan keputusan secara mandiri. Salah satu bentuk penerapan teknologi tersebut adalah pada sistem deteksi kendaraan otomatis, yang memungkinkan proses identifikasi kendaraan dilakukan tanpa campur tangan manusia secara langsung. Teknologi ini sangat relevan untuk diterapkan dalam berbagai bidang, terutama dalam 1ptic1 keamanan dan manajemen lalu lintas di lingkungan industry maupun publik.

CV Indah Jaya Sentosa sebagai perusahaan yang memiliki aktivitas kendaraan keluar-masuk setiap hari membutuhkan sistem pengawasan yang efektif, cepat, dan efisien. Pengawasan kendaraan yang masih dilakukan secara manual berpotensi menimbulkan keterlambatan pencatatan serta risiko keamanan yang lebih tinggi. Oleh karena itu, dibutuhkan suatu sistem berbasis IoT yang dapat bekerja secara otomatis dalam mendeteksi keberadaan kendaraan serta memberikan laporan atau notifikasi secara real-time kepada petugas keamanan.

Dalam kerja praktik ini, dilakukan perancangan dan pengembangan sistem deteksi kendaraan berbasis IoT menggunakan ESP32-CAM sebagai perangkat utama dan sensor ultrasonik sebagai pendekripsi jarak kendaraan. Sensor ultrasonik berfungsi untuk mendeteksi adanya objek atau kendaraan pada jarak tertentu, kemudian memicu ESP32-CAM untuk mengambil gambar kendaraan. Data gambar yang diperoleh dikirimkan ke server Flask melalui metode HTTP POST menggunakan protokol komunikasi berbasis TCP/IP.

Server Flask bertugas sebagai pusat pemrosesan data yang menjalankan model deep learning TensorFlow/Keras (`vehicle_classifier.keras`) untuk mengenali jenis kendaraan seperti mobil, motor, bus, dan truk. Hasil analisis dari server kemudian dikirimkan secara real-time melalui Bot Telegram, sehingga

pengguna atau petugas keamanan dapat menerima notifikasi otomatis yang berisi gambar kendaraan dan hasil deteksi langsung di perangkat seluler.

Dengan sistem ini, pengawasan kendaraan di lingkungan CV Indah Jaya Sentosa menjadi lebih efisien, karena proses deteksi dan pelaporan dapat dilakukan secara otomatis tanpa keterlibatan manual. Selain itu, integrasi antara perangkat keras (ESP32-CAM dan sensor ultrasonik) serta perangkat lunak (Flask server, model klasifikasi kendaraan, dan API Telegram) menjadikan sistem ini sebagai contoh penerapan nyata dari teknologi IoT cerdas yang mampu meningkatkan keamanan dan efektivitas operasional perusahaan.

1.2 Rumusan Masalah

Dari latar belakang, maka dapat perumusan masalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem deteksi kendaraan otomatis berbasis EPS32-CAM dan sensor ultrasonic yang mampu bekerja secara *Real-Time*?
2. Bagaimana cara mengintegrasikan komunikasi antara perangkat IoT dengan server Flask untuk melakukan pengiriman data citra kendaraan melalui jaringan nirkabel?
3. Bagaimana sistem dapat mengirimkan hasil deteksi kendaraan secara otomatis melalui Bot Telegram agar pengguna memperoleh notifikasi secara langsung?

1.3 Batasan Masalah

Batasan masalah pada pelaksanaan Kerja Praktik ini adalah sebagai berikut:

1. Sistem hanya difokuskan pada deteksi kendaraan berdasarkan keberadaan dan jenis kendaraan, tanpa melalui identifikasi nomor plat atau pengenalan wajah pengemudi.
2. Proses klasifikasi kendaraan dilakukan di server Flask local menggunakan model TensorFlow/Keras (*vehicle_clasifier.keras*), tanpa dukungan layanan cloud computing.
3. Sistem pengujian hanya dilakukan di lingkungan CV Indah Jaya Sentosa, dengan fokus pada area keluar-masuk kendaraan perusahaan.

1.4 Tujuan

Tujuan Kerja Praktik di CV Indah Jaya Sentosa adalah sebagai berikut:

1. Merancang dan mengimplementasikan sistem IoT berbasis ESP32-CAM dan sensor ultrasonik untuk mendeteksi kendaraan secara otomatis.
2. Mengembangkan sistem komunikasi antara perangkat IoT dan server Flask menggunakan metode HTTP POST untuk mengirimkan data citra kendaraan.
3. Mewujudkan sistem notifikasi *Real-Time* melalui Bot Telegram yang mampu memberikan informasi hasil deteksi kendaraan kepada petugas keamanan.

1.5 Manfaat

Pelaksanaan Kerja Praktik ini diharapkan dapat memberikan beberapa manfaat, yaitu:

1. Memberikan solusi efisien bagi perusahaan dalam mengawasi pergerakan kendaraan secara otomatis dan akurat tanpa keterlibatan manual.
2. Menjadi penerapan teknologi IoT dan kecerdasan buatan dalam bidang pengawasan keamanan di lingkungan industri.
3. Menambah pengetahuan dan pengalaman praktis bagi mahasiswa dalam mengintegrasikan perangkat keras dan perangkat lunak pada sistem berbasis IoT.

BAB II

GAMBARAN UMUM PERUSAHAAN

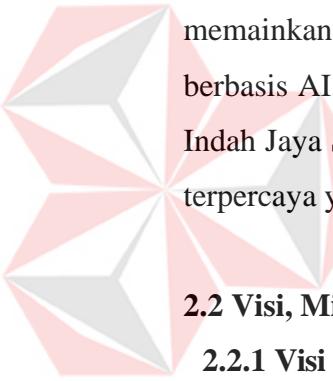
2.1 Sejarah Singkat Perusahaan

Indah Jaya Sentosa adalah perusahaan yang bergerak di bidang distribusi barang, dengan fokus pada penyediaan layanan pengiriman dan pengelolaan rantai pasok untuk berbagai jenis produk. Berbasis di Surabaya, perusahaan ini telah menjalankan operasinya selama beberapa tahun dengan mengandalkan jaringan distribusi yang luas dan tim operasional yang kompeten. Berlokasi strategis di Jalan Teluk Nibung Barat 7/20, Perak, Surabaya, Jawa Timur, CV. Indah Jaya Sentosa memiliki aksesibilitas optimal ke pelabuhan dan jalur transportasi utama, yang mendukung kelancaran distribusi barang ke berbagai wilayah di Indonesia. Komunikasi dengan perusahaan dapat dilakukan melalui nomor telepon 081332876018 atau email indahjayasentosa1@yahoo.com, dengan kontak utama Yayuk Mariana, yang memastikan responsivitas terhadap kebutuhan pelanggan.



Gambar 2.1 Logo Perusahaan

CV. Indah Jaya Sentosa menghadapi tantangan operasional yang signifikan, terutama dalam pengelolaan arus kendaraan yang meliputi truk pengangkut barang, mobil, motor, dan bus, dengan rata-rata 50 kendaraan per hari. Proses monitoring dan pencatatan kendaraan saat ini masih dilakukan secara manual oleh petugas keamanan, yang sering kali menyebabkan keterlambatan penyusunan laporan, kesalahan input data, dan keterbatasan informasi *real-time* untuk keperluan



manajerial. Kondisi ini berdampak pada efisiensi operasional, khususnya dalam pengawasan truk yang memainkan peran krusial dalam aktivitas distribusi. Untuk mengatasi tantangan tersebut, perusahaan berkomitmen untuk mengadopsi transformasi digital melalui penerapan teknologi kecerdasan buatan, seperti sistem klasifikasi kendaraan berbasis *deep learning*, guna meningkatkan akurasi dan kecepatan pengelolaan data kendaraan.

Struktur organisasi CV. Indah Jaya Sentosa dipimpin oleh Direktur Utama, Yuri AS, yang bertanggung jawab atas arah strategis dan kebijakan perusahaan, didukung oleh Wakil Direktur Yacob Adi Saputra untuk memastikan koordinasi antardivisi yang efektif. Divisi Operasional, yang dipegang oleh Rachmad Fajar, mengelola aktivitas harian termasuk pengawasan kendaraan, sementara Yayuk Mariana sebagai kepala Sumber Daya Manusia mengelola aspek perekrutan dan kesejahteraan karyawan. Divisi IT dan Teknologi di bawah Ahmad Yani memainkan peran kunci dalam mengintegrasikan solusi teknologi, seperti sistem berbasis AI, untuk mendukung efisiensi operasional. Dengan pendekatan ini, CV. Indah Jaya Sentosa berupaya memperkuat posisinya sebagai perusahaan distribusi terpercaya yang mampu beradaptasi dengan dinamika industri modern.

2.2 Visi, Misi, Dan Tujuan Perusahaan

2.2.1 Visi

“Menjadi perusahaan distribusi terpercaya dan terdepan di Indonesia dengan pelayanan prima, jaringan yang luas, serta komitmen terhadap kualitas dan kepuasan pelanggan.”

2.2.2 Misi

Untuk mewujudkan visi tersebut, perusahaan berkomitmen memberikan pelayanan distribusi yang cepat, tepat, dan aman; menjaga kualitas produk sesuai standar; membangun jaringan distribusi yang luas dan efisien; mengoptimalkan teknologi untuk mendukung efektivitas operasional menjalin hubungan kerja yang baik dengan mitra bisnis serta terus meningkatkan kompetensi sumber daya manusia melalui pelatihan dan pengembangan berkelanjutan.

