

**PERANCANGAN SISTEM PEOPLE COUNTING DAN DETEKSI
ANTRIAN MENGGUNAKAN TINY MACHINE LEARNING
(TINYML) PADA KAMERA IOT (ESP32-CAM) UNTUK
OPTIMALISASI MANAJEMEN RUANG
PADA COFFEESHOP**

SKRIPSI

DISUSUN OLEH

MUHAMMAD ARIQ ADRIAN

NPM. 2209020060



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2026

**PERANCANGAN SISTEM PEOPLE COUNTING DAN DETEKSI
ANTRIAN MENGGUNAKAN TINY MACHINE LEARNING
(TINYML) PADA KAMERA IOT (ESP32-CAM) UNTUK
OPTIMALISASI MANAJEMEN RUANG
PADA COFFEESHOP**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer (S.Kom) dalam Program Studi Teknologi Informasi. pada
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas
Muhammadiyah Sumatera Utara**

**MUHAMMAD ARIQ ADRIAN
NPM. ISI 2209020060**

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN
2026**

LEMBAR PENGESAHAN

Judul Skripsi : PERANCANGAN SISTEM PEOPLE COUNTING DAN
DETEKSI ANTRIAN MENGGUNAKAN TINY
MACHINE LEARNING (TINYML) PADA KAMERA
IOT (ESP32-CAM) UNTUK OPTIMALISASI
MANAJEMEN RUANG PADA COFFEESHOP

Nama Mahasiswa : MUHAMMAD ARIQ ADRIAN

NPM : 2209020060

Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



(Amrullah, S.Kom., M.Kom.)
NIDN. 0125118604

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom, M.Kom)
NIDN. 0117019301

Dekan



(Dr. Al-Khoyarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

PERANCANGAN SISTEM PEOPLE COUNTING DAN DETEKSI ANTRIAN MENGGUNAKAN TINY MACHINE LEARNING (TINYML) PADA KAMERA IOT (ESP32-CAM) UNTUK OPTIMALISASI MANAJEMEN RUANG PADA COFFEESHOP

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 10 April 2026

Yang membuat pernyataan



Muhammad Ariq Adrian

NPM. 2209020060

**PERNYATAAN PERSETUJUAN PUBLIKASI KARYA ILMIAH
UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Muhammad Ariq Adrian
NPM : 2209020060
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**PERANCANGAN SISTEM PEOPLE COUNTING DAN DETEKSI
ANTRIAN MENGGUNAKAN TINY MACHINE LEARNING
(TINYML) PADA KAMERA IOT (ESP32-CAM) UNTUK
OPTIMALISASI MANAJEMEN RUANG
PADA COFFEESHOP**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 10 April 2026

Yang membuat pernyataan



Muhammad Ariq Adrian

NPM. 2209020060

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Muhammad Ariq Adrian
Tempat dan Tanggal Lahir : Medan, 29 November 2004
Alamat Rumah : Jln. Psr Nippon Gg. Bestari Link.1
Telepon/Faks/HP : 081269699889
E-mail : muhammadariqadrian@gmail.com
Instansi Tempat Kerja :
Alamat Kantor :

DATA PENDIDIKAN

SD : Dr. Wahidin Sudirohusodo TAMAT: 2016
SMP : Pertiwi Medan TAMAT: 2019
SMA : Pertiwi Medan TAMAT: 2022

KATA PENGANTAR



Puji dan syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan skripsi ini dengan judul *“Perancangan Sistem People Counting dan Deteksi Antrian Menggunakan Tiny Machine Learning (TinyML) pada Kamera IoT (ESP32-CAM) untuk Optimalisasi Manajemen Ruang pada Coffeeshop”*. Skripsi ini disusun sebagai salah satu syarat untuk memperoleh gelar Sarjana pada Program Studi Teknologi Informasi.

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Akrim, M.Pd., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Assoc. Prof. Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Dr. Firaahmy Rizky, S.Kom., M.Kom Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi.
4. Ibu Fatma Sari Hutagalung, S.Kom, M.Kom Ketua Program Studi Teknologi Informasi yang telah membimbing dan mengarahkan penulis selama menempuh pendidikan di Program Studi Teknologi Informasi.
5. Bapak Okvi Nugroho, S.Kom., M.Kom Sekretaris Program Studi Teknologi Informasi.
6. Bapak Dosen Pembimbing, Amrullah, S.Kom.,M.Kom. yang dengan penuh kesabaran telah meluangkan waktu, tenaga, dan pikiran untuk memberikan bimbingan, masukan, serta arahan yang sangat berarti dalam penyusunan skripsi ini.
7. Ayah tercinta, yang senantiasa memberikan doa, kasih sayang, dukungan moral maupun material, serta menjadi teladan dalam semangat kerja keras dan tanggung jawab bagi penulis

8. Ibu tercinta, yang dengan penuh kasih sayang, kesabaran, dan doa yang tiada henti telah menjadi sumber kekuatan dan penyemangat utama bagi penulis dalam menyelesaikan skripsi ini.
9. Saudari Saudari penulis, adik Cut Aira Faiza, adik Cut Alya Octari yang telah memberikan semangat, kebersamaan, serta doa sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
10. Kerabat dan sahabat penulis, Dwi Nisyatul Wardah, M. Rifqy Aditya Lbs, Andhika Pratama Putra, Farhan Jihad, Almiftah Nazwa Khairunizzam, Rahil Lizar Altamis, Rahmat Hidayat Pane, Abdullah Yudi, Nabila Azura Putri, Nadya Aulya Putri, Gaizka Pasya Dermawan Sinukaban, yang dimana menjadi teman dan layaknya seperti kerabat, dan telah menemani perjalanan akademik penulis, melewati suka duka, memberikan dukungan, motivasi, serta kebersamaan yang berharga selama masa hidup penulis.
11. Coffeeshop langganan penulis, Ruang Kopi Koju marelan. Yang dimana telah menjadi tempat penulis bersuka cita dengan keluarga, teman, dan orang orang yang penulis cintai, hingga menjadi saksi penulis tumbuh kembang hingga dewasa dan sampai di tahap penyusunan skripsi
12. Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

PERANCANGAN SISTEM PEOPLE COUNTING DAN DETEKSI ANTRIAN
MENGUNAKAN TINY MACHINE LEARNING (TINYML) PADA
KAMERA IOT (ESP32-CAM) UNTUK OPTIMALISASI
MANAJEMEN RUANG PADA COFFEESHOP

ABSTRAK

Peningkatan jumlah pengunjung pada coffeeshop menuntut pengelolaan ruang dan antrian yang lebih efektif agar kenyamanan pelanggan tetap terjaga. Namun, proses pemantauan jumlah pengunjung dan kondisi antrian masih banyak dilakukan secara manual sehingga kurang efisien dan berpotensi menimbulkan ketidaktepatan dalam pengambilan keputusan. Oleh karena itu, penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem people counting dan deteksi antrian berbasis Internet of Things (IoT) dengan memanfaatkan Tiny Machine Learning (TinyML) pada perangkat ESP32-CAM.

Sistem yang dirancang menggunakan kamera ESP32-CAM sebagai perangkat akuisisi data visual, yang selanjutnya diproses secara on-device menggunakan model TinyML untuk mendeteksi dan menghitung jumlah orang serta kondisi antrian secara real-time. Data hasil deteksi kemudian ditampilkan melalui antarmuka monitoring sebagai informasi pendukung dalam pengelolaan ruang. Metode penelitian yang digunakan meliputi perancangan sistem, pengembangan model TinyML, implementasi perangkat keras dan perangkat lunak, serta pengujian kinerja sistem.

Hasil penelitian menunjukkan bahwa sistem mampu melakukan deteksi jumlah orang dan kondisi antrian dengan baik serta dapat berjalan secara real-time pada perangkat IoT dengan keterbatasan sumber daya. Sistem ini diharapkan dapat membantu pihak pengelola coffeeshop dalam mengoptimalkan manajemen ruang, meningkatkan efisiensi operasional, serta mendukung pengambilan keputusan berbasis data.

Kata Kunci: Internet of Things; TinyML; ESP-32CAM;

DESIGN OF A PEOPLE COUNTING AND QUEUE DETECTION SYSTEM
USING TINY MACHINE LEARNING (TINYML) ON AN IOT
CAMERA (ESP32-CAM) FOR SPACE MANAGEMENT
OPTIMIZATION IN A COFFEESHOP

ABSTRACT

The increasing number of visitors in coffee shops requires effective space and queue management to maintain customer comfort. However, monitoring the number of visitors and queue conditions is often still performed manually, resulting in inefficiency and potential inaccuracies in decision-making. Therefore, this study aims to design and implement an Internet of Things (IoT)-based people counting and queue detection system by utilizing Tiny Machine Learning (TinyML) on the ESP32-CAM device.

The proposed system employs the ESP32-CAM camera as a visual data acquisition device, where the data are processed directly on-device using a TinyML model to detect and count the number of people and queue conditions in real-time. The detection results are then presented through a monitoring interface as supporting information for space management. The research methodology includes system design, TinyML model development, hardware and software implementation, and system performance testing.

The results indicate that the system is capable of detecting the number of people and queue conditions effectively and operates in real-time on IoT devices with limited computational resources. This system is expected to assist coffee shop management in optimizing space utilization, improving operational efficiency, and supporting data-driven decision-making.

Keywords: Internet of Things;TinyML;ESP-32CAM;.

DAFTAR ISI

ABSTRAK	i
ABSTRACT	ii
DAFTAR ISI	iii
DAFTAR TABEL	v
DAFTAR GAMBAR	vi
BAB I. PENDAHULUAN	1
1.1. LATAR BELAKANG MASALAH	1
1.2. RUMUSAN MASALAH.....	3
1.3. BATASAN MASALAH.....	4
1.4. TUJUAN PENELITIAN.....	5
1.5. MANFAAT PENELITIAN.....	6
BAB II. LANDASAN TEORI	8
2.1. Internet of Things (IoT).....	8
2.2. Kamera IoT dan ESP32-CAM	9
2.3. Tiny Machine Learning (TinyML).....	11
2.4. Machine Learning dan Deep Learning	13
2.5. People Counting dan Deteksi Antrean	16
2.6. Dataset untuk People Counting.....	19
2.7. Model dan Framework TinyML	21
2.8. Sistem Optimalisasi Manajemen Ruang	22
2.9. Ringkasan Peneliti Terdahulu	25
2.10. Analisis Gap	28
BAB III. METODOLOGI PENELITIAN	31
3.1. Jenis dan Pendekatan Penelitian.....	31
3.2. Tahapan Penelitian	32
3.3. Lokasi dan Waktu Penelitian	35
3.4. Perancangan Sistem	36
3.5. Pengumpulan Data.....	43
3.6. Pra-pemrosesan Data	45
3.7. Perancangan dan Pelatihan Model.....	47
3.8. Optimasi dan Mekanisme Implementasi TinyML	49
3.9. Perancangan Sistem People Counting dan Deteksi Antrean.....	51
3.10. Metode Pengujian Sistem.....	54
3.11. Metode Evaluasi dan Analisis Data.....	58
3.12. Alat dan Bahan Penelitian.....	60
3.13. Etika dan Keamanan Data.....	62
3.14. Kendala dan Batasan Penelitian	63
3.15. Jadwal Penelitian.....	64
BAB IV. IMPLEMENTASI SISTEM	66
4.1. Implementasi Sistem	66
4.2. Implementasi Perangkat Keras.....	68
4.3. Implementasi Perangkat Lunak.....	71
4.4. Integrasi Sistem.....	76
4.5. Pengujian Sistem.....	79
4.6. Analisis Hasil	82
BAB V. KESIMPULAN DAN SARAN	86

5.1. Kesimpulan	86
5.2. Saran.....	86
DAFTAR PUSTAKA.....	88

DAFTAR TABEL

	HALAMAN
TABEL 2.1. Penelitian Terdahulu	31
TABEL 3.1. Rancangan Sistem Database	43
TABEL 3.2. Alat dan Bahan Penelitian	61
TABEL 3.3. Jadwal Penelitian	65
TABEL 4.1. Komponen Sistem dan Fungsi nya	68
TABEL 4.2. Spesifikasi Perangkat Keras Sistem	71
TABEL 4.3. Deskripsi Fungsi Utama Program	74
TABEL 4.4. Komponen Sistem Terintegrasi	77
TABEL 4.5. Hasil Pengujian Performa Sistem	81
TABEL 4.6. Evaluasi Akurasi Counting	83
TABEL 4.7. Evaluasi Deteksi Status Antrean	84

DAFTAR GAMBAR

	HALAMAN	
GAMBAR 3.1.	Flowchart Tahapan Penelitian	39
GAMBAR 3.2.	Diagram Arsitektur Sistem	42
GAMBAR 3.3.	Deployment Diagram Sistem	45
GAMBAR 3.4.	Wireframe Dashboard Sistem	46
GAMBAR 3.5.	Activity Diagram Sistem	53
GAMBAR 3.6.	Contoh Citra Manusia dari Dataset COCO	55
GAMBAR 3.7.	Contoh Citra Uji dari Dataset CrowdHuman dengan Variasi Jumlah Individu	55
GAMBAR 3.8.	Contoh Frame Hasil Ekstraksi Video Indoor yang Digunakan Dalam Pengujian Sistem.	58
GAMBAR 4.1.	Diagram Arsitektur Sistem	67
GAMBAR 4.2.	Flowchart Proses Sistem	67
GAMBAR 4.3.	Perangkat ESP32-CAM	69
GAMBAR 4.4.	Pemasangan Kamera ESP32-CAM pada Lokasi Coffeeshop	70
GAMBAR 4.5.	Diagram Blok Perangkat Keras	70
GAMBAR 4.6.	Hasil Training Model dengan Edge Impulse	73
GAMBAR 4.7.	Diagram Flowchart Software	74
GAMBAR 4.8.	Tampilan Web Dashboard	76
GAMBAR 4.9.	Screenshot Firebase Database Realtime	78
GAMBAR 4.10.	Screenshot Firebase Database Realtime	79
GAMBAR 4.11.	Blok Diagram Alur Integrasi Sistem	79
GAMBAR 4.12.	Live Streaming dan Dashboard	80
GAMBAR 4.13.	Serial Monitor Output Inferensi	82
GAMBAR 4.14.	Sinkronisasi Event ke Real-time Database	82

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Industri *Food & Beverage* (F&B), khususnya pada sektor *coffeeshop*, menghadapi tantangan dalam mengelola ruang secara efisien akibat tingginya mobilitas pelanggan, keterbatasan kapasitas, serta variasi jumlah pengunjung pada jam-jam tertentu. Pengelolaan jumlah pelanggan dan kondisi antrean yang masih dilakukan secara manual seringkali menimbulkan permasalahan seperti penumpukan antrean, ketidakpastian waktu tunggu, dan ketidakefisienan pemanfaatan ruang. Situasi ini berdampak pada penurunan kenyamanan pelanggan, penurunan kualitas layanan, hingga hilangnya peluang penjualan.

Sejalan dengan berkembangnya teknologi *Internet of Things (IoT)*, sistem *monitoring* otomatis berbasis sensor dan kamera telah menjadi solusi efektif dalam pengumpulan data secara real-time (Ashton, 2020). Salah satu perangkat yang paling banyak digunakan dalam penelitian *IoT* visual adalah ESP32-CAM, yaitu kamera *IoT* berdaya rendah yang mampu melakukan pemrosesan dasar secara mandiri. Namun, perangkat *IoT* seperti ESP32-CAM memiliki keterbatasan pada komputasi, memori, dan konsumsi daya, sehingga tidak dapat menjalankan model *machine learning* konvensional yang berukuran besar (Banbury et al., 2021).

Kemajuan teknologi *Tiny Machine Learning (TinyML)* membuka peluang penerapan pembelajaran mesin pada perangkat *IoT* bersumber daya rendah. *TinyML* memungkinkan model *deep learning* yang telah dioptimasi melalui teknik seperti *quantization* dan *pruning*, sehingga dapat dijalankan langsung pada perangkat *edge*

dengan memori < 1 MB (Warden & Situnayake, 2022). Teknologi ini terbukti efektif untuk aplikasi seperti *person detection*, *face recognition*, dan *object tracking* pada perangkat seperti ESP32 dan ARM *Cortex-M* (Dautov et al., 2022).

Dalam konteks *people counting*, berbagai penelitian telah mengembangkan metode berbasis *computer vision* menggunakan kamera CCTV dan *deep learning*. Penelitian terkini menunjukkan bahwa metode berbasis *convolutional neural networks* (CNN) efektif dalam melakukan *people counting* dan estimasi kerumunan pada berbagai tingkat kepadatan, baik pada lingkungan indoor maupun outdoor (Zhang et al., 2021; Li et al., 2022).

Namun, sebagian besar metode tersebut memerlukan perangkat berperforma tinggi atau pemrosesan berbasis *cloud*, yang tidak cocok diterapkan pada lingkungan kecil seperti *coffeeshop* yang membutuhkan perangkat *low-cost*, *low-power*, dan *on-device processing*.

Penelitian terbaru menunjukkan bahwa *TinyML* dapat diterapkan untuk *people detection* pada perangkat kecil, misalnya model *TensorFlow Lite Micro* yang dijalankan pada ESP32 untuk mendeteksi keberadaan manusia dengan konsumsi daya rendah (Gonçalves et al., 2021). Sementara itu, penelitian lain mengembangkan sistem *queue detection* berbasis *vision* untuk mendeteksi panjang antrean dan kondisi kepadatan (Khan et al., 2020), namun implementasinya masih mengandalkan server eksternal dan belum berjalan sepenuhnya di *edge device*.

Dengan demikian, penelitian yang mengintegrasikan *TinyML* dan ESP32-CAM untuk melakukan *people counting* dan deteksi antrean secara *real-time* masih sangat terbatas, terutama pada konteks operasional *coffeeshop* yang membutuhkan sistem sederhana, murah, dan hemat energi. Implementasi sistem seperti ini

diharapkan dapat membantu pemilik *coffeeshop* dalam melakukan optimasi manajemen ruang, seperti pengaturan kapasitas ruangan, distribusi staf, estimasi antrean, dan peningkatan kualitas pelayanan.

Berdasarkan kondisi tersebut, penelitian ini mengusulkan “Perancangan Sistem *People Counting* dan Deteksi Antrean Berbasis *Tiny Machine Learning* pada Kamera *IoT* (ESP32-CAM) untuk Optimalisasi Manajemen Ruang di *Coffeeshop*” sebagai solusi inovatif yang menggabungkan *IoT* visual dan *TinyML* guna menghasilkan sistem *monitoring* ruang yang cerdas, efisien, dan ekonomis.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana merancang dan membangun sistem *people counting* berbasis kamera *IoT* (ESP32-CAM) yang mampu mendeteksi dan menghitung jumlah orang secara *real-time* dengan akurasi yang memadai?
2. Bagaimana mengembangkan model *Tiny Machine Learning* (*TinyML*) yang ringan dan efisien untuk melakukan deteksi manusia (*person detection*) dan deteksi antrean (*queue detection*) sehingga dapat dijalankan langsung pada perangkat berdaya rendah seperti ESP32-CAM?
3. Bagaimana mengintegrasikan model *TinyML* yang telah dilatih ke dalam sistem ESP32-CAM dan memastikan proses inferensi dapat berjalan dengan latensi rendah, penggunaan memori yang sesuai, serta konsumsi daya yang optimal?
4. Bagaimana merancang alur sistem *monitoring* yang dapat menampilkan hasil deteksi jumlah orang dan panjang antrean secara *real-time*, sehingga dapat

mendukung proses pengambilan keputusan dalam manajemen ruang pada *coffeeshop*?

5. Sejauh mana sistem yang dikembangkan mampu meningkatkan efektivitas *monitoring* kapasitas ruangan dan deteksi antrean jika dibandingkan dengan metode manual atau sistem konvensional yang sebelumnya digunakan oleh *coffeeshop*?

1.3. Batasan Masalah

Agar penelitian lebih terfokus dan dapat diselesaikan secara terukur, maka penelitian ini dibatasi pada beberapa ruang lingkup sebagai berikut:

1. Perangkat keras yang digunakan terbatas pada modul kamera *IoT* ESP32-CAM, termasuk keterbatasan sumber daya komputasi seperti kapasitas RAM ± 520 KB dan *flash memory* internal. Penelitian tidak membahas implementasi pada perangkat *IoT* lain seperti Raspberry Pi, Jetson Nano, maupun kamera IP berperforma tinggi.
2. Model pembelajaran mesin yang digunakan adalah model *Tiny Machine Learning (TinyML)* yang telah dioptimasi (*quantized*) agar dapat dijalankan pada perangkat dengan kemampuan komputasi rendah. Penelitian tidak mencakup penggunaan model *deep learning* berukuran besar atau model yang membutuhkan GPU atau *server cloud* untuk inferensi.
3. Deteksi objek dibatasi hanya pada deteksi manusia (*person detection*) untuk tujuan *people counting* dan pendefinisian antrean. Objek lain seperti meja, kursi, barang, maupun kendaraan tidak dianalisis dalam penelitian ini.
4. Cakupan deteksi antrean (*queue detection*) dibatasi pada identifikasi panjang antrean berdasarkan jumlah manusia yang terdeteksi dalam area tertentu.