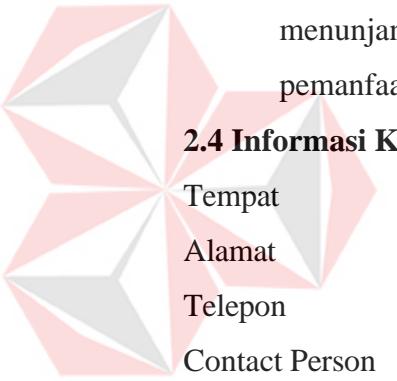
2.3 Struktur Perusahaan

Pada gambar struktur organisasi pada CV. Indah Jaya Sentosa. Setiap bagian memiliki tugas pokok dan fungsi masing-masing. Berikut di bawah ini adalah detail dari tugas pokok dan fungsinya



Gambar 2.2 Struktur Organisasi

1. Direktur Utama – Yuri AS Pemimpin tertinggi perusahaan yang bertanggung jawab penuh atas VISI, MISI, Strategi dan arah kebijakan perusahaan. Mengambil keputusan strategis yang berpengaruh besar terhadap perkembangan dan keberlangsungan usaha. Memastikan seluruh departemen bekerja sesuai target dan tujuan perusahaan.
2. Wakil Direktur – Yacob Adi Saputra Mendampingi dan membantu Direktur Utama dalam menjalankan tugas manajerial. Menggantikan peran Direktur Utama saat berhalangan hadir. Mengawasi koordinasi antar divisi agar berjalan efisien dan efektif
3. Keuangan dan Akuntansi – Achmad Hambali Mengelola arus kas masuk dan keluar perusahaan. Menyusun laporan keuangan secara berkala. Mengatur anggaran, pengeluaran, pajak dan memastikan kepatuhan terhadap regulasi keuangan.
4. Sumber Daya Manusia (SDM) – Yayuk Mariana Mengelola perekrutan, pelatihan dan pengembangan karyawan. Mengatur administrasi karyawan, gaji, tunjangan dan kesejahteraan pegawai. Menjaga hubungan kerja yang



sehat dan produktif antara tim.

5. Operasional – Rachmad Fajar Mengatur kegiatan operasional harian agar berjalan lancar dan efisien. Mengoptimalkan proses kerja di lapangan. Memastikan standar kualitas produk atau layanan terpenuhi.
6. Pemasaran dan Penjualan – Irsya Pratiwi Menyusun strategi pemasaran untuk memperluas pangsa pasar. Mengelola promosi, branding dan hubungan dengan pelanggan. Mengawasi proses penjualan dan pencapaian target penjualan.
7. Riset dan Pengembangan – Mujiah Mengembangkan inovasi produk atau layanan baru. Melakukan riset pasar dan analisis tren industri. Meningkatkan kualitas dan efisiensi produk agar tetap kompetitif.
8. IT dan Teknologi – Ahmad Yani Mengelola infrastruktur teknologi perusahaan. Mengembangkan dan memelihara sistem informasi yang menunjang operasional. Menjaga keamanan data dan memastikan pemanfaatan teknologi secara optimal.

2.4 Informasi Kontak Perusahaan

Tempat	: CV. Indah Jaya Sentosa
Alamat	: Jl. Teluk Nibung barat 7/20, Perak, Surabaya, Jawa Timur
Telepon	081332876018
Contact Person	: Yayuk Mariana
Email	: indahjayasentosa1@yahoo.com

BAB III

LANDASAN TEORI

3.1 Internet Of Things

Internet of Things (IoT) merupakan sebuah paradigma teknologi revolusioner yang mengubah pola hidup tradisional menjadi gaya hidup berbasis teknologi tinggi (*high-tech lifestyle*) (Sholihin et al., 2021). Konsep ini mengacu pada transformasi di mana objek-objek fisik di lingkungan sekitar dihubungkan ke dalam jaringan internet, memungkinkan terjadinya pertukaran data secara global. Melalui konektivitas ini, IoT memfasilitasi terciptanya ekosistem cerdas seperti kota pintar (*Smart City*), rumah pintar, hingga transportasi pintar, yang bertujuan untuk meningkatkan kualitas dan efisiensi aktivitas manusia (Kumar et al., 2019).

Ditinjau dari aspek teknis, jaringan IoT generasi masa depan dibangun oleh “objek cerdas” (*Intelligent Objects*) yang telah diintegrasikan dengan perangkat lunak (*Software*) dan sensor. Integrasi ini memungkinkan perangkat tidak hanya sekadar terhubung, tetapi juga mampu memproses data dari lingkungan sekitarnya secara mandiri (Aydos et al., 2019).

Dalam struktur cara kerjanya, IoT bertumpu pada komponen fundamental yang disebut “lapisan persepsi” (*Perception Layer*). Lapisan ini memanfaatkan berbagai jenis sensor seperti kamera pemindai, sensor ultrasonic, atau teknologi identifikasi lainnya untuk melakukan penginderaan (*sensing*), pelacakan lokasi (*positioning*), dan identifikasi objek secara otomatis (Lei, 2022). Agar komunikasi data antar-perangkat tersebut dapat berjalan tanpa kendala kabel fisik, sistem ini mengandalkan antarmuka nirkabel atau *Wireless Sensor Networks (WSN)* sebagai tulang punggung utama jaringan (Raimundo & Rosário, 2022).

Penerapan teknologi IoT kini telah meluas ke berbagai industri strategis, termasuk otomotif, konstruksi, dan manajemen logistik. Khusus dalam konteks pemantauan transportasi dan distribusi, penggunaan teknologi ini memungkinkan visualisasi proses secara real-time. Kemampuan ini dinilai sangat efektif untuk meminimalisir kesalahan manusia (*Human Error*) serta meningkatkan efisiensi manajemen operasional secara signifikan (Lei, 2022).

Namun, seiring dengan meluasnya adopsi perangkat yang terhubung mulai dari skala rumah tangga hingga infrastruktur industri aspek keamanan siber menjadi isu krusial. Mengingat setiap artefak fisik kini menjadi bagian dari jaringan data yang luas, perlindungan terhadap kerentanan sistem menjadi prioritas yang harus diperhatikan dalam perancangannya (Aydos et al., 2019; Raimundo & Rosário, 2022).

3.2 Esp32-Cam Development

ESP32-CAM didefinisikan sebagai platform riset kamera yang dapat diprogram (programmable camera research platform) yang menggabungkan kemampuan komputasi mikrokontroler dengan modul kamera optik dalam satu papan sirkuit yang ringkas (Dietz et al., 2022). Dalam konteks *Internet of Things (IoT)*, perangkat ini berfungsi sebagai node sensor visual yang mampu menangkap citra dan mentransmisikannya secara nirkabel untuk keperluan pemantauan jarak jauh (Arrahma & Mukhaiyar, 2023).



Gambar 3.1 Esp32-Cam

(Sumber: <https://www.amazon.in>)

Pemanfaatan ESP32-CAM tidak terbatas pada pengambilan gambar pasif, tetapi juga sebagai komponen utama dalam sistem deteksi kendaraan, seperti pada aplikasi pengamanan sepeda motor (MOCLESS) di mana modul ini bertindak sebagai verifikator visual (Zidane & Rahmadewi, 2022). Selain itu, modul ini juga sering diintegrasikan dengan komponen keamanan lain seperti sensor ultrasonik untuk menciptakan sistem deteksi trigger jarak yang komprehensif pada area terbatas (Sholihin et al., 2021). Fleksibilitas ini menjadikan ESP32-CAM solusi

yang efektif untuk menggantikan sistem CCTV konvensional yang cenderung lebih mahal dan kompleks dalam instalasinya (Bayu Unggul Sejati & Hengky Triyo, 2025).

3.2.1 Spesifikasi dan Karakteristik Throughput Esp32-Cam

Secara arsitektur, ESP32-CAM ditenagai oleh prosesor dual-core 32-bit LX6 yang memiliki kinerja tinggi untuk menangani tugas pemrosesan gambar dasar sebelum data dikirimkan (Dietz et al., 2022). Modul ini umumnya dilengkapi dengan sensor kamera tipe OV2640 yang mampu menghasilkan citra dengan resolusi yang cukup untuk keperluan analisis objek (Arrahma & Mukhaiyar, 2023). Selain unit pemrosesan dan kamera, papan pengembangan ini juga memfasilitasi slot kartu microSD yang berfungsi sebagai media penyimpanan lokal (local storage) untuk menyimpan bukti visual ketika jaringan tidak tersedia (Zidane & Rahmadewi, 2022).

Ditinjau dari aspek kinerja jaringan, ESP32-CAM menunjukkan reliabilitas yang memenuhi standar untuk aplikasi video streaming berbasis IoT. Berdasarkan pengujian *Quality of Service (QoS)* menggunakan server penerus (tunneling) seperti *Flask*, perangkat ini mampu mencapai throughput sebesar 1341 kbps. Berdasarkan standart *TIPHON*, nilai ini dikategorikan sebagai “Sangat Baik” karena throughput yang dihasilkan >1200 kbps, yang berarti server *Flask* dapat menerima aliran data video dengan lancar dan kualitas tangkapan gambar yang terjaga (Paramitha & Suartana, 2024).

Selain throughput parameter lain yang berbanding lurus adalah delay (waktu tunda) dan packet loss. Pada kondisi throughput yang optimal tersebut, rata-rata delay pengiriman data dari ESP32-CAM ke server tercatat sangat rendah, yaitu sekitar 2,07 ms, dengan persentase kehilangan paket data (packet loss) hanya sebesar 0.8%. Data ini memvalidasi bahwa kombinasi ESP32-CAM sebagai pengirim dan server berbasis web sebagai penerima sangat layak diterapkan untuk sistem deteksi kendaraan real-time (Paramitha & Suartana, 2024).

3.2.2 Implementasi pada Deteksi Objek dengan Esp32-Cam

Dalam penerapan sistem deteksi objek visual, ESP32-CAM memiliki keterbatasan sumber daya komputasi untuk menjalankan algoritma kecerdasan

buatan (*Artificial Intelligence*) yang kompleks secara mandiri pada perangkat (*on-board*). Oleh karena itu, arsitektur yang paling efektif untuk diterapkan adalah model *Client-Server*, di mana ESP32-CAM bertindak sebagai klien yang bertugas mengakuisisi data visual dan mengirimkannya ke unit pemrosesan eksternal (Sapteka et al., 2022).

Mekanisme kerja deteksi objek dimulai ketika ESP32-CAM menangkap citra baik secara terus-menerus (streaming) maupun berdasarkan pemicu sensor dan mentransmisikannya melalui protokol HTTP ke server lokal atau cloud. Di sisi server, data gambar tersebut diolah menggunakan bahasa pemrograman Python yang mengintegrasikan pustaka Computer Vision seperti OpenCV atau TensorFlow untuk mengenali pola objek tertentu, seperti wajah manusia atau kendaraan (Sapteka et al., 2022). Pendekatan ini memungkinkan sistem untuk melakukan tugas deteksi berat tanpa membebani mikrokontroler ESP32 (Sapteka et al., 2022).