Sistem tidak menganalisis perilaku pelanggan, waktu tunggu individual, atau pola pergerakan kompleks.

5. Sistem *monitoring* yang dikembangkan hanya menampilkan informasi jumlah orang dan panjang antrean secara *real-time*. Analitik lanjutan seperti prediksi jumlah pengunjung, *forecasting* jam sibuk, maupun integrasi dengan sistem kasir (POS) tidak dibahas dalam ruang lingkup penelitian ini.

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk merancang dan mengembangkan sebuah sistem *people counting* dan deteksi antrean berbasis *Tiny Machine Learning (TinyML)* pada kamera *IoT ESP32-CAM*, yang diharapkan mampu meningkatkan efektivitas manajemen ruang pada sebuah *coffeeshop*. Secara khusus, tujuan penelitian ini meliputi:

1. Merancang arsitektur sistem *IoT* menggunakan perangkat *ESP32-CAM* yang mampu melakukan akuisisi gambar dan pemrosesan awal sebagai dasar untuk deteksi manusia dan pengukuran antrean secara *real-time*.
2. Mengembangkan dan melatih model *Tiny Machine Learning (TinyML)* untuk melakukan deteksi manusia (*person detection*) dan pendeteksian antrean dengan ukuran model yang ringan, efisien, serta sesuai dengan keterbatasan memori dan komputasi perangkat *edge* seperti *ESP32-CAM*.
3. Melakukan proses optimasi model, termasuk *quantization* dan konversi ke format *TensorFlow Lite Micro*, agar model dapat berjalan pada perangkat berdaya rendah dengan tetap mempertahankan tingkat akurasi yang memadai.
4. Mengintegrasikan model *TinyML* yang telah dioptimasi ke dalam *ESP32-CAM*, serta membangun sistem inferensi *on-device* yang memiliki latensi

rendah dan mampu menghitung jumlah orang serta panjang antrean tanpa perlu memproses data melalui server eksternal.

5. Merancang sistem *monitoring real-time* yang menampilkan informasi jumlah pengunjung dan kondisi antrean melalui *dashboard* atau *platform* berbasis web sehingga dapat mendukung pengambilan keputusan dalam manajemen ruang pada *coffeeshop*.

1.5. Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat sebagai berikut:

1. Manfaat bagi Universitas

Penelitian ini diharapkan dapat memberikan kontribusi akademik bagi universitas dan fakultas dalam pengembangan kajian di bidang *Internet of Things (IoT)*, *computer vision*, dan *Tiny Machine Learning (TinyML)*. Hasil penelitian dapat menjadi referensi ilmiah serta bahan pendukung pembelajaran dan penelitian lanjutan, khususnya terkait implementasi sistem cerdas berbasis perangkat edge berdaya rendah. Selain itu, penelitian ini dapat memperkuat portofolio riset fakultas dalam pengembangan teknologi terapan yang relevan dengan kebutuhan industri dan masyarakat.

2. Manfaat bagi Mitra (Ruang Kopi Koju)

Bagi mitra, penelitian ini diharapkan dapat memberikan solusi awal berupa sistem *people counting* dan deteksi antrean berbasis kamera *IoT* yang dapat dimanfaatkan sebagai alat bantu monitoring kapasitas ruangan dan kondisi antrean secara *real-time*. Informasi yang dihasilkan sistem dapat membantu mitra dalam meningkatkan efisiensi pengelolaan ruang, mendukung pengambilan keputusan operasional, serta menjadi dasar pengembangan

sistem *monitoring* yang lebih lanjut sesuai dengan kebutuhan lingkungan *coffeeshop*.

3. Manfaat bagi Mahasiswa

Penelitian ini memberikan manfaat bagi mahasiswa dalam meningkatkan pemahaman dan keterampilan teknis terkait perancangan sistem IoT, pengembangan model *Tiny Machine Learning*, serta penerapan *computer vision* pada perangkat dengan keterbatasan sumber daya. Selain itu, penelitian ini dapat menjadi sarana pembelajaran dalam menerapkan metodologi penelitian ilmiah secara sistematis, mulai dari perancangan, implementasi, hingga evaluasi sistem, sehingga dapat mendukung pengembangan kompetensi mahasiswa di bidang teknologi cerdas dan sistem tertanam.

BAB II

LANDASAN TEORI

2.1. Internet of Things (IoT)

Internet of Things (IoT) merupakan konsep integrasi berbagai perangkat fisik yang dibekali sensor, aktuator, modul komunikasi, dan kemampuan komputasi sehingga dapat saling berinteraksi melalui jaringan internet tanpa intervensi manusia secara langsung. Istilah *IoT* pertama kali dipopulerkan oleh Kevin Ashton pada tahun 1999, yang menekankan pentingnya penghubungan objek fisik ke dalam sistem informasi melalui identifikasi otomatis dan kemampuan komunikasi digital (Ashton, 2009). Seiring perkembangan teknologi miniaturisasi perangkat, komputasi berdaya rendah, dan konektivitas nirkabel, *IoT* telah berkembang menjadi salah satu fondasi utama dalam sistem cerdas modern.

Arsitektur *IoT* umumnya terdiri dari tiga lapisan, yaitu *perception layer*, *network layer*, dan *application layer*. Lapisan *perception* berfungsi mengumpulkan data dari sensor, kamera, atau perangkat fisik lainnya. Lapisan *network* mengirimkan data melalui protokol seperti *Wi-Fi*, *MQTT*, atau *HTTP*. Sementara itu, *application layer* menyediakan layanan pengolahan dan visualisasi data bagi pengguna. Penelitian terkini menekankan bahwa desain arsitektur *IoT* yang efisien berfokus pada pengurangan latensi, peningkatan reliabilitas sistem, serta minimisasi konsumsi daya, khususnya pada implementasi *edge computing* yang melibatkan perangkat dengan sumber daya terbatas (Shi et al., 2020; Abbas et al., 2021).

IoT telah diterapkan secara luas pada industri manufaktur, sistem *monitoring* lingkungan, *smart home*, transportasi, dan layanan publik. Dalam konteks pemantauan ruang dan pelanggan, *IoT* memungkinkan sistem mengamati kondisi ruangan secara otomatis, termasuk jumlah pengunjung, tingkat keramaian, dan pola mobilitas. Penelitian sebelumnya menunjukkan bahwa *IoT* berbasis kamera dapat meningkatkan efektivitas analisis perilaku pelanggan karena menyediakan data visual yang lebih akurat dibandingkan sensor konvensional (Chen et al., 2020).

Pada penelitian ini, *IoT* menjadi kerangka dasar yang menghubungkan perangkat kamera ESP32-CAM dengan modul *Tiny Machine Learning (TinyML)* untuk melakukan pemrosesan visual secara langsung pada perangkat (*on-device inference*). Pendekatan ini sejalan dengan tren *edge computing*, di mana proses komputasi dilakukan di perangkat *edge* guna mengurangi ketergantungan pada server *cloud*, menghemat *bandwidth*, dan meningkatkan keamanan data (Li et al., 2020). Sejumlah studi juga menegaskan bahwa kombinasi *IoT* dan *TinyML* memungkinkan pembangunan sistem cerdas yang hemat energi dan berbiaya rendah, terutama untuk perangkat kamera berdaya terbatas (Banbury et al., 2021).

Dengan demikian, *IoT* tidak hanya menjadi fondasi bagi integrasi perangkat ESP32-CAM dalam penelitian ini, tetapi juga berperan penting dalam mendukung proses pengumpulan data visual, analisis kondisi ruang, dan penyampaian informasi *people counting* serta deteksi antrean secara real-time untuk optimalisasi manajemen ruang pada *coffeeshop*.

2.2. Kamera IoT dan ESP32-CAM

ESP32-CAM merupakan modul kamera berbasis *mikrokontroler* ESP32 yang mengintegrasikan kemampuan pemrosesan, konektivitas *Wi-Fi/Bluetooth*, serta

sensor kamera OV2640 dalam satu paket yang berukuran kecil dan berdaya rendah. Modul ini dirancang untuk aplikasi *Internet of Things (IoT)* yang membutuhkan kemampuan pengambilan gambar atau video, seperti sistem pengawas ruangan, pengenalan objek, pemantauan lingkungan, dan berbagai aplikasi berbasis visi komputer. ESP32-CAM memiliki spesifikasi teknis yang mendukung operasi *IoT*, di antaranya prosesor *dual-core Tensilica LX6*, RAM internal terbatas namun dapat diperluas dengan PSRAM, serta dukungan antarmuka kamera menggunakan protokol SCCB/OV2640.

Perangkat ini banyak digunakan dalam sistem *IoT* karena menawarkan kombinasi antara harga yang ekonomis, konsumsi daya rendah, serta kemampuan pemrosesan yang cukup untuk aplikasi *edge computing* berbasis visi. Menurut penelitian oleh Nugraha & Darmawan (2020), ESP32-CAM terbukti efektif digunakan pada sistem pemantauan berbasis video karena kemampuannya mengirimkan citra secara *real-time* melalui jaringan Wi-Fi. Penelitian lain menunjukkan bahwa integrasi kamera OV2640 pada ESP32-CAM menghasilkan kualitas citra yang memadai untuk tugas deteksi objek skala ringan, meskipun dengan keterbatasan resolusi dan kecepatan pemrosesan (Putra et al., 2021).

Keunggulan utama ESP32-CAM pada aplikasi *Tiny Machine Learning* terletak pada dukungannya terhadap model *TensorFlow Lite Micro* yang memungkinkan inferensi berjalan langsung dalam perangkat (*on-device inference*). Hal ini memungkinkan sistem melakukan deteksi sederhana tanpa harus mengirim seluruh citra ke server eksternal. Banbury et al. (2021) menunjukkan bahwa teknik optimasi model seperti *quantization* dapat mengurangi ukuran model hingga 4×, memungkinkan model deteksi berjalan pada perangkat *mikrokontroler* terbatas

seperti ESP32. Beberapa studi juga mengaplikasikan ESP32-CAM untuk tugas deteksi objek dan penghitungan orang, dengan hasil yang menunjukkan tingkat akurasi cukup baik pada kondisi cahaya stabil dan latar belakang tidak terlalu kompleks (Sasmoko et al., 2022).

Dalam konteks IoT, ESP32-CAM berfungsi sebagai perangkat edge yang menggabungkan fungsi *data acquisition* (melalui kamera) dan *data processing* (melalui *TinyML*). Pendekatan ini mendukung arsitektur edge computing yang lebih efisien karena dapat mengurangi beban jaringan dan meningkatkan keamanan data visual. Li et al. (2020) menegaskan bahwa pemrosesan di *edge device* memiliki manfaat signifikan dalam mengurangi latensi, terutama pada aplikasi real-time seperti deteksi orang dan analisis antrean.

Dengan demikian, ESP32-CAM merupakan perangkat yang sesuai untuk penelitian ini karena mampu menangani pengambilan gambar, menjalankan model *TinyML*, serta mengirimkan hasil deteksi melalui jaringan *IoT*. Kombinasi tersebut menjadikan ESP32-CAM pilihan ideal untuk pengembangan sistem people counting dan deteksi antrean yang ekonomis, mudah diimplementasikan, dan dapat diterapkan pada lingkungan operasional seperti *coffeeshop*.

2.3. Tiny Machine Learning (TinyML)

Tiny Machine Learning (TinyML) merupakan cabang dari *machine learning* yang berfokus pada pengembangan, optimasi, dan penerapan model kecerdasan buatan pada perangkat dengan sumber daya terbatas seperti *mikrokontroler*, sensor pintar, dan perangkat *IoT*. *TinyML* memungkinkan model pembelajaran mesin dijalankan langsung pada *edge device* dengan konsumsi daya sangat rendah biasanya hanya dalam kisaran milliwatt serta kapasitas memori yang jauh lebih

kecil dibandingkan perangkat komputasi konvensional seperti laptop atau server (Warden & Situnayake, 2020).

Konsep *TinyML* berkembang pesat seiring meningkatnya kebutuhan akan sistem cerdas yang dapat bekerja tanpa ketergantungan pada *cloud computing*. Pemrosesan langsung di perangkat (*on-device inference*) memberikan beberapa keunggulan, antara lain latensi rendah, privasi lebih baik, konsumsi *bandwidth* lebih hemat, serta kemampuan beroperasi meskipun perangkat tidak memiliki koneksi internet yang stabil (Banbury et al., 2021). Oleh karena itu, *TinyML* banyak diadopsi dalam aplikasi seperti deteksi objek sederhana, pengenalan suara, klasifikasi sensor, dan *vision-based monitoring*.

Dalam implementasinya, *TinyML* umumnya memanfaatkan *framework TensorFlow Lite Micro (TFLM)*, yang dirancang untuk menjalankan model *machine learning* pada perangkat dengan memori yang sangat terbatas, biasanya di bawah 256 KB RAM. *Framework* ini mendukung teknik optimasi seperti *quantization* misalnya *post-training quantization* dan *full integer quantization* yang dapat menurunkan ukuran model hingga 4–10 kali lebih kecil tanpa kehilangan akurasi signifikan (Jacob et al., 2018). Teknik ini sangat penting karena model *deep learning* standar, seperti *Convolutional Neural Network (CNN)*, pada umumnya memiliki ukuran terlalu besar untuk dijalankan pada mikrokontroler. Sehingga didapatkan persamaan rumus ukuran model setelah *quantization*

$$\text{Model Size}_{\text{INT8}} \approx 41 \times \text{Model Size}_{\text{FP32}} \quad (1)$$

Sejumlah penelitian menunjukkan keberhasilan penerapan *TinyML* untuk tugas visi komputer berbasis kamera berdaya terbatas. Penelitian oleh David et al. (2020) menunjukkan bahwa model deteksi objek skala ringan seperti *MobileNet*

dapat berhasil diporting ke *TFLite Micro* dengan performa inferensi yang tetap stabil pada perangkat *edge*. Studi lain oleh Sasmoko et al. (2022) mengimplementasikan *TinyML* pada ESP32-CAM untuk mendeteksi keberadaan objek di ruangan, dengan hasil bahwa model kuantisasi integer dapat berjalan dengan kecepatan inferensi yang memadai untuk kebutuhan *real-time monitoring*.

Dalam konteks *people counting* dan deteksi antrean, *TinyML* memberikan peluang signifikan karena dapat memproses citra secara lokal tanpa mengirimkan video ke server, sehingga lebih efisien dan menjaga privasi pengunjung. Beberapa penelitian terbaru menekankan bahwa penggunaan *TinyML* pada sistem pemantauan ruangan dapat meningkatkan efisiensi operasional dan mendukung pengambilan keputusan berbasis data dalam lingkungan komersial (Rahman et al., 2023).

Dengan demikian, *TinyML* menjadi komponen kunci dalam penelitian ini, karena memungkinkan model deteksi orang berjalan langsung pada perangkat ESP32-CAM yang memiliki keterbatasan memori dan komputasi. Kombinasi antara efisiensi model, kemampuan *edge computing*, dan konektivitas *IoT* menjadikan *TinyML* sebagai teknologi yang tepat untuk mewujudkan sistem *people counting* dan deteksi antrean yang ringan, murah, dan dapat digunakan dalam operasional nyata seperti pada *coffeeshop*.

2.4. Machine Learning dan Deep Learning

Machine Learning (ML) merupakan cabang kecerdasan buatan yang memungkinkan komputer mempelajari pola dari data dan membuat prediksi tanpa harus diprogram secara eksplisit. Algoritma *ML* bekerja dengan mengolah data pelatihan untuk menghasilkan model yang mampu mengenali pola tertentu,

melakukan klasifikasi, regresi, atau pengambilan keputusan secara otomatis (Mitchell, 1997). Perkembangan *ML* semakin penting seiring meningkatnya ketersediaan data digital, daya komputasi, serta kebutuhan akan sistem yang mampu melakukan analisis secara cepat dan akurat.

Di sisi lain, *Deep Learning (DL)* merupakan subbidang dari *ML* yang meniru struktur jaringan saraf biologis melalui *Artificial Neural Networks (ANN)* yang terdiri atas banyak lapisan (*deep architecture*). *DL* mencapai performa unggul dalam tugas-tugas kompleks seperti visi komputer, pengenalan suara, pemrosesan bahasa alami, dan sistem rekomendasi. Salah satu keunggulan utama *DL* adalah kemampuannya melakukan *feature extraction* secara otomatis tanpa perlu rekayasa fitur manual (LeCun, Bengio, & Hinton, 2015).

Salah satu model *DL* yang paling banyak digunakan dalam pemrosesan citra adalah *Convolutional Neural Network (CNN)*. *CNN* dirancang untuk menangkap pola spasial dan tekstur dalam citra melalui operasi konvolusi. Penelitian oleh Krizhevsky et al. (2012) menunjukkan keberhasilan *CNN* dalam kompetisi *ImageNet* melalui model *AlexNet*, yang menandai kemajuan besar dalam akurasi deteksi objek. Keunggulan *CNN* dalam pengenalan pola visual menjadikannya pilihan utama untuk tugas *people counting*, deteksi objek, dan analisis antrean, yang dimana memiliki rumus dasarnya.

$$Y(i, j) = \sum_{m, n} X(i + m, j + n) \cdot K(m, n) \quad (2)$$

Dalam konteks sistem *IoT*, penggunaan *ML* dan *DL* seringkali dibatasi oleh keterbatasan perangkat *edge* yang memiliki memori dan daya komputasi terbatas. Oleh karena itu, model *DL* yang digunakan harus menjalani proses optimasi seperti *pruning*, *quantization*, atau penggunaan *lightweight architecture* seperti *MobileNet*,

SqueezeNet, atau *ShuffleNet* (Howard et al., 2020). Teknik-teknik ini memungkinkan model tetap efisien tanpa pengurangan akurasi yang signifikan saat diimplementasikan pada perangkat berspesifikasi rendah.

Beberapa penelitian menunjukkan efektivitas penggunaan model *DL* ringan pada perangkat *IoT* dan *mikrokontroler*. Misalnya, David et al. (2020) berhasil melakukan porting model *CNN* kecil ke *TensorFlow Lite Micro* agar dapat berjalan pada platform *embedded*. Sementara itu, studi oleh Zhang et al. (2019) menunjukkan bahwa model *MobileNetV2* dapat mencapai performa inferensi yang andal pada perangkat *ARM Cortex-M* dengan memori terbatas.

Penerapan *ML* dan *DL* dalam konteks *people counting* dan deteksi antrean juga telah banyak diteliti. Penelitian oleh Chan et al. (2008) memperkenalkan metode *density-based counting* menggunakan pendekatan *learning-based*, sementara Zhang et al. (2016) mengembangkan model *CNN* yang mampu mengekstraksi peta kepadatan (*density map*) untuk menghitung jumlah orang pada kerumunan padat. Studi-studi tersebut memperkuat dasar teori bahwa metode berbasis *DL* sangat efektif untuk menghitung jumlah orang dalam berbagai kondisi pencahayaan, sudut kamera, dan tingkat keramaian.

Dalam penelitian ini, konsep *ML* dan *DL* digunakan sebagai dasar dalam membangun model penghitungan orang dan deteksi antrean. Meskipun model *CNN* konvensional terlalu besar untuk dijalankan pada modul ESP32-CAM, pendekatan optimasi melalui *TinyML* memungkinkan model yang telah diperkecil tetap dapat berjalan secara lokal. Integrasi *ML/DL* dengan teknologi *IoT* dan *edge computing* menjadi aspek fundamental dalam sistem yang dikembangkan, sehingga dapat melakukan deteksi secara *real-time*, efisien, dan tanpa ketergantungan pada *cloud*.

2.5. People Counting dan Deteksi Antrean

People counting adalah proses kuantifikasi jumlah individu yang berada dalam suatu area tertentu pada waktu tertentu menggunakan sensor atau sistem visi komputer. Teknik ini memainkan peran penting dalam manajemen ruang, analisis aliran pengunjung, perencanaan sumber daya, dan pengambilan keputusan operasional pada lingkungan seperti *coffeeshop*, pusat perbelanjaan, dan transportasi umum. Pendekatan *people counting* umumnya dapat dibagi menjadi dua kategori besar: berbasis sensor bukan-visual (mis. sensor tekanan, sensor IR, sensor pintu) dan berbasis visi komputer (*camera-based*). Pendekatan berbasis visi menawarkan keunggulan dalam hal detail spasial dan fleksibilitas analitik (mis. menghitung arus masuk/keluar, mendeteksi kerumunan pada zona tertentu), namun juga membawa tantangan terkait privasi, kondisi pencahayaan, dan kebutuhan komputasi (Zhang et al., 2016; Chen et al., 2018).