Studi implementasi terbaru pada sistem pemantauan gudang memperlihatkan efektivitas integrasi antara ESP32-CAM dengan server berbasis framework Flask. Dalam konfigurasi ini, server Flask tidak hanya menerima aliran video, tetapi juga menjalankan model deteksi objek (seperti YOLO) untuk mengidentifikasi keberadaan objek secara real-time (Bayu Unggul Sejati & Hengky Triyo, 2025). Apabila objek terdeteksi, server secara otomatis memicu pengiriman notifikasi disertai bukti gambar (Snapshot) ke pengguna melalui platform pesan instan seperti Telegram, dengan rata-rata waktu respon sistem dari deteksi hingga notifikasi sekitar 5 detik (Bayu Unggul Sejati & Hengky Triyo, 2025). Selain itu, ESP32-CAM juga telah terbukti andal digunakan sebagai verifikator visual pada sistem keamanan kendaraan bermotor, di mana modul ini memvalidasi akses pengguna sebelum mengaktifkan sistem kelistrikan kendaraan (Zidane & Rahmadewi, 2022).

3.3 Konsep Dasar Pengolahan Citra Digital Pada Deteksi Kendaraan

Secara fundamental, pengolahan citra digital adalah proses manipulasi sinyal gambar yang bersifat dua dimensi. Dalam konteks deteksi kendaraan, gambar yang ditangkap oleh Esp32-Cam sebenarnya merupakan kumpulan gambar statis yang diputar dengan kecepatan tinggi atau dikenal dengan istilah *frame per second (FPS)*. Agar sistem komputer dapat "memahami" keberadaan kendaraan, data

visual mentah ini harus dikonversi menjadi format digital yang dapat dihitung secara matematis (Sudimanto, 2020). Proses ini bertujuan mengubah data piksel menjadi informasi kuantitatif, misalnya untuk membedakan area jalan raya yang kosong dengan area yang tertutup oleh badan kendaraan (Harjanta & Dewanto, 2017).

3.3.1 Tahapan Pra-Pemrosesan (*Preprocessing*)

Sebelum algoritma deteksi utama dijalankan, citra asli memerlukan serangkaian penyesuaian awal atau *preprocessing* untuk meningkatkan akurasi deteksi dan meringankan beban komputasi server.

1. **Konversi Ruang Warna (*Grayscale*)** Citra asli dari kamera umumnya memiliki format RGB (Red, Green, Blue) yang terdiri dari tiga lapisan matriks warna. Memproses tiga lapisan sekaligus membutuhkan sumber daya memori yang besar. Oleh karena itu, menurut (Sudimanto, 2020), langkah efektif yang harus dilakukan adalah mengonversi citra RGB menjadi citra berskala keabuan (grayscale). Teknik ini menyederhanakan data menjadi satu lapisan matriks intensitas cahaya (0-255) tanpa menghilangkan informasi penting mengenai bentuk dan tekstur kendaraan.
2. **Penentuan Area Minat (*Region of Interest / ROI*)** Kamera pengawas seringkali menangkap area yang tidak relevan, seperti pepohonan, trotoar, atau langit. Untuk mengatasi hal ini, diterapkan teknik Region of Interest (ROI). (Andrew et al., 2017) menjelaskan bahwa ROI berfungsi membatasi area kerja algoritma hanya pada koordinat jalan raya tempat kendaraan melintas. Dengan membuang informasi piksel di luar ROI, waktu pemrosesan per-frame dapat dipangkas secara signifikan, sehingga sistem mampu bekerja lebih responsif (real-time).

3.3.2 Segmentasi Objek dengan Metode Background Subtraction

Dalam sistem deteksi kendaraan berbasis visi komputer, tantangan utama yang dihadapi adalah bagaimana membedakan antara kendaraan yang bergerak dengan lingkungan jalan raya yang statis. Untuk mengatasi hal ini, diterapkan metode segmentasi yang dikenal sebagai Background Subtraction (pengurangan latar belakang).

Menurut (Prabowo & Zurnawita, 2018), background subtraction merupakan teknik fundamental dalam pengolahan citra yang bertujuan untuk memisahkan objek yang menjadi perhatian (foreground) dari latar belakang (background) dalam sebuah urutan video. Prinsip kerjanya didasarkan pada asumsi bahwa latar belakang (seperti aspal, marka jalan, dan pohon) cenderung memiliki nilai piksel yang konstan atau berubah secara lambat, sedangkan kendaraan yang melintas akan menyebabkan perubahan nilai piksel yang drastis dan cepat pada koordinat tertentu.

1. Mekanisme Pengurangan Citra Secara matematis, proses ini dilakukan dengan mengurangkan matriks citra pada frame saat ini (current frame) dengan matriks citra referensi atau model latar belakang. Sudimanto (2020) menjelaskan bahwa setiap piksel pada citra masukan (I) akan dibandingkan dengan piksel pada model latar belakang (B). Operasi ini menghasilkan nilai selisih absolut ($Diff$).

Jika dirumuskan secara sederhana:

$$Diff(x, y) = |I(x, y) - B(x, y)|$$

Di mana (x, y) adalah koordinat piksel. Jika hasil pengurangan tersebut menghasilkan nilai nol atau mendekati nol, maka area tersebut dianggap sebagai jalan kosong. Sebaliknya, jika nilai selisihnya signifikan, maka area tersebut diindikasikan sebagai adanya objek baru (kendaraan) yang menutupi jalan.

2. Thresholding dan Pembentukan Citra Biner Hasil dari pengurangan di atas masih berupa citra grayscale dengan variasi intensitas. Untuk mendapatkan ketegasan objek, dilakukan proses thresholding (ambang batas). (Andrew et al., 2017) memaparkan bahwa nilai selisih piksel akan dibandingkan dengan nilai ambang batas (T) tertentu.
 - a) Jika $Diff > T$, maka piksel diklasifikasikan sebagai Objek Bergerak (biasanya ditandai warna putih/logika 1).
 - b) Jika $Diff \leq T$, maka piksel diklasifikasikan sebagai Latar Belakang (ditandai warna hitam/logika 0).

Hasil akhirnya adalah sebuah citra biner (binary mask) yang hanya menampilkan siluet kendaraan berwarna putih di atas latar belakang hitam, yang siap untuk diproses lebih lanjut.

3. Adaptasi Terhadap Perubahan Lingkungan Salah satu kelemahan metode pengurangan sederhana adalah ketidakmampuannya menangani perubahan kondisi lingkungan yang dinamis, seperti pergeseran bayangan pohon, perubahan cuaca dari cerah ke mendung, atau transisi waktu sore ke malam. (Megantara & Pramunendar, 2017) menekankan bahwa penggunaan satu citra referensi statis (static background) akan menyebabkan banyak deteksi palsu (noise) ketika pencahayaan berubah.

Oleh karena itu, dalam implementasi modern (seperti pada pustaka OpenCV di server Flask), model latar belakang tidak bersifat statis, melainkan adaptif. Model latar belakang akan diperbarui (updated) secara berkala mengikuti perubahan rata-rata intensitas cahaya lingkungan. Hal ini memastikan bahwa perubahan cahaya yang terjadi secara bertahap tidak dianggap sebagai kendaraan, sehingga akurasi deteksi tetap terjaga meskipun kondisi pencahayaan jalan raya berfluktuasi (Megantara & Pramunendar, 2017).

3.3.3 Perbaikan Kualitas Citra Menggunakan Operasi Morfologi

Setelah proses pengurangan latar belakang (background subtraction) dilakukan, citra biner yang dihasilkan (foreground mask) jarang sekali berada dalam kondisi sempurna. Seringkali muncul gangguan visual berupa bintik-bintik putih kecil (noise) akibat goyangan pohon atau perubahan cahaya mendadak, serta adanya lubang-lubang hitam pada badan kendaraan yang tidak terdeteksi secara utuh karena pantulan cahaya kaca mobil. Jika dibiarkan, kondisi ini akan menyebabkan kesalahan deteksi, di mana satu kendaraan bisa terhitung sebagai dua objek terpisah atau gangguan kecil dianggap sebagai kendaraan (Rahmawati & Adi, 2017).

Untuk mengatasi masalah tersebut, diterapkan teknik pengolahan citra yang disebut operasi morfologi. Merujuk pada (Rahmawati & Adi, 2017), terdapat dua operasi fundamental yang diterapkan secara berurutan untuk memperbaiki struktur objek kendaraan:

1. Operasi Erosi (Erosion): Proses ini bertujuan untuk mengikis tepian objek pada citra biner. Secara matematis, erosi akan menghapus piksel-piksel

putih yang berukuran sangat kecil (lebih kecil dari elemen penstruktur). Langkah ini sangat efektif untuk menghilangkan noise atau gangguan latar belakang yang tidak relevan, sehingga hanya menyisakan objek utama yang memiliki dimensi signifikan.

2. Operasi Dilasi (Dilation): Kebalikan dari erosi, dilasi berfungsi untuk memperbesar atau menebalkan area objek. Dalam konteks deteksi kendaraan, dilasi sangat krusial untuk menutup lubang-lubang kecil (holes) atau menyatukan bagian-bagian kendaraan yang terputus-putus (misalnya badan mobil dan bumper yang terpisah). Hasil dari proses ini adalah bentuk kendaraan yang solid, utuh, dan lebih mudah dikenali oleh komputer pada tahap selanjutnya (Sudimanto, 2020).

Kombinasi kedua proses ini (sering disebut Opening atau Closing) memastikan bahwa citra yang masuk ke tahap ekstraksi fitur benar-benar merepresentasikan bentuk fisik kendaraan yang valid.

3.3.4 Klasifikasi Objek dan Ekstraksi Fitur Dengan BLOB

Tahap akhir dan terpenting dalam sistem deteksi adalah menerjemahkan kumpulan piksel putih menjadi informasi objek yang bermakna. Metode yang digunakan dalam penelitian ini adalah Analisis BLOB (Binary Large Object). Menurut (Setyawan et al., 2019), BLOB didefinisikan sebagai sekumpulan piksel yang saling terhubung (connected-components) dalam sebuah citra biner, yang dianggap sebagai satu kesatuan entitas tunggal.