Metode *people counting* berbasis visi meliputi beberapa teknik utama:

1. Deteksi objek per-frame (*object detection*) yang menghitung jumlah bounding box bertipe `person`;
2. Pelacakan objek (*multi-object tracking, MOT*) yang menyatukan deteksi antar-frame untuk menghitung arus masuk/keluar.
3. Estimasi kepadatan (*density estimation*) yang menghasilkan peta kepadatan (*density map*) dan cocok untuk kerumunan padat.

Masing-masing pendekatan memiliki *trade-off* yang berbeda. Metode *object detection* yang dikombinasikan dengan *multi-object tracking (MOT)* umumnya memberikan akurasi yang baik pada kondisi jarak pandang yang jelas serta tingkat kepadatan rendah hingga sedang. Sebaliknya, pendekatan *density estimation*

cenderung lebih *robust* pada kondisi kepadatan tinggi, namun memerlukan anotasi peta kepadatan (*density map*) pada tahap pelatihan yang meningkatkan kompleksitas proses anotasi data (Zhang et al., 2021; Li et al., 2022).

Deteksi antrean (*queue detection*) adalah ekstensi aplikasi *people counting* yang fokus pada identifikasi pembentukan barisan, pengukuran panjang antrean, dan metrik terkait seperti waktu tunggu rata-rata dan tingkat kepadatan di area layanan (*counter/kasir*). Pendekatan vision-based untuk *queue detection* sering menggabungkan:

1. *Region-of-interest (ROI)* atau zona antrean yang ditentukan secara manual;
2. Hitungan orang di dalam *ROI*;
3. Analisis spasial untuk menentukan jarak antar-individu dan pembentukan baris; serta
4. Pola temporal untuk menentukan durasi dan dinamika antrean (Khan et al., 2020).

Teknik berbasis visi juga memungkinkan pengukuran metrik tambahan seperti kecepatan pergerakan rata-rata, arah aliran pelanggan, dan distribusi kepadatan spasial yang berguna untuk optimasi *layout* dan penjadwalan staf.

Implementasi *people counting* dan *queue detection* pada perangkat edge berdaya rendah (mis. ESP32-CAM) memerlukan perhatian khusus terhadap desain model dan *pipeline* pemrosesan. Tantangan teknis utama meliputi keterbatasan memori, laju inferensi yang harus rendah agar *real-time*, dan *robustness* terhadap kondisi operasional (pencahayaan variatif, *occlusion*, variasi pose). Untuk lingkungan seperti *coffeeshop* area sempit, variasi sudut pandang (atas, diagonal), dan sejumlah *occlusion* akibat furnitur beberapa strategi teknis praktis direkomendasikan:

1. Pemilihan metode: untuk *coffeeshop* dengan kepadatan rendah-sedang, kombinasi *object detection* ringan (mis. MobileNet-SSD lite) dengan sederhana *MOT* berbasis *IOU* dapat memberikan akurasi dan efisiensi terbaik. Untuk situasi kepadatan tinggi, gunakan pendekatan *density estimation* meskipun sulit di-*porting* ke *TinyML* tanpa optimasi lanjut.
2. Zona Antrean (*ROI*): tentukan *ROI* untuk area kasir/barista sehingga hitungan terfokus pada area relevan dan mengurangi *false positive* dari area lain.
3. Post-processing sederhana: gunakan *filtering temporal* (mis. *majority voting*, *temporal smoothing*) untuk mengurangi fluktuasi hitungan akibat deteksi sesaat atau *noise*.
4. Kalibrasi perspektif: gunakan koreksi perspektif atau threshold ukuran *bounding box* untuk membedakan orang dekat kamera versus jauh agar hitungan lebih akurat pada sudut pandang miring.
5. Optimasi model untuk *edge*: lakukan *quantization (INT8)*, *pruning*, dan *knowledge distillation* untuk mengecilkan model tanpa kehilangan akurasi signifikan sehingga dapat berjalan pada ESP32-CAM (Banbury et al., 2021; Jacob et al., 2018).

Evaluasi sistem *people counting* dan *queue detection* umumnya menggunakan metrik berikut: *Mean Absolute Error (MAE)* dan *Mean Squared Error (MSE)* untuk *people counting*, *precision/recall* dan *mAP* untuk deteksi objek, serta metrik domain antrean seperti akurasi estimasi panjang antrean dan waktu tunggu estimasi (Zhang et al., 2016; Khan et al., 2020). Selain metrik numerik, uji lapangan dalam kondisi nyata (*field test*) sangat penting untuk mengukur keandalan sistem terhadap variasi lingkungan.

Dalam konteks penelitian ini, target desain adalah sistem yang dapat:

1. Menghitung jumlah pengunjung secara *real-time* dengan akurasi yang memadai pada kondisi *coffeeshop* normal;
2. Mengidentifikasi pembentukan antrean pada ROI yang ditentukan;
3. Berjalan *on-device* pada ESP32-CAM setelah proses optimasi *TinyML*; dan
4. Menyediakan *output* terstruktur (jumlah, status antrean, *timestamp*) untuk integrasi ke *dashboard* manajemen ruang.

Dengan pendekatan yang tepat, kombinasi visi komputer, algoritma pelacakan sederhana, dan optimasi model memungkinkan sistem yang hemat biaya dan praktis untuk *deployment* di *coffeeshop* (Mundhenk et al., 2020; Rahman et al., 2023).

2.6. Dataset untuk People Counting

Dataset memegang peranan penting dalam proses pelatihan model deteksi orang dan estimasi antrean. Kualitas dan relevansi dataset sangat memengaruhi kemampuan model dalam mengenali objek manusia pada berbagai kondisi, terutama pada lingkungan indoor seperti *coffeeshop*. Model *computer vision* umumnya dilatih menggunakan dataset publik berskala besar, yang menyediakan keragaman pose manusia, tingkat kepadatan, serta variasi pencahayaan.

Dataset *COCO* (*Common Objects in Context*) merupakan salah satu dataset standar industri yang digunakan dalam pengembangan sistem deteksi objek modern. *COCO* menyediakan lebih dari 200.000 gambar dengan anotasi objek manusia yang luas (Lin et al., 2014), sehingga sering digunakan sebagai basis pretraining untuk model deteksi. Selain itu, dataset *CrowdHuman* dirancang khusus untuk skenario dengan kepadatan tinggi dan *overlapping* antar objek manusia.

Penelitian oleh Shao et al. (2018) menunjukkan bahwa *CrowdHuman* dapat meningkatkan *robustness* model terhadap situasi “*crowded scenes*” yang kompleks. Dataset lain yang sering digunakan adalah *Open Images* Dataset, yang menawarkan lebih dari 9 juta gambar dengan anotasi *bounding box* yang beragam (Kuznetsova et al., 2020), menjadikannya sumber data tambahan yang relevan untuk memperluas generalisasi model.

Meskipun demikian, sebagian besar dataset tersebut dikumpulkan dari ruang publik seperti jalan raya, pusat transportasi, dan area terbuka sehingga tidak sepenuhnya mencerminkan kondisi *indoor* pada ruang terbatas seperti *coffeeshop*. Penelitian dalam domain smart retail menegaskan bahwa lingkungan *indoor* memiliki karakteristik unik seperti pencahayaan tidak stabil, ruangan sempit, serta sudut pandang kamera yang tetap (Zhao et al., 2019). Hal ini menyebabkan model yang dilatih menggunakan dataset umum sering mengalami penurunan akurasi ketika di-*deploy* pada aplikasi *indoor*, sebuah fenomena yang disebut sebagai *domain gap*.

Tantangan yang biasanya muncul pada dataset khusus ruang *café* antara lain:

1. Fluktuasi intensitas cahaya (siang–malam, lampu kuning hangat, refleksi kaca).
2. Posisi pelanggan yang saling menutupi (*occlusion*).
3. Variasi pose yang tidak ekstrem tetapi sering terhalang meja dan furnitur.
4. Resolusi input rendah akibat keterbatasan lensa ESP32-CAM.

Untuk mengurangi *domain gap*, beberapa penelitian merekomendasikan fine-tuning menggunakan data khusus lingkungan target. Pendekatan ini terbukti meningkatkan akurasi deteksi pada ruang *indoor*, seperti dijelaskan dalam studi

domain *adaptation* untuk aplikasi retail dan café (Gong et al., 2019; Wang et al., 2021). Dengan demikian, pemilihan dataset yang tepat dan penyesuaian dataset terhadap kondisi nyata menjadi langkah penting dalam membangun sistem *people counting* berbasis *TinyML*.

2.7. Model dan Framework TinyML

Framework TinyML menjadi dasar pengembangan model pembelajaran mesin yang mampu berjalan pada perangkat mikro seperti ESP32-CAM. Dalam konteks ini, *TinyML* berfokus pada efisiensi memori, latensi rendah, dan kemampuan inferensi lokal tanpa bergantung pada server. Berbagai penelitian menyebutkan bahwa kombinasi *TensorFlow Lite*, *TensorFlow Lite Micro*, dan teknik optimasi seperti *quantization* merupakan pendekatan yang paling banyak digunakan untuk menjalankan model pada perangkat dengan keterbatasan komputasi (Banbury et al., 2021).

TensorFlow Lite (TFLite) berfungsi sebagai *framework* untuk mengonversi model *deep learning* berukuran besar menjadi model yang lebih ringan dan dapat dijalankan pada perangkat *edge*. Penelitian oleh David et al. (2020) menunjukkan bahwa *TFLite* dapat mengompres ukuran model secara signifikan melalui teknik optimasi graf, reduksi operator, dan kompresi parameter. Sementara itu, *TensorFlow Lite Micro (TFLite Micro)* merupakan interpreter statis yang dirancang khusus untuk perangkat ber-RAM kecil seperti ESP32, sehingga tidak memerlukan *dynamic memory allocation*. Hal ini memungkinkan model bekerja konsisten pada perangkat 32-bit dengan memori sangat terbatas.

Salah satu teknik penting dalam *TinyML* adalah *quantization*, yakni proses mengubah representasi numerik *floating point* menjadi format lebih ringan seperti

int8. Menurut Jacob et al. (2018), quantization dapat mengurangi ukuran model hingga $4\times$ lebih kecil dan mempercepat inferensi $2-3\times$ lebih cepat pada perangkat *edge*, tanpa dampak besar pada akurasi. Oleh karena itu, teknik ini sangat relevan untuk aplikasi *real-time* seperti *people counting*.

Alur proses deployment model *TinyML* pada ESP32-CAM meliputi:

1. Pelatihan model menggunakan *TensorFlow*.
2. Konversi ke format `.tflite`.
3. Optimasi melalui *post-training quantization* atau *quantization-aware training*.
4. Integrasi model ke program Arduino/ESP-IDF menggunakan *library TFLite Micro*.
5. Alokasi *tensor arena* sebagai memori kerja model.
6. Inferensi dijalankan langsung dari input kamera ESP32-CAM.

Peneliti *TinyML* melaporkan bahwa pipeline tersebut menghasilkan manfaat seperti:

1. Latensi inferensi rendah karena tidak membutuhkan *server cloud* (Warden & Situnayake, 2020).
2. Privasi lebih baik, karena data video tidak keluar dari perangkat.
3. Konsumsi daya rendah, cocok untuk perangkat *IoT* skala kecil.

Dengan struktur dan kemampuan tersebut, *framework TinyML* mendukung pengembangan sistem *people counting* dan deteksi antrian yang akurat, efisien, dan dapat berjalan secara *real-time* pada ESP32-CAM.

2.8. Sistem Manajemen Ruang pada Coffeeshop

Sistem manajemen ruang pada *coffeeshop* merupakan pendekatan strategis untuk mengatur kapasitas tempat duduk, alur pergerakan pelanggan, distribusi

antrian, serta efisiensi pelayanan dengan tujuan meningkatkan pengalaman pelanggan dan memaksimalkan penggunaan ruang. Dalam literatur manajemen layanan, efisiensi tata ruang menjadi salah satu faktor utama yang mempengaruhi kepuasan pelanggan dan durasi interaksi di dalam *café* (Bitner, 1992). Oleh karena itu, pengelolaan ruang yang terukur dan berbasis data menjadi semakin penting, terutama pada *coffeeshop* dengan kapasitas terbatas.

Konsep *customer flow* dalam ruang *café* berkaitan dengan bagaimana pelanggan bergerak mulai dari pintu masuk, area pemesanan, kasir, meja tunggu, hingga area duduk. Pengaturan yang tidak optimal dapat menyebabkan *bottleneck*, antrean panjang, dan peningkatan waktu tunggu. Penelitian dalam *smart retail* oleh Chih-Lun et al. (2020) menunjukkan bahwa pemahaman terhadap pola pergerakan pelanggan dapat meningkatkan efisiensi operasional dan memberikan insight terhadap perilaku konsumen. Pada *coffeeshop*, pola aliran pelanggan dipengaruhi oleh penempatan counter, jarak antar meja, kepadatan pengunjung, serta ritme waktu sibuk (*peak hour*). Indikator optimalisasi ruang yang umum digunakan meliputi:

1. Tingkat okupansi kursi (*seat occupancy rate*).
2. Pola antrean dan waktu tunggu pelanggan.
3. Pergantian pelanggan per satuan waktu (*customer turnover*).
4. Kesesuaian kapasitas dengan permintaan pada jam sibuk.
5. Efektivitas pergerakan staf dan pelanggan dalam area pelayanan.

Penelitian oleh Kim & Lee (2020) menegaskan bahwa sistem pemantauan berbasis data mampu meningkatkan efisiensi penggunaan ruang melalui

penyesuaian dinamika operasional, seperti pengalokasian staf atau penataan ulang furnitur untuk mengurangi kemacetan (*congestion*) di titik-titik tertentu.

Dalam konteks teknologi, sistem *people counting* memainkan peranan penting dalam menyajikan data aktual mengenai jumlah pelanggan dalam ruangan. Sistem ini memungkinkan pengelola *café* untuk melakukan monitoring kapasitas secara *real-time*, mendeteksi terjadinya kepadatan berlebih, serta memprediksi kebutuhan staf pada waktu-waktu sibuk. Selain itu, integrasi *people counting* dengan deteksi antrean dapat meningkatkan ketepatan estimasi waktu tunggu pelanggan, sebagaimana dibahas dalam penelitian sistem antrian berbasis visi komputer oleh Chen et al. (2021), yang menunjukkan bahwa pengukuran antrean secara otomatis dapat meningkatkan efisiensi pelayanan sebesar 20–40% pada lingkungan retail.

Hubungan antara *people counting* dan efisiensi pelayanan bersifat langsung. Dengan mengetahui jumlah pelanggan dan intensitas antrean, manajemen dapat mengoptimalkan:

1. Kecepatan proses pemesanan.
2. Penempatan barista atau kasir tambahan pada jam sibuk.
3. Penataan ruang untuk mengurangi penumpukan orang.
4. Alur keluar–masuk pelanggan agar tidak terjadi kemacetan.

Lebih lanjut, studi oleh Zhang et al. (2022) juga menunjukkan bahwa penggunaan data okupansi ruang membantu bisnis F&B dalam melakukan *demand forecasting*, penyesuaian jadwal operasional, serta pengendalian kualitas layanan.

Secara keseluruhan, sistem manajemen ruang yang terintegrasi dengan teknologi *people counting* memberikan dasar penting dalam meningkatkan efisiensi

operasional, mengurangi waktu tunggu, dan meningkatkan kenyamanan pelanggan di *coffeeshop*. Pendekatan berbasis data ini juga sejalan dengan tren implementasi *smart café* dan smart retail yang semakin berkembang dalam industri layanan modern.

2.9. Ringkasan Penelitian Terdahulu

Penelitian mengenai *people counting* dan deteksi antrean telah berkembang pesat seiring meningkatnya kebutuhan sistem *monitoring* cerdas di berbagai sektor seperti retail, transportasi, dan perkantoran. Metode yang digunakan umumnya berbasis kamera dan visi komputer, karena pendekatan ini menawarkan fleksibilitas tinggi serta akurasi yang baik untuk memantau pergerakan manusia secara *real-time*.

Penelitian *people counting* berbasis kamera banyak memanfaatkan model deteksi objek modern seperti *YOLO*, *SSD*, dan *Faster R-CNN*. Redmon & Farhadi (2020) menunjukkan bahwa *YOLOv3* mampu melakukan deteksi objek secara *real-time* dengan akurasi tinggi. Model serupa digunakan oleh Chen et al. (2021) dalam sistem *counting* untuk area publik, dan hasilnya memperlihatkan peningkatan akurasi ketika *good lighting* dan sudut kamera optimal digunakan. Pendekatan berbasis kamera juga banyak dipilih karena mampu mengatasi keterbatasan sensor inframerah atau PIR yang kurang efektif ketika objek bergerak dalam jumlah banyak.

Dalam beberapa tahun terakhir, penelitian *people counting* berbasis *TinyML* mulai berkembang. *TinyML* memungkinkan model *deep learning* dijalankan pada perangkat mikro berkonsumsi daya rendah, termasuk ESP32. Penelitian oleh David et al. (2020) menunjukkan bahwa model *TFLite* yang dioptimasi dengan

quantization dapat berjalan dengan *footprint* memori kecil tanpa kehilangan akurasi signifikan. Sementara itu, Imteaj et al. (2021) mendemonstrasikan implementasi *TinyML* pada perangkat *IoT* untuk deteksi manusia berbasis *edge computing*, menghasilkan sistem yang efisien dan tidak bergantung pada server *cloud*.

Penelitian mengenai deteksi antrian juga banyak ditemukan dalam *smart retail*. Zhao et al. (2020) mengembangkan metode identifikasi antrian menggunakan kombinasi deteksi objek dan tracking berbasis *DeepSORT*. Hasil studi tersebut menunjukkan bahwa antrian dapat terdeteksi secara otomatis melalui estimasi jumlah orang dalam area tertentu. Pada aplikasi lain, Gao et al. (2021) mengimplementasikan *queue detection* berbasis *YOLO* dan berhasil memperkirakan waktu tunggu pelanggan melalui pengukuran panjang antrian dan kecepatan pergerakan individu.

Di sisi perangkat keras, penelitian penggunaan ESP32-CAM sebagai *platform* visi komputer semakin luas karena harganya murah dan memiliki modul kamera terintegrasi. Warden & Situnayake (2019) mencatat bahwa ESP32 merupakan kandidat ideal untuk aplikasi *TinyML*. Penelitian oleh S. Kumaresan et al. (2022) menunjukkan bahwa ESP32-CAM mampu menjalankan model *TFLite Micro* untuk deteksi sederhana seperti *face detection* dengan latensi rendah. Penelitian lain oleh Ahmad et al. (2021) mengintegrasikan ESP32-CAM untuk *surveillance* dan deteksi objek ringan, membuktikan bahwa perangkat ini dapat digunakan dalam sistem *edge-vision* dengan tingkat efisiensi tinggi.

Untuk memberikan gambaran komprehensif, berikut adalah tabel ringkasan penelitian terdahulu:

Tabel 2.1 Penelitian Terdahulu

No	Peneliti & Tahun	Topik	Metode / Perangkat	Kontribusi	Keterbatasan
1	Redmon & Farhadi (2018)	<i>People counting</i> berbasis kamera	YOLOv3	Deteksi <i>real-time</i> dengan akurasi tinggi	Membutuhkan GPU; tidak cocok untuk perangkat <i>edge</i>
2	Chen et al. (2021)	<i>People counting</i> publik	Kamera CCTV + <i>object detection</i>	Akurasi tinggi pada kondisi ramai	Bergantung pada server; latensi tinggi
3	David et al. (2020)	<i>TinyML</i> untuk deteksi objek	<i>TFLite quantization</i>	Model ringan & efisien	Tidak diuji pada kamera ESP32
4	Imteaj et al. (2021)	<i>Human detection</i> berbasis <i>TinyML</i>	<i>Edge ML</i> di <i>mikrokontroler</i>	Latensi rendah, tanpa <i>cloud</i>	Keterbatasan resolusi kamera
5	Zhao et al. (2019)	<i>Queue detection</i>	YOLO + DeepSORT	Deteksi antrean otomatis	Perlu kamera berkualitas tinggi
6	Gao et al. (2021)	Estimasi waktu antrean	<i>Object detection</i> + <i>tracking</i>	Prediksi waktu tunggu	Tidak cocok untuk perangkat <i>low-power</i>
7	Kumaresan et al. (2022)	<i>Vision on microcontroller</i>	ESP32-CAM + <i>TFLite</i> Micro	Deteksi sederhana pada ESP32	Tidak mendukung model besar
8	Ahmad et al. (2021)	<i>Object surveillance</i>	ESP32-CAM	Sistem monitoring ringan & murah	Performa rendah pada <i>low-light</i>

Secara keseluruhan, penelitian terdahulu menunjukkan bahwa sistem *people counting* dan *queue detection* dapat dicapai dengan akurasi tinggi menggunakan teknik visi komputer modern. Namun, sebagian besar penelitian masih mengandalkan perangkat komputasi berat atau server eksternal. Penelitian *TinyML* pada ESP32-CAM masih terbatas pada kasus deteksi sederhana, sehingga terdapat ruang untuk kontribusi baru dalam mengoptimalkan sistem *people counting* dan

deteksi antrean berbasis *TinyML* pada kamera *IoT* berdaya rendah untuk aplikasi ruang *indoor* seperti *coffeeshop*.