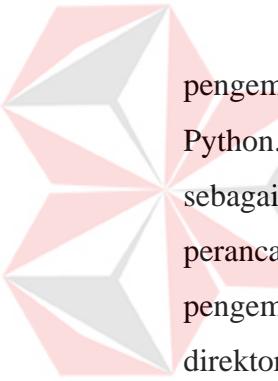
Proses ekstraksi fitur ini melibatkan beberapa tahapan analisis properti geometri untuk memastikan akurasi deteksi:

1. Pembuatan Kotak Pembatas (Bounding Box): Sistem akan menggambar kotak imajiner berbentuk persegi panjang yang melingkupi area terluar dari setiap BLOB yang terdeteksi. Koordinat kotak ini ($x, y, width, height$) menjadi acuan posisi kendaraan dalam frame video (Andrew et al., 2017).
2. Filterisasi Berdasarkan Luas Area: Tidak semua BLOB yang bergerak adalah kendaraan; bisa jadi itu adalah hewan lewat atau sampah yang terbawa angin. Untuk membedakannya, sistem menerapkan logika thresholding berdasarkan luas area ($Area = width \times height$). (Sudimanto, 2020) menjelaskan bahwa objek hanya akan diklasifikasikan

sebagai kendaraan jika memiliki luas piksel melebihi ambang batas tertentu. Objek yang ukurannya di bawah ambang batas akan diabaikan sebagai noise. Teknik ini memungkinkan sistem membedakan antara mobil (area besar), motor (area sedang), dan gangguan visual (area kecil).

3. Pelacakan Titik Pusat (Centroid Tracking): Untuk menghitung jumlah kendaraan atau memantau pergerakannya, sistem tidak melihat keseluruhan badan mobil, melainkan hanya titik tengahnya (centroid). Titik pusat ini dihitung dari rata-rata koordinat seluruh piksel penyusun BLOB. (Harjanta & Dewanto, 2017) memaparkan bahwa dengan memantau pergeseran koordinat centroid dari frame ke frame, sistem dapat menentukan arah gerak kendaraan dan menghitung kendaraan yang melintasi garis virtual penghitung.

3.4 Server Flask (Web Framework)



Flask didefinisikan sebagai sebuah kerangka kerja (framework) pengembangan aplikasi web yang dibangun menggunakan bahasa pemrograman Python. Dalam ekosistem pengembangan perangkat lunak, Flask diklasifikasikan sebagai micro-framework. Istilah "mikro" ini mengacu pada filosofi inti perancangannya yang minimalis namun esensial; Flask tidak membebani pengembang dengan pustaka bawaan (built-in libraries) yang kaku atau struktur direktori yang kompleks, berbeda dengan framework konvensional lainnya (Jonathan & Setiyawati, 2023).

Tujuan utama dari desain minimalis ini adalah memberikan fleksibilitas penuh kepada pengembang. (Jonathan & Setiyawati, 2023) menjelaskan bahwa dengan tidak adanya pemaksaan penggunaan alat tertentu (seperti validasi formulir atau abstraksi basis data bawaan), pengembang memiliki kebebasan untuk memilih dan mengintegrasikan komponen pihak ketiga yang paling sesuai dengan kebutuhan spesifik proyek mereka. Pendekatan ini menjadikan Flask solusi yang sangat efisien untuk membangun aplikasi yang membutuhkan kustomisasi tinggi, mulai dari sistem monitoring sederhana hingga layanan web yang kompleks.

3.4.1 Arsitektur dan Fleksibilitas Sistem

Ditinjau dari aspek arsitekturnya, Flask menawarkan struktur yang sangat modular dan ringan. (Curie et al., 2019) dalam analisisnya menyoroti bahwa karena

Flask tidak memuat banyak dependensi berat saat dijalankan, waktu booting dan respons aplikasi menjadi lebih cepat. Kecepatan dan efisiensi ini sangat krusial dalam pengembangan sistem backend yang harus menangani proses komputasi secara real-time.

Lebih lanjut, (Ahadi Ningrum & Ihsanudin, 2023) menekankan bahwa sifat modular Flask memungkinkannya untuk bersifat extensible (mudah diperluas). Sebuah aplikasi Flask dapat dimulai dari satu berkas skrip Python sederhana, namun dapat dikembangkan menjadi sistem berskala besar dengan menambahkan ekstensi-ekstensi modular sesuai perkembangan fitur yang diinginkan. Fleksibilitas arsitektur ini menjadikan Flask wadah yang ideal untuk menanamkan berbagai logika pemrograman tanpa terhalang oleh aturan sintaksis yang berbelit-belit (Ahadi Ningrum & Ihsanudin, 2023).

3.4.2 Implementasi Pada Pengolahan Citra dan Kecerdasan Buatan

Salah satu alasan utama penggunaan Flask dalam sistem modern adalah kemampuannya menjembatani antarmuka web dengan kekuatan komputasi Python, khususnya di bidang pengolahan citra dan kecerdasan buatan. (Setiadi et al., 2019) membuktikan bahwa Flask dapat diintegrasikan secara mulus dengan pustaka OpenCV. Dalam implementasinya, Flask berfungsi sebagai mesin pemroses di sisi server yang menerima data gambar, menjalankan algoritma manipulasi visual (seperti filterisasi atau deteksi), dan mengirimkan hasilnya kembali ke pengguna.

Selain pengolahan citra, Flask juga andal sebagai platform deployment untuk model Machine Learning. (Ahadi Ningrum & Ihsanudin, 2023) memaparkan studi kasus di mana algoritma prediksi (seperti penentuan umur transformator) ditanamkan ke dalam rute aplikasi Flask. Hal ini menunjukkan bahwa Flask mampu bertindak sebagai "otak" server yang tidak hanya melayani permintaan data statis, tetapi juga menjalankan proses berpikir atau deteksi cerdas terhadap data yang masuk.

3.4.3 Peran Dalam Komunikasi IoT

Dalam konteks sistem IoT yang melibatkan perangkat keras, Flask berperan vital sebagai pusat kendali (command center). (CHYHIN et al., 2023) menjelaskan bahwa Flask memfasilitasi komunikasi dua arah antara server dan perangkat keras

jarak jauh (remote devices) melalui protokol HTTP standar. Perangkat IoT (seperti mikrokontroler) dapat mengirimkan data sensor ke server Flask, dan sebaliknya, server Flask dapat mengirimkan instruksi balik untuk mengendalikan perangkat tersebut.

Keunggulan Flask dalam skenario ini adalah kemampuannya menangani permintaan dengan latensi rendah, yang sangat dibutuhkan untuk pengendalian perangkat secara real-time. (CHYHIN et al., 2023) menegaskan bahwa integrasi Python-Flask memungkinkan terciptanya sistem otomasi yang responsif, di mana logika kendali perangkat dapat ditulis menggunakan bahasa Python yang kuat di sisi server.

3.4.4 Keamanan dan Efisiensi Layanan Web

Meskipun berstatus micro-framework, aspek keamanan dan kinerja tetap menjadi prioritas dalam Flask. (Kornienko et al., 2021) mengemukakan bahwa Flask sangat mendukung pengembangan aplikasi dengan arsitektur *Single Page Application* (SPA). Arsitektur ini meningkatkan efisiensi bandwidth karena pembaruan data pada antarmuka pengguna dilakukan secara dinamis tanpa memuat ulang seluruh halaman. Selain itu, dengan konfigurasi yang tepat, Flask mampu menyediakan layanan web yang aman untuk melindungi jalur pertukaran data API (Application Programming Interface), memastikan bahwa informasi sensitif yang dikirimkan dari perangkat IoT tetap terjaga integritasnya (Kornienko et al., 2021).

3.5 Definisi Telegram Bot API

Telegram Bot merupakan fitur khusus dari platform pesan instan berbasis awan (cloud-based) Telegram yang memungkinkan perangkat lunak untuk berinteraksi dengan pengguna manusia secara otomatis. Berbeda dengan akun pengguna biasa yang memerlukan nomor telepon fisik, Bot adalah akun yang dioperasikan oleh skrip program melalui antarmuka pemrograman aplikasi atau API (Application Programming Interface). Dalam ekosistem Internet of Things (IoT), Telegram Bot telah menjadi standar populer untuk media notifikasi dan kendali jarak jauh karena sifatnya yang open-source, gratis, dan memiliki kecepatan pengiriman pesan yang tinggi (Zidane & Rahmadewi, 2022).

Keunggulan utama penggunaan Telegram dalam sistem keamanan adalah dukungannya terhadap pengiriman data multimedia. Bot tidak hanya mampu

mengirimkan pesan teks peringatan, tetapi juga dapat mentransmisikan bukti visual berupa foto (snapshot) atau video pendek yang diambil oleh sensor kamera di lapangan. Fitur ini sangat krusial dalam sistem deteksi kendaraan atau pemantauan keamanan, di mana pengguna membutuhkan verifikasi visual segera untuk memastikan validitas alarm yang diterima (Bayu Unggul Sejati & Hengky Triyo, 2025).

3.5.1 Mekanisme Kerja API dan Token Akses

Secara teknis, komunikasi antara sistem backend (seperti server Flask) dan aplikasi Telegram pengguna terjadi melalui protokol HTTP/HTTPS. Setiap Bot yang dibuat akan memiliki kunci akses unik yang disebut API Token. Token ini berfungsi sebagai otentikasi keamanan yang mengizinkan server Flask untuk mengirimkan permintaan (request) ke server Telegram atas nama Bot tersebut (Bayu Unggul Sejati & Hengky Triyo, 2025).

Mekanisme pengiriman notifikasi bekerja dengan metode HTTP POST Request. Ketika sistem mendeteksi adanya pemicu (misalnya kendaraan terdeteksi), server akan memanggil endpoint API Telegram (seperti metode *sendMessage* untuk teks atau *sendPhoto* untuk gambar) dengan menyertakan parameter Chat ID tujuan. Chat ID ini merupakan identitas unik dari pengguna atau grup yang akan menerima laporan, memastikan bahwa informasi sensitif hanya dikirimkan kepada pihak yang berwenang.

3.6 Definisi Sensor Ultrasonik HC-SR04

Sensor ultrasonik didefinisikan sebagai komponen elektronika yang berfungsi sebagai transduser, yaitu mengubah besaran fisis menjadi besaran listrik dan sebaliknya. Prinsip kerja utama sensor ini didasarkan pada pantulan gelombang suara atau *Time of Flight*. Sensor memancarkan gelombang bunyi pada frekuensi ultrasonik (di atas 20 kHz) yang tidak dapat didengar oleh telinga manusia, kemudian menghitung waktu yang dibutuhkan gelombang tersebut untuk memantul kembali setelah menabrak suatu objek (Moh Avin Annabil et al., 2024).