2.10. Analisis Gap

Analisis gap bertujuan mengidentifikasi kekurangan penelitian terdahulu serta ruang inovasi yang belum banyak dieksplorasi, sehingga mempertegas posisi dan kontribusi penelitian yang dilakukan. Berdasarkan ringkasan penelitian sebelumnya mengenai *people counting*, deteksi antrean, *TinyML*, dan penggunaan ESP32-CAM, dapat dilihat bahwa sebagian besar studi masih berfokus pada penggunaan kamera berkinerja tinggi atau komputasi server, sementara penelitian berbasis perangkat *low-power* untuk lingkungan *indoor* seperti *coffeeshop* masih sangat terbatas.

Penelitian *people counting* berbasis kamera umumnya menggunakan model deteksi objek berukuran besar seperti *YOLO*, *SSD*, atau *Faster R-CNN* yang membutuhkan perangkat GPU atau sistem server untuk melakukan inferensi. Hal ini menyebabkan implementasi menjadi kurang efisien dan tidak praktis untuk usaha kecil menengah (UKM) seperti *coffeeshop* yang membutuhkan solusi murah dan hemat energi. Selain itu, sebagian besar penelitian *queue detection* dilakukan pada lingkungan retail besar dan ruang publik, bukan pada ruang sempit seperti *café* dengan karakteristik visual dan aliran pelanggan yang berbeda.

Pada sisi *TinyML*, meskipun penelitian sebelumnya menunjukkan bahwa model kuantisasi dapat berjalan dengan baik pada mikrokontroler, studi aplikasi *TinyML* untuk visi komputer lebih banyak fokus pada tugas sederhana seperti klasifikasi citra atau pendeteksian objek tunggal. Implementasi *TinyML* untuk deteksi banyak objek secara *real-time*, termasuk manusia dalam kondisi *indoor*

dengan *occlusion*, masih merupakan tantangan yang belum banyak dibahas dalam literatur. Begitu pula penggunaan ESP32-CAM umumnya terbatas pada tugas-tugas dasar seperti *streaming* video atau deteksi wajah sederhana, belum banyak diterapkan untuk tugas kompleks seperti *people counting* dan *queue detection* yang membutuhkan konsistensi dan akurasi tinggi.

Dari sisi lingkungan aplikasi, sebagian besar penelitian sebelumnya menggunakan dataset publik atau lingkungan *outdoor* dan *semi-outdoor*. Hal ini menimbulkan domain gap ketika model diterapkan pada ruang *café* yang memiliki kondisi cahaya rendah, jarak kamera pendek, sudut pandang tetap, dan kepadatan pelanggan yang tidak terlalu tinggi namun variatif sepanjang hari. Minimnya dataset indoor spesifik *café* membuat banyak sistem gagal mempertahankan akurasi ketika diimplementasikan dalam kondisi nyata.

Berdasarkan kekurangan tersebut, terdapat ruang inovasi penelitian pada beberapa aspek berikut:

1. Implementasi sistem *people counting* berbasis *TinyML* pada perangkat *low-power* seperti ESP32-CAM, yang belum banyak dibahas dalam penelitian terdahulu.
2. Pengembangan model deteksi objek yang dioptimalkan untuk lingkungan *café* termasuk kondisi *low-light*, *occlusion*, dan ruang sempit.
3. Integrasi *people counting* dan *queue detection* secara bersamaan dalam satu sistem berbasis *edge computing*.
4. Penggunaan pendekatan *real-time* tanpa ketergantungan pada server, sehingga lebih hemat biaya dan meningkatkan privasi pengguna.

5. Penerapan pada konteks *coffeeshop*, yang secara spesifik memiliki pola aliran pelanggan berbeda dari retail besar maupun lingkungan publik.

Oleh karena itu, penelitian ini menawarkan kontribusi unik berupa pengembangan sistem *people counting* dan deteksi antrean berbasis *TinyML* yang berjalan langsung pada ESP32-CAM sebagai perangkat *IoT* hemat energi. Sistem ini diharapkan mampu memberikan pemantauan ruang secara *real-time* tanpa membutuhkan perangkat mahal, sekaligus menjadi solusi praktis untuk meningkatkan efisiensi manajemen ruang pada *coffeeshop*.

BAB III

METODOLOGI PENELITIAN

3.1. Jenis dan Pendekatan Penelitian

Penelitian ini termasuk dalam kategori penelitian rekayasa (*engineering research*) dengan pendekatan eksperimental, yang berfokus pada perancangan, pengembangan, dan pengujian suatu sistem berbasis teknologi. Penelitian rekayasa bertujuan menghasilkan solusi teknis terhadap permasalahan nyata melalui proses desain sistem, implementasi, serta evaluasi kinerja sistem yang dikembangkan. Menurut Creswell (2014), pendekatan eksperimental digunakan untuk menguji efektivitas suatu metode atau sistem melalui pengamatan dan pengukuran terhadap variabel tertentu dalam kondisi terkontrol.

Pendekatan yang digunakan dalam penelitian ini adalah pendekatan kuantitatif, karena kinerja sistem yang dikembangkan dievaluasi menggunakan parameter numerik dan metrik pengukuran yang terukur, seperti akurasi deteksi, kesalahan perhitungan jumlah orang, serta waktu proses inferensi. Pendekatan kuantitatif umum digunakan dalam penelitian berbasis sistem cerdas dan *machine learning* karena memungkinkan analisis objektif terhadap performa sistem (Sugiyono, 2019).

Secara teknis, penelitian ini juga menerapkan pendekatan eksperimen sistem, di mana sistem *people counting* dan deteksi antrean dikembangkan menggunakan kamera IoT ESP32-CAM dan model *Tiny Machine Learning*, kemudian diuji pada lingkungan nyata yang menyerupai kondisi operasional *coffeeshop*. Pendekatan ini sejalan dengan penelitian di bidang *Internet of Things* dan *edge computing* yang

menekankan pengujian sistem langsung pada perangkat target untuk memastikan keandalan dan efisiensi penggunaan sumber daya (Banbury et al., 2021).

Tahapan penelitian dilakukan secara sistematis mulai dari studi literatur, pengumpulan dan pengolahan data, perancangan model *machine learning*, optimasi model menggunakan pendekatan *TinyML*, hingga implementasi dan pengujian sistem. Fokus utama penelitian ini bukan hanya pada pengembangan model, tetapi juga pada integrasi model ke dalam perangkat *IoT* berdaya rendah untuk mendukung pemrosesan data secara lokal (*on-device inference*). Pendekatan ini sesuai dengan paradigma *edge computing* yang bertujuan mengurangi ketergantungan terhadap server dan meningkatkan efisiensi sistem (Shi et al., 2016).

Dengan demikian, jenis dan pendekatan penelitian yang digunakan diharapkan mampu menghasilkan sistem *people counting* dan deteksi antrian yang efektif, efisien, serta aplikatif untuk optimalisasi manajemen ruang pada *coffeeshop*.

3.2. Tahapan Penelitian

Tahapan penelitian disusun secara sistematis untuk memastikan proses perancangan dan pengembangan sistem *people counting* dan deteksi antrian berbasis *Tiny Machine Learning (TinyML)* berjalan terstruktur dan dapat direplikasi. Setiap tahapan dirancang saling terkait, dimulai dari identifikasi permasalahan hingga evaluasi kinerja sistem, sesuai dengan pendekatan penelitian rekayasa dan eksperimen sistem.

Tahap awal penelitian diawali dengan identifikasi masalah dan studi literatur. Pada tahap ini dilakukan pengamatan terhadap permasalahan manajemen ruang

pada *coffeeshop*, khususnya terkait keterbatasan informasi *real-time* mengenai jumlah pengunjung dan kondisi antrean. Studi literatur dilakukan untuk mengkaji konsep *Internet of Things*, visi komputer, *people counting*, deteksi antrean, serta perkembangan *TinyML* pada perangkat *edge*. Tahap ini bertujuan memperoleh landasan teoretis yang kuat dan menentukan pendekatan teknis yang sesuai dengan keterbatasan perangkat ESP32-CAM (Creswell, 2014; Banbury et al., 2021).

Tahap berikutnya adalah pengumpulan data, yang mencakup pemanfaatan dataset publik dan data kustom. Dataset publik digunakan untuk memberikan keragaman data manusia dalam berbagai pose dan kondisi, sedangkan data kustom dikumpulkan dari lingkungan *indoor* yang menyerupai kondisi *coffeeshop* untuk mengurangi *domain gap*. Proses pengumpulan data dilakukan dengan mempertimbangkan sudut pandang kamera, pencahayaan, serta kepadatan pengunjung agar data yang diperoleh representatif terhadap kondisi nyata.

Setelah data terkumpul, dilakukan pra-pemrosesan data (*preprocessing*). Tahapan ini meliputi penyesuaian ukuran citra, normalisasi nilai piksel, serta augmentasi data untuk meningkatkan keragaman dan ketahanan model terhadap variasi lingkungan. Dataset kemudian dibagi menjadi data latih, data validasi, dan data uji untuk mendukung proses pelatihan dan evaluasi model secara objektif (Goodfellow et al., 2021).