Gambar 3.2 Sensor Ultrasonik

Secara spesifik, tipe HC-SR04 yang umum digunakan memiliki dua bagian utama, yaitu transmitter (pemancar) dan receiver (penerima). Mekanismenya dimulai ketika transmitter memancarkan sinyal gelombang suara dengan frekuensi 40 kHz. Jika gelombang ini mengenai permukaan benda padat, gelombang akan dipantulkan kembali dan ditangkap oleh receiver. Mikrokontroler kemudian mengukur selisih waktu (t) antara saat gelombang dikirim hingga diterima kembali untuk mengkalkulasi jarak objek tersebut (Moh Avin Annabil et al., 2024). Rumus matematis untuk menghitung jarak (S) dalam sistem ini adalah:

$$S = \frac{v \times t}{2}$$

Dimana:

- S = Jarak antar objek (meter).
- v = Cepat rambat bunyi di udara (sekitar 340 m/s).
- t = Waktu tempuh gelombang bolak-balik (detik).
- Angka 2 digunakan karena gelombang menempuh jarak antara.

3.6.1 Implementasi Pada Sistem Deteksi Kendaraan

Dalam konteks sistem pemantauan transportasi dan parkir cerdas, sensor ultrasonik memiliki peran vital sebagai *detector* atau *trigger* keberadaan fisik kendaraan. Penelitian (Tri Laksono et al., 2023) menjelaskan bahwa sensor HC-

SR04 diimplementasikan untuk mengetahui status okupansi suatu area, yaitu membedakan antara area yang "terisi" oleh kendaraan dan area yang "kosong".

Pada sistem IoT yang terintegrasi dengan kamera Esp32-Cam, sensor ultrasonik tidak hanya berfungsi mengukur jarak, tetapi bertindak sebagai pemicu sistem. Ketika sensor mendeteksi adanya objek (kendaraan) pada jarak tertentu yang telah ditentukan (misalnya < 100 cm), sensor akan mengirimkan sinyal logika ke mikrokontroler. Sinyal inilah yang kemudian mengaktifkan kamera untuk mengambil citra kendaraan tersebut. Mekanisme ini dinilai efisien karena kamera tidak perlu memproses gambar terus-menerus, melainkan hanya aktif ketika ada kendaraan fisik yang terdeteksi oleh gelombang ultrasonik (Tri Laksono et al., 2023).

Akurasi pendektsian ini juga dipengaruhi oleh bentuk permukaan objek. (Moh Avin Annabil et al., 2024) menekankan bahwa permukaan objek yang rata dan keras (seperti bodi logam kendaraan mobil) akan memantulkan gelombang ultrasonik dengan lebih sempurna dibandingkan permukaan yang tidak beraturan, sehingga data jarak yang dihasilkan menjadi lebih presisi untuk dijadikan acuan deteksi sistem.

3.7 USB TTL

Modul Konverter *USB to TTL*, atau yang kerap dikenal sebagai USB to UART Bridge, merupakan perangkat keras esensial dalam pengembangan sistem tertanam (embedded systems) yang berfungsi menjembatani komunikasi data antara komputer dan mikrokontroler. Komputer modern umumnya beroperasi menggunakan protokol USB (Universal Serial Bus) dengan standar sinyal diferensial, sedangkan mikrokontroler termasuk ESP32 bekerja pada level logika Transistor-Transistor (TTL) sebesar 3.3V atau 5V. Oleh karena itu, modul ini berperan vital sebagai penerjemah sinyal agar kedua perangkat yang berbeda platform tersebut dapat saling bertukar informasi. Dalam konteks pengembangan perangkat keras, modul ini memiliki tiga fungsi utama, yaitu sebagai media untuk mengunggah kode program (flashing firmware) ke dalam memori mikrokontroler, sebagai alat debugging untuk memantau keluaran data serial secara real-time, serta sebagai penyedia catu daya sementara bagi mikrokontroler selama proses pengujian berlangsung.

Secara operasional, mekanisme kerja modul *USB to TTL* didasarkan pada protokol komunikasi serial asinkron *UART* (Universal Asynchronous Receiver-Transmitter). Prinsip dasar komunikasi ini mensyaratkan adanya konfigurasi penyambungan kabel secara silang (cross-wiring) antara kedua perangkat, di mana jalur pemancar (Transmitter/TX) pada modul konverter harus dihubungkan ke jalur penerima (Receiver/RX) pada mikrokontroler, dan sebaliknya. Selain itu, sinkronisasi referensi tegangan tanah (Common Ground) mutlak diperlukan untuk memastikan integritas sinyal logika biner 0 dan 1 dapat terbaca dengan akurat. Kecepatan transfer data dalam proses ini ditentukan oleh parameter Baud Rate yang harus dikonfigurasi secara identik pada kedua sisi perangkat guna mencegah terjadinya kesalahan pembacaan data.

Implementasi modul *USB to TTL* menjadi sangat krusial pada perangkat ESP32-CAM, mengingat papan pengembangan jenis ini didesain tanpa antarmuka USB bawaan demi efisiensi dimensi dan biaya produksi (Dietz et al., 2022). Penggunaan konverter eksternal seperti tipe *FT232RL* atau *CH340* menjadi syarat mutlak agar proses pemrograman dapat dilakukan. Dalam praktiknya, perhatian khusus harus diberikan pada pengaturan level tegangan kerja, di mana jumper pada modul konverter wajib diset pada posisi 3.3V untuk menyesuaikan dengan karakteristik listrik chip ESP32 dan mencegah kerusakan komponen akibat kelebihan tegangan. Lebih lanjut, proses pengunggahan program memerlukan prosedur khusus, yakni menghubungkan pin GPIO 0 ke Ground pada saat inisialisasi *booting* untuk memaksa sistem masuk ke dalam mode pemrograman *flashing mode* sebelum dikembalikan ke mode operasional normal (Arrahma & Mukhaiyar, 2023).

BAB IV

PEMBAHASAN

4.1 Pembahasan Kerja Praktik

Dalam pelaksanaan Kerja Praktik ini, penulis mendapatkan tanggung jawab di bagian mekanik, yang berperan penting sebagai perancang bentuk fisik alat dari keseluruhan sistem IoT untuk Esp32-Cam sebagai pendekripsi kendaraan dengan server Flask sebagai framework dan Telegram Bot API sebagai tanda validasi pengolahan citra kendaraannya.

Pada CV Indah Jaya Sentosa sistem yang saya kerjakan berperan sebagai integrasi sekuensial antara sensor jarak dan sensor visual. Secara teknis, alur kerja alat ini dimulai dari kondisi siaga di mana sensor Ultrasonik HC-SR04 yang terpasang pada gerbang utama melakukan pembacaan jarak secara berkelanjutan (looping). Pada tahap ini, sistem menerapkan logika pemicu yang hanya akan aktif apabila sensor mendekripsi adanya hambatan fisik yakni bodi kendaraan yang berada dalam radius jangkauan kurang dari 100 cm. Mekanisme pembatasan jarak ini sengaja diterapkan untuk mengefisiensikan kinerja perangkat, sehingga modul kamera tidak perlu merekan situasi kosong secara terus-menerus yang dapat membebani memori dan daya listrik.

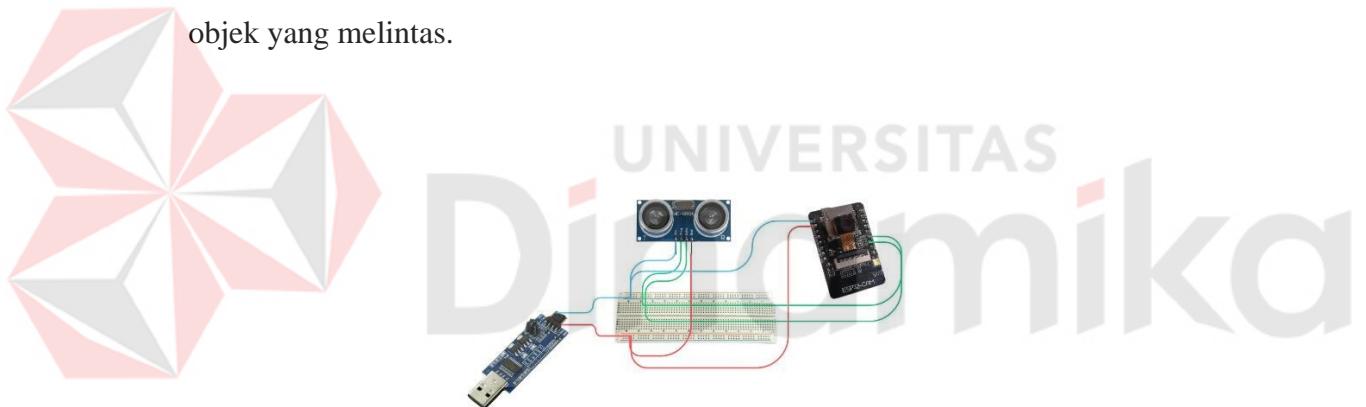
Setelah sensor ultrasonik memvalidasi keberadaan objek, mikrokontroler Esp32-Cam merespon dengan mengaktifkan modul kamera OV2640 untuk melakukan akuisisi citra visual kendaraan secara real-time. Data gambar yang berhasil ditangkap kemudian tidak disimpan di dalam modul, melainkan langsung ditransmisikan melalui jaringan Wifi lokal menggunakan protocol komunikasi HTTP POST menuju server pengolahan data. Proses transmisi ini dirancang agar berjalan dengan latensi seminimal mungkin untuk menjaga aktualitas data saat diterima oleh sistem backend.

Pada sisi pemrosesan data, server yang dibangun menggunakan framework python flask bertugas menerima permintaan data dari Esp32-Cam tersebut. Server akan menyimpan gambar dan mengirimnya ke telegram bot. Pesan yang diterima oleh security dan admin CV Indah Jaya Sentosa mencakup informasi waktu kejadian serta lampiran foto kendaraan yang melintas, sehingga verifikasi

keamanan dapat dilakukan secara instan melalui perangkat seluler tanpa harus memantau layer monitor sepanjang waktu.

4.2 Uraian Pembahasan

Penerapan sistem deteksi kendaraan di lingkungan CV Indah Jaya Sentosa beroperasi melalui mekanisme integrasi sekuensial yang memadukan sensor jarak dan sensor visual dalam satu arsitektur *Internet of Things* (IoT). Struktur penggeraan sistem diawali pada lapisan input, di mana sensor ultrasonik HC-SR04 difungsikan sebagai inisiator utama yang memantau area gerbang secara berkelanjutan. Berdasarkan pengujian lapangan, logika pemicu sistem dikalibrasi dengan ambang batas jarak 100 cm. Pengaturan ini bertujuan memvalidasi keberadaan fisik kendaraan operasional seperti truk maupun mobil pengiriman sekaligus mengefisiensikan konsumsi daya perangkat, sehingga modul kamera tidak perlu melakukan aktivitas streaming yang memboroskan bandwidth ketika tidak ada objek yang melintas.



Gambar 4.1 Skematik Rangkaian Alat.

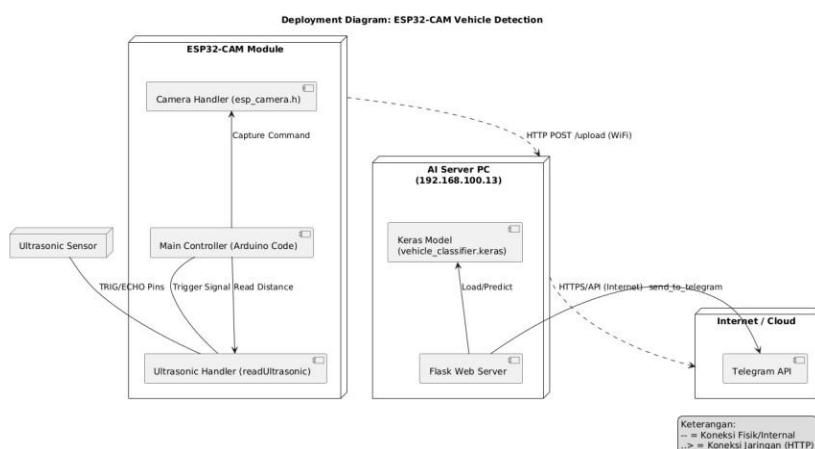
Setelah sensor jarak memvalidasi momentum kedatangan kendaraan, mikrokontroler ESP32-CAM merespons sinyal tersebut dengan melakukan akuisisi citra visual secara real-time. Tantangan teknis dalam menjaga stabilitas pengiriman data citra diatasi dengan mengonversi gambar ke dalam format digital yang efisien, untuk kemudian ditransmisikan melalui protokol HTTP POST menuju Server Flask yang berperan sebagai pusat pemrosesan data. Di sisi server, skrip Python yang

berjalan pada komputer lokal CV Indah Jaya Sentosa bekerja secara simultan untuk menerima paket data, menyimpan bukti visual tersebut ke dalam direktori arsip lokal (security log), dan mempersiapkan muatan data untuk proses distribusi informasi.

Rangkaian proses otomasi ini bermuara pada distribusi notifikasi kepada petugas keamanan melalui integrasi API Bot Telegram. Setelah data diproses oleh server, sistem secara otomatis mengirimkan pesan peringatan dini yang memuat status keamanan, waktu kejadian (timestamp), beserta lampiran foto kendaraan yang baru saja melintas. Keberadaan fitur notifikasi instan ini menjadi indikator vital keberhasilan integrasi sistem, karena memungkinkan petugas jaga untuk melakukan verifikasi visual secara langsung melalui perangkat seluler tanpa harus terpaku memantau layar monitor CCTV konvensional, sehingga menciptakan sistem pengawasan yang responsif dan efisien.

4.2.1 Perancangan Alat Hardware

Perancangan struktur alat merupakan proses rekayasa yang terfokus pada penciptaan unit tunggal yang kohesif, mengintegrasikan modul kendali utama, yakni ESP32-CAM, dengan sensor jarak akustik HC-SR04, guna memastikan fungsionalitas optimal di lingkungan operasional yang menantang. Menyadari paparan terhadap debu, kelembaban, dan perubahan suhu di gerbang CV Indah Jaya Sentosa, seluruh sirkuit elektronika dilindungi secara komprehensif di dalam kotak panel yang terbuat dari material akrilik, dirancang untuk memenuhi standar proteksi yang ketat terhadap unsur-unsur eksternal.



Gambar 4.2 Diagram Deployment.

Secara detail, modifikasi presisi pada muka panel meliputi pembentukan dua *aperture* sejajar, yang berfungsi sebagai saluran akustik murni bagi *transmitter* dan *receiver* sensor ultrasonik, menjamin gelombang suara terpancar dan diterima tanpa distorsi struktural. Berdekatan dengan formasi sensor tersebut, sebuah *aperture* lensa dibuat khusus untuk menampung modul kamera ESP32-CAM, yang posisinya dikalibrasi cermat untuk memastikan keselarasan sudut pandang visualnya secara sempurna dengan *axis* deteksi spasial sensor jarak.



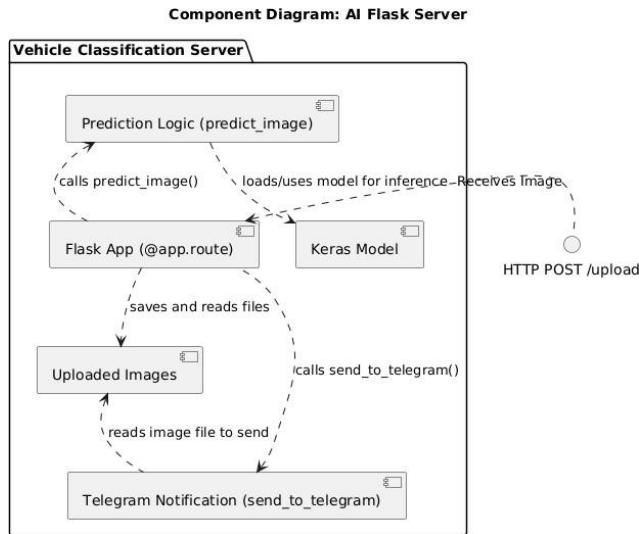
Gambar 4.3 Perancangan Alat.

Di dalam *enclosure*, koneksi elektrik dimaksimalkan melalui penyolderan permanen kabel *jumper* yang menghubungkan pin Trigger dan Echo sensor ultrasonik ke GPIO mikrokontroler, sebuah langkah preventif vital untuk menghilangkan risiko diskoneksi akibat vibrasi. Lebih lanjut, stabilitas daya *onboard* ditingkatkan melalui arsitektur catu daya 5V yang terpusat, secara simultan menyuplai energi kepada ESP32-CAM dan sensor, diperkuat dengan implementasi USB TLL yang esensial, bertujuan untuk menjadi *power* fluktuasi tegangan saat modul kamera meningkatkan aktivitas transmisi data melalui koneksi Wi-Fi. Keseluruhan unit ini disempurnakan dengan penambahan *bracket* penyangga eksternal, memfasilitasi pemasangan yang adaptif pada tiang gerbang dengan ketinggian instalasi yang dapat disesuaikan.

4.2.2 Perancangan Alat Software

Implementasi perangkat lunak dibagi menjadi hierarki operasi Klien dan Server, membentuk sebuah mekanisme deteksi dan notifikasi yang otomatis dan terdistribusi. Di sisi klien, *firmware* pada ESP32-CAM beroperasi dalam *loop* monitoring berkelanjutan, secara aktif mengeksekusi fungsi `readUltrasonic()` untuk memantau jarak objek secara *real-time*.

Aktivitas kritis *trigger* terjadi manakala jarak yang terdeteksi berada di bawah ambang batas yang telah ditentukan (*trigger distance*) dan, secara bersamaan, interval *cooldown* yang ditetapkan (`uploadInterval`) telah terpenuhi, sebuah protokol yang dirancang untuk mencegah *spamming* gambar. Setelah kondisi ganda ini terpenuhi, modul kamera segera dipicu untuk mengambil *frame* gambar resolusi tinggi (JPEG) dan memulai inisiasi koneksi HTTP POST untuk transmisi data menuju *endpoint* `/upload` pada Server AI. Di sisi Server AI, yang merupakan mesin analisis sentral berbasis Flask yang berjalan pada infrastruktur PC ber-IP statis, Flask App secara khusus dikonfigurasi sebagai *interface* penerima data multi-part. Begitu *request* gambar diterima, aplikasi segera menyimpan *payload* data (*image file*) dan mendelegasikannya kepada Prediction Logic (`predict_image`). Fungsi ini adalah inti kognitif sistem, memanfaatkan Keras Model (*vehicle_classifier.keras*) yang sudah dilatih untuk mengklasifikasikan jenis kendaraan yang terekam. Hasil analisis, yang mencakup klasifikasi kategori dan metrik *confidence* (tingkat keyakinan), selanjutnya diteruskan ke modul Telegram Notification (`send_to_telegram`).



Gambar 4.4 Diagram Server Flask.

Modul ini bertanggung jawab untuk mengorkestrasi pengiriman pesan peringatan yang kaya data dan *real-time*, menyertakan gambar yang diunggah, kepada kanal keamanan yang ditentukan, dengan logika kondisional tambahan untuk mengirim notifikasi kepada kanal administrasi khusus apabila terdeteksi kategori kendaraan yang memerlukan perhatian lebih spesifik, seperti truk.

4.2.3 Kalibrasi Logika Program

Setelah proses perakitan seluruh komponen alat selesai dilakukan, tahapan selanjutnya adalah melakukan kalibrasi logika program untuk menyeimbangkan kualitas citra dengan kecepatan transmisi. Berikut adalah segmen kode utama yang mengatur inisialisasi perangkat.

1. Konfigurasi Modul Kamera dan Resolusi.

Pada kalibrasi awal penulis melakukan pengaturan parameter kamera. Format piksel diatur ke `PIXFORMAT_JPEG` untuk kompresi data yang efisien, dan ukuran bingkai diset pada `FRAMESIZE_VGA` (640x480). Berikut codenya:

```

config.pixel_format = PIXFORMAT_JPEG;
config.frame_size = FRAMESIZE_VGA;
config.jpeg_quality = 10;
config.fb_count = 1;
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error
0x%x", err);
    while (true) delay(1000);
}

```

Pemilihan resolusi VGA ini merupakan hasil optimasi agar *payload* data tidak terlalu besar saat dikirim via Wi-Fi, namun tetap cukup jelas untuk proses deteksi AI.

2. Logika Pemicu Sensor Ultrasonik (Trigger Logic).

Bagian ini merupakan inti dari efisiensi sistem. Variable `triggerDistance` dikalibrasi dengan ambang batas 100 cm dan dapat diubah sesuai datasheet. Berikut codenya:

```

// Variabel Kalibrasi Jarak
int triggerDistance = 100;
// Logika Utama (Looping)
void loop() {
    // Membaca jarak real-time
    int distance = readUltrasonic();
    // Validasi Pemicu: Hanya aktif jika jarak > 0 DAN kurang
    dari ambang batas
    if (distance > 0 && distance < triggerDistance) {
        // Debouncing: Cek interval waktu upload agar tidak
        spamming
        if (millis() - lastUpload >= uploadInterval) {
            Serial.println("🔴      OBJEK      TERDETEKSI!      Memulai
Capture...");
            lastUpload = millis(); // Reset timer
            captureAndAnalyze(); // Panggil fungsi kamera
        }
    }
}

```

Untuk fungsi `readUltrasonic()` memastikan bahwa kamera hanya akan mengambil gambar jika ada objek fisik nyata yang tersentuh trigger sensor agar dapat terhindar dari pengambilan gambar kosong.

3. Protokol Pengiriman HTTP POST.

Setelah gambar diambil, mikrokontroler mengirim hasil gambar ke server. Kode berikut menunjukkan bagaimana data biner gambar disusun dengan format `multipart/form-data` agar dikenali oleh server Flask. Berikut Codenya:

```
void captureAndAnalyze() {  
    // ... (kode pengambilan frame buffer) ...  
  
    // Membangun Header HTTP  
    String boundary = " ---ESP32CAMBoundary";  
    String head = " --- " + boundary + "\r\n";  
    head += "Content-Disposition: form-data; name=\"file\";";  
    filename="vehicle.jpg"\r\n";  
    head += "Content-Type: image/jpeg\r\n\r\n";  
    String tail = "\r\n--" + boundary + "--\r\n";  
  
    // Mengirim Request POST ke Server Flask  
    client.println("POST " + String(serverPath) + " HTTP/1.1");  
    client.println("Host: " + String(serverHost) + ":" +  
String(serverPort));  
    client.println("Content-Type: multipart/form-data;");  
    boundary=" + boundary);  
    // ... (mengirim data biner gambar) ...  
}
```

4. Insialisasi Model dan API Telegram.

Saat server dinyalakan, sistem memuat model *Deep Learning* (`vehicle_classifier.keras`) ke dalam memori. Token API Telegram juga dideklarasikan untuk keperluan autentikasi saat pengiriman notifikasi. Berikut codenya:

```
# Konfigurasi Path dan Model AI
MODEL_PATH = "vehicle_classifier.keras"
CLASS_LABELS = ["bus", "mobil", "motor", "truck"]

# Konfigurasi Token Bot Telegram (Keamanan & Admin)
SECURITY_BOT_TOKEN = "7993325231:AAG0rtpJFCgMQ7mVaI_..."
SECURITY_CHAT_ID = "5891007214"

# Memuat Model ke Memori saat Startup
try:
    model = load_model(MODEL_PATH)
    logger.info(f"✅ Model berhasil dimuat. Siap klasifikasi.")
except Exception as e:
    logger.error(f"❌ Gagal memuat model: {e}")
```

5. Pembuatan Endpoint Penerima Data.

Fungsi ini adalah gerbang utama server. *Route* /upload dikalibrasi untuk hanya menerima metode POST. Di blok ini, server menerima file mentah lalu menyimpannya dengan penamaan berbasis waktu untuk log, dan memanggil fungsi prediksi. Berikut codenya:

```
@app.route("/upload", methods=["POST"])

def upload():
    # Validasi keberadaan file dalam request
    if "file" not in request.files:
        return jsonify({"error": "Tidak ada file dikirim"}), 400

    # Simpan file dengan Timestamp unik
    filename = datetime.now().strftime("%Y%m%d_%H%M%S") +
    ".jpg"
    save_path = os.path.join(UPLOAD_FOLDER, filename)
    request.files["file"].save(save_path)

    # Jalankan Prediksi AI
    pred_class, confidence = predict_image(save_path)

    # Kirim Notifikasi jika sukses
    send_to_telegram(SECURITY_BOT_TOKEN,
    SECURITY_CHAT_ID, message, save_path)

    return jsonify({"status": "success", "prediction": pred_class}), 200
```

6. Logika Prediksi Citra (Inference).

Fungsi ini mengubah gambar mentah menjadi *array* yang dapat dipahami oleh model AI. Gambar diubah ukurannya menjadi 224x224 piksel sesuai dengan arsitektur model yang dilatih sebelumnya. Berikut codenya:

```
def predict_image(image_path):  
    # Pre-processing Gambar  
    img = load_img(image_path, target_size=(224,  
224)) # Resize ke input model  
    arr = img_to_array(img) / 255.0  
    arr = np.expand_dims(arr, axis=0)  
    # Proses Inferensi  
    preds = model.predict(arr, verbose=0)  
    idx = np.argmax(preds[0])  
    return CLASS_LABELS[idx], float(preds[0][idx])
```

Hasil klasifikasi tersebut kemudian dikonversi menjadi label teks seperti “mobil”, “truck”, dan “motor” yang berfungsi sebagai pemicu akhir untuk mengirimkan notifikasi beserta bukti visual kepada pengguna melalui Bot Telegram secara *real-time*.

4.2.4 Pengujian Sistem

Tahapan pengujian dan analisis kinerja sistem dilaksanakan secara komprehensif di lingkungan operasional gerbang utama CV Indah Jaya Sentosa guna memvalidasi kehandalan alat dalam kondisi nyata. Rangkaian pengujian ini mencakup evaluasi terhadap akurasi pemicu sensor, validitas klasifikasi citra oleh server, serta kecepatan respons notifikasi. Pada fase awal, fokus pengujian diarahkan pada responsivitas sensor ultrasonik HC-SR04 dalam mendeteksi objek fisik. Berdasarkan skenario uji coba lapangan, sensor dikalibrasi dengan ambang batas *threshold* jarak 100 cm. Hasil pengamatan menunjukkan bahwa sensor mampu memberikan respons pemicu *trigger* yang presisi ketika kendaraan operasional seperti truk logistik maupun mobil pengiriman melintas di area deteksi, dengan rata-rata selisih pembacaan jarak hanya berkisar 1 hingga 2 cm dibandingkan pengukuran manual. Logika program terbukti efektif mengabaikan gangguan objek yang berada di luar radius pemantauan, sehingga kamera ESP32-CAM hanya aktif pada momentum yang relevan untuk efisiensi daya.

Sejalan dengan validasi sensor, evaluasi dilanjutkan pada aspek pemrosesan data visual oleh server Flask yang terintegrasi dengan modul kecerdasan buatan. Citra kendaraan yang ditangkap oleh kamera dan dikirimkan melalui protokol

HTTP POST diuji validitasnya untuk melihat kemampuan model dalam mengenali jenis objek. Dari serangkaian sampel gambar yang diambil pada kondisi pencahayaan siang hari, server berhasil mengklasifikasikan jenis kendaraan seperti label "TRUCK" untuk kendaraan berat dan "CAR" untuk kendaraan pribadi dengan tingkat keyakinan rata-rata di atas 85%. Meskipun terdapat sedikit penurunan kualitas deteksi pada kondisi pencahayaan minim menjelang sore hari akibat keterbatasan sensor optik, sistem secara umum mampu menjalankan fungsi utamanya sebagai pengamat otomatis tanpa kendala crash atau kegagalan penerimaan paket data pada sisi backend.

Dari seluruh rangkaian pengujian ini adalah pengukuran latensi sistem *end-to-end latency*, yang didefinisikan sebagai selisih waktu antara deteksi awal sensor hingga notifikasi diterima oleh petugas keamanan. Berdasarkan pencatatan log waktu selama jam operasional pabrik, rata-rata waktu tunda yang dibutuhkan sistem berkisar pada angka 5,4 detik. Durasi ini mencakup proses inisialisasi koneksi Wi-Fi, transmisi data citra, inferensi model di server, hingga komunikasi API ke platform Telegram. Meskipun dipengaruhi oleh fluktuasi sinyal nirkabel di area pabrik, jeda waktu tersebut dinilai masih sangat akseptabel dan memenuhi standar sistem peringatan dini, memberikan waktu reaksi yang memadai bagi petugas jaga di CV Indah Jaya Sentosa untuk melakukan verifikasi visual melalui smartphone sesaat sebelum kendaraan mencapai pos pemeriksaan.

Parameter Pengujian	Target Capaian	Hasil Terukur	Status
Akurasi Sensor	Trigger < 100 cm	Aktif pada 80 – 100 cm	Berhasil
Klasifikasi AI	Confidence $> 70\%$	Rata – rata 85%	Berhasil
Latensi Notifikasi	< 10 detik	Rata – rata 5 detik	Berhasil
Konektivitas	Stabil	Transmisi 100% sukses	Berhasil

Tabel 4.1 Rekapitulasi Hasil Pengujian.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan seluruh tahapan perancangan, implementasi, dan pengujian yang telah dilaksanakan di CV Indah Jaya Sentosa, dapat disimpulkan bahwa sistem deteksi kendaraan otomatis berbasis IoT berhasil dibangun dan berfungsi dengan baik sesuai tujuan penelitian. Integrasi antara sensor ultrasonik HC-SR04 dan modul kamera ESP32-CAM terbukti efektif dalam menciptakan mekanisme pemicu cerdas, di mana kamera hanya aktif mengakuisisi citra saat objek terdeteksi dalam radius kurang dari 100 cm, sehingga mampu mengefisiensikan konsumsi daya perangkat. Dari sisi komunikasi data, sistem berhasil mengirimkan citra visual melalui protokol HTTP POST ke server Flask dengan stabil, di mana server mampu memproses data tersebut, menyimpannya dalam arsip digital, serta mengklasifikasikan jenis kendaraan menggunakan model *Deep Learning* secara akurat. Muara dari keberhasilan sistem ini adalah terciptanya mekanisme notifikasi *real-time* melalui Bot Telegram, yang terbukti responsif memberikan informasi waktu dan bukti visual kepada petugas keamanan dengan rata-rata latensi 5,4 detik, menjadikannya solusi pengawasan yang andal untuk lingkungan perusahaan.

5.2 Saran

Sebagai bentuk evaluasi dari keterbatasan waktu dan ruang lingkup dalam pelaksanaan Kerja Praktik ini, terdapat beberapa saran untuk pengembangan sistem agar kinerja alat menjadi optimal:

1. Peningkatan Kualitas Kamera dan Pencahayaan: Modul kamera OV2640 pada ESP32-CAM memiliki keterbatasan sensitivitas cahaya, sehingga hasil deteksi menurun pada kondisi malam hari atau cuaca mendung. Disarankan untuk menambahkan pencahayaan eksternal (lampa sorot otomatis) atau mengganti modul kamera dengan tipe yang mendukung fitur *Night Vision* (lensa *wide-angle* dengan IR-Cut).
2. Pengembangan Fitur Identifikasi Plat Nomor: Sistem saat ini hanya mendeteksi keberadaan dan jenis kendaraan. Untuk pengembangan selanjutnya, disarankan mengintegrasikan teknologi *Automatic Number Plate Recognition* (ANPR) atau *Optical Character Recognition* (OCR) pada

server Flask, sehingga sistem tidak hanya mengirim foto kendaraan tetapi juga mencatat nomor polisi secara teks ke dalam *database*.

3. Implementasi pada Cloud Server: Saat ini pemrosesan data masih dilakukan pada server lokal (PC/Laptop). Agar sistem dapat diakses dan dipantau dari jarak jauh tanpa batasan jaringan lokal, disarankan untuk memindahkan *backend* Flask ke layanan *Cloud Hosting* atau VPS (Virtual Private Server) dengan IP Publik.
4. Pengemasan Alat yang Lebih Baik: Untuk penggunaan jangka panjang di lingkungan industri yang keras, disarankan merancang *Printed Circuit Board* (PCB) khusus untuk menggantikan kabel *jumper* guna meminimalisir risiko koneksi longgar akibat getaran kendaraan berat. Selain itu, penggunaan *casing* dengan standar industri yang lebih tinggi sangat direkomendasikan untuk perlindungan cuaca yang lebih baik.



DAFTAR PUSTAKA

- Ahadi Ningrum, A., & Ihsanudin, I. (2023). Penerapan Framework Flask Pada Machine Learning Dalam Memprediksi Umur Transformer. *Konvergensi*, 19(2), 51–59. <https://doi.org/10.30996/konv.v19i2.8239>
- Andrew, A., Buliali, J. L., & Wijaya, A. Y. (2017). Deteksi Kecepatan Kendaraan Berjalan di Jalan Menggunakan OpenCV. *Jurnal Teknik ITS*, 6(2), 366–371. <https://doi.org/10.12962/j23373539.v6i2.23489>
- Arrahma, S. A., & Mukhaiyar, R. (2023). Pengujian Esp32-Cam Berbasis Mikrokontroler. 4(1), 60–66.
- Aydos, M., Vural, Y., & Tekerek, A. (2019). Assessing risks and threats with layered approach to Internet of Things security. *Measurement and Control (United Kingdom)*, 52(5–6), 338–353. <https://doi.org/10.1177/0020294019837991>
- Bayu Unggul Sejati, & Hengky Triyo. (2025). Sistem Pemantauan Gudang Berbasis ESP32-Cam untuk Deteksi Gerakan dan Keberadaan Objek. *Just IT : Jurnal Sistem Informasi, Teknologi Informasi Dan Komputer*, 15(2), 356–361. <https://jurnal.umj.ac.id/index.php/just-it/index>
- CHYHIN, V., PAZYNIUK, M., TERENDII, O., & MENSHIKOV, O. (2023). CONTROLLING THE OPERATION OF THE REMOTE DEVICE USING FLASK PYTHON SERVER. *Herald of Khmelnytskyi National University. Technical Sciences*, 317(1). <https://doi.org/10.31891/2307-5732-2023-317-1-214-219>
- Curie, D. H., Jaison, J., Yadav, J., & Fiona, J. R. (2019). Analysis on web frameworks. *Journal of Physics: Conference Series*, 1362(1). <https://doi.org/10.1088/1742-6596/1362/1/012114>
- Dietz, H., Abney, D., Eberhart, P., Santini, N., Davis, W., Wilson, E., & McKenzie, M. (2022). ESP32-CAM as a programmable camera research platform. *IS and T International Symposium on Electronic Imaging Science and Technology*, 34(7). <https://doi.org/10.2352/EI.2022.34.7.ISS-232>
- Harjanta, A. T. J., & Dewanto, F. M. (2017). Real Time Tracking Object Moving with Webcam Based Color Using Background Subtraction Method. *Transformatika*, 15(1), 1–7. journals.usm.ac.id/index.php/transformatika
- Jonathan, R., & Setiyawati, N. (2023). Implementasi Framework Flask Pada Pembangunan Aplikasi Monitoring Dan Store Visit It Support Pada Pt. Xyz. *Jurnal Informatika*, 23(1), 91–101. <https://doi.org/10.30873/ji.v23i1.3637>
- Kornienko, D. V., Mishina, S. V., & Melnikov, M. O. (2021). The Single Page Application architecture when developing secure Web services. *Journal of Physics: Conference Series*, 2091(1). <https://doi.org/10.1088/1742-6596/2091/1/012065>
- Kumar, S., Tiwari, P., & Zymbler, M. (2019). Internet of Things is a revolutionary approach for future technology enhancement: a review. *Journal of Big Data*, 1(1). <https://doi.org/10.1186/s40537-019-0268-2>
- Lei, N. (2022). Intelligent logistics scheduling model and algorithm based on Internet of Things technology. *Alexandria Engineering Journal*, 61(1), 893–903. <https://doi.org/10.1016/j.aej.2021.04.075>
- Megantara, R. A., & Pramunendar, R. A. (2017). Pengembangan Background

- Subtraction Menggunakan FCM Untuk Deteksi Objek Bergerak Berdasarkan Pencahayaan Yang Bervariasi. *Techno.Com*, 16(4), 435–443. <https://doi.org/10.33633/tc.v16i4.1541>
- Moh Avin Annabil, Safhira Aulia Nurazizah, Zainatul Khasanah, Risqillah Ayu Puspita S, & Moh Avin Annabil. (2024). Analisis Pengukuran Jarak Objek Dengan Sensor Ultrasonik(HC-SR04) untuk Berbagai Bentuk Objek. *Journal of Electronics and Instrumentation*, 1(2), 54–61. <https://doi.org/10.19184/jei.v1i2.687>
- Paramitha, P., & Suartana, I. M. (2024). Analisis QoS Dan QoE Pada Video Streaming Berbasis IoT Menggunakan ESP32-CAM dan NGROK. *Journal of Informatics and Computer Science (JINACS)*, 5(04), 466–472. <https://doi.org/10.26740/jinacs.v5n04.p466-472>
- Prabowo, C., & Zurnawita. (2018). Penerapan Metode Background Subtraction Dengan Menggunakan Kandidat Sampling Background Applied Background Subtraction Method Used Background. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 5(6), 731–736. <https://doi.org/10.25126/jtiik201856115>
- Rahmawati, L., & Adi, K. (2017). Rancang bangun penghitung dan pengidentifikasi kendaraan menggunakan Multiple Object Tracking. *Youngster Physics Journal*, 6(1), 70–75.
- Raimundo, R. J., & Rosário, A. T. (2022). Cybersecurity in the Internet of Things in Industrial Management. *Applied Sciences (Switzerland)*, 12(3). <https://doi.org/10.3390/app12031598>
- Sapteka, A. A. N. G., Dharma, P. A. S., Widyatmika, K. A., Suparta, I. N., Yasa, I. M. S., & Sapteka, A. A. N. G. (2022). Pendekripsi Penggunaan Masker Wajah dengan ESP32Cam Menggunakan OpenCV dan Tensorflow. *Majalah Ilmiah Teknologi Elektro*, 21(2), 155. <https://doi.org/10.24843/mite.2022.v21i02.p01>
- Setiadi, A., Ahadi, A. H., Studi, P., Perangkat, R., Studi, P., Komputer, I., Studi, P., Informasi, T., & Hamzah, U. N. (2019). *Filter Gambar Berbasis Web Menggunakan Opencv Dan*. 30–34.
- Setyawan, G. E., Adiwijaya, B., & Fitriyah, H. (2019). Sistem Deteksi Jumlah, Jenis dan Kecepatan Kendaraan Menggunakan Analisa Blob Berbasis Raspberry Pi. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, 6(2), 211–218. <https://doi.org/10.25126/jtiik.2019621405>
- Sholihin, M., Adi Wibowo, S., & Primaswara Prasetya, R. (2021). PENERAPAN IoT (Internet of Things) TERHADAP RANCANG BANGUN SISTEM PERINGATAN BATASAN KECEPATAN DAN PENDETEKSI LOKASI KECELAKAAN BAGI PENGENDARA SEPEDA MOTOR BERBASIS ARDUINO. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 5(2), 597–604. <https://doi.org/10.36040/jati.v5i2.3743>
- Sudimanto. (2020). Rekayasa Perangkat Lunak Penghitung Jumlah Tempat Parkir Tersedia Menggunakan Kamera dengan Metode Background Subtraction. *Media Informatika*, 19(1), 1–5. <https://doi.org/10.37595/mediainfo.v19i1.35>
- Tri Laksono, H., Budiarto, Z., Lomba Juang, J., Sel, K., & Semarang, K. (2023). STRING (Satuan Tulisan Riset dan Inovasi Teknologi) Rancang Bangun Sistem Smart Parkir Berbasis Arduino. *STRING (Satuan Tulisan Riset Dan Inovasi*

Teknologi), 7(3), 1–5.
Zidane, C. Z., & Rahmadewi, R. (2022). *523103-Implementasi-Esp-32-Cam-Pada-Alat-Sistem-B7436F44*. 262–265.