Tahap selanjutnya adalah perancangan dan pelatihan model *machine learning*. Pada tahap ini ditentukan arsitektur model yang sesuai untuk tugas deteksi manusia dengan mempertimbangkan keterbatasan memori dan komputasi perangkat target. Model dilatih menggunakan data latih yang telah diproses, sedangkan data validasi digunakan untuk memantau performa model selama proses

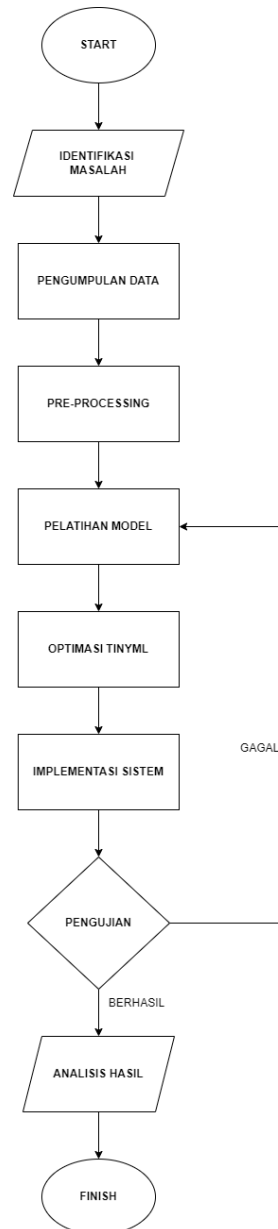
pelatihan. Tahap ini bertujuan menghasilkan model awal yang memiliki kemampuan deteksi manusia yang memadai sebelum dilakukan optimasi.

Model yang telah dilatih kemudian memasuki tahap optimasi model dan mekanisme implementasi *TinyML*. Tahap ini mencakup penerapan teknik optimasi seperti quantization serta konversi model ke format yang kompatibel dengan *TensorFlow Lite Micro*. Model hasil optimasi selanjutnya diintegrasikan ke dalam sistem ESP32-CAM untuk memungkinkan proses inferensi dilakukan secara langsung pada perangkat (*on-device inference*). Tahap ini dilakukan dengan fokus pada prosedur dan mekanisme implementasi, tanpa membahas hasil kinerja sistem.

Setelah sistem terintegrasi, dilakukan perancangan logika *people counting* dan deteksi antrean. Tahapan ini mencakup penentuan *region of interest* (ROI), mekanisme penghitungan jumlah orang, serta aturan logika untuk mendeteksi kondisi antrean berdasarkan jumlah dan posisi individu dalam area tertentu. Perancangan logika ini menjadi dasar bagi sistem dalam menghasilkan keluaran yang relevan untuk manajemen ruang.

Tahap akhir penelitian adalah pengujian dan evaluasi sistem. Pengujian dilakukan untuk menilai kemampuan sistem dalam mendeteksi manusia, menghitung jumlah pengunjung, serta mengidentifikasi kondisi antrean. Evaluasi kinerja dilakukan menggunakan metrik kuantitatif yang sesuai untuk menilai akurasi dan keandalan sistem. Hasil pengujian selanjutnya dianalisis untuk menarik kesimpulan mengenai efektivitas sistem yang dikembangkan dalam mendukung optimalisasi manajemen ruang pada *coffeeshop* (Shi et al., 2020).

Berikut adalah alur tahapan penelitian menggunakan Diagram *Flowchart* pada gambar 3.1 di bawah ini:



Gambar 3.1 Flowchart Tahapan Penelitian

3.3. Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan pada sebuah lingkungan *coffeeshop* yang digunakan sebagai objek studi untuk menguji sistem *people counting* dan deteksi antrian berbasis *Tiny Machine Learning (TinyML)*. Pemilihan lokasi penelitian didasarkan pada karakteristik ruang yang sesuai dengan tujuan penelitian, yaitu ruang *indoor* dengan kapasitas terbatas, pola pergerakan pengunjung yang dinamis,

serta adanya area antrean yang jelas pada titik pelayanan. Lingkungan tersebut dinilai representatif untuk menguji kinerja sistem dalam kondisi nyata yang sering ditemui pada operasional *coffeeshop*.

Secara umum, lokasi penelitian berada di The Gade Coffee and Gold yang beralamat di Jl. Babura, Kec. Medan Baru, Kota Medan. Area yang menjadi fokus penelitian meliputi Bar, Indoor, Pintu Masuk *Indoor*, karena area tersebut memiliki intensitas pergerakan pelanggan yang tinggi dan berpengaruh langsung terhadap manajemen ruang. Kamera IoT ESP32-CAM ditempatkan pada posisi sudut meja bartender, sebelah pintu masuk *indoor* untuk memperoleh sudut pandang yang optimal dalam mendeteksi keberadaan dan pergerakan pengunjung.

Waktu pelaksanaan penelitian dilakukan pada periode Desember 2025 – Februari 2026, yang mencakup tahap pengumpulan data, pengembangan sistem, hingga pengujian dan evaluasi. Pengambilan data dilakukan pada beberapa rentang waktu operasional, termasuk jam sepi dan jam ramai, dengan tujuan memperoleh variasi kondisi jumlah pengunjung dan pola antrean. Pendekatan ini dilakukan agar sistem yang dikembangkan dapat diuji pada kondisi yang berbeda dan mencerminkan situasi operasional sebenarnya.

Penentuan lokasi dan waktu penelitian ini diharapkan dapat mendukung proses evaluasi sistem secara objektif dan realistis, sehingga hasil penelitian dapat menggambarkan kinerja sistem *people counting* dan deteksi antrean berbasis *TinyML* dalam konteks penerapan nyata pada *coffeeshop*.

3.4. Perancangan Sistem

Perancangan sistem merupakan tahap penting dalam metodologi penelitian yang bertujuan untuk mendefinisikan struktur, komponen, serta alur kerja sistem

people counting dan deteksi antrean berbasis *Tiny Machine Learning (TinyML)*. Pada tahap ini, sistem dirancang secara menyeluruh dengan mempertimbangkan keterbatasan perangkat keras, kebutuhan pemrosesan *real-time*, serta tujuan utama penelitian yaitu mendukung optimalisasi manajemen ruang pada *coffexeshop*.

Secara umum, sistem yang dikembangkan mengadopsi arsitektur *Internet of Things (IoT)* berbasis *edge computing*, di mana proses utama pengolahan data dilakukan langsung pada perangkat kamera IoT ESP32-CAM. Pendekatan ini dipilih untuk mengurangi ketergantungan pada server eksternal, menekan latensi, serta menjaga privasi data pengunjung karena citra tidak perlu dikirim secara mentah ke *cloud*. ESP32-CAM berperan sebagai perangkat akuisisi data visual sekaligus perangkat inferensi model *TinyML*.

Alur kerja sistem dimulai dari proses pengambilan citra oleh kamera ESP32-CAM. Citra yang diperoleh kemudian diproses melalui tahapan *preprocessing* sederhana di perangkat, sebelum dijadikan input bagi model *TinyML* yang telah dioptimasi. Model tersebut melakukan inferensi untuk mendeteksi keberadaan manusia pada setiap *frame*. Hasil inferensi selanjutnya diproses oleh modul logika sistem untuk menghitung jumlah orang dan menentukan kondisi antrean berdasarkan *region of interest (ROI)* yang telah ditetapkan. Informasi hasil penghitungan kemudian dapat ditampilkan atau dikirimkan ke sistem *monitoring* sebagai data pendukung manajemen ruang.

Sebagai pelengkap dari arsitektur sistem yang telah dirancang, penelitian ini juga mengintegrasikan sistem monitoring berbasis web yang berfungsi sebagai antarmuka visual bagi pengguna. Sistem monitoring dirancang untuk menampilkan hasil inferensi sistem *people counting* dan deteksi antrean yang dihasilkan oleh

ESP32-CAM, yaitu informasi jumlah pengunjung dan status antrean, secara *real-time*. Sistem ini tidak melakukan pemrosesan *machine learning*, melainkan hanya berperan sebagai media visualisasi data hasil inferensi.

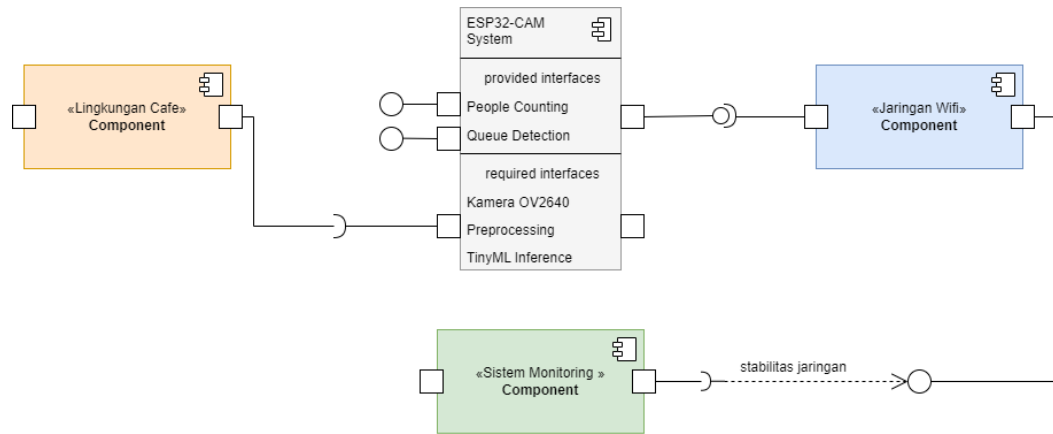
Sistem monitoring menerima data hasil pemrosesan dari ESP32-CAM melalui jaringan Wi-Fi dalam bentuk data numerik dan status logis, sehingga tidak melibatkan pengiriman citra mentah. Pendekatan ini sejalan dengan prinsip *edge computing* yang menempatkan beban komputasi utama pada perangkat edge, sekaligus menjaga efisiensi sistem dan perlindungan privasi pengunjung. Informasi yang ditampilkan pada sistem monitoring selanjutnya dapat dimanfaatkan oleh pengelola coffeeshop untuk memantau kondisi kapasitas ruang dan antrean secara lebih mudah dan terstruktur.

Sebagai pelengkap dari arsitektur sistem yang telah dirancang, penelitian ini juga mengintegrasikan sistem *monitoring* berbasis web yang berfungsi sebagai antarmuka visual bagi pengguna. Sistem *monitoring* dirancang untuk menampilkan hasil inferensi sistem *people counting* dan deteksi antrean yang dihasilkan oleh ESP32-CAM, yaitu informasi jumlah pengunjung dan status antrean, secara *real-time*. Sistem ini tidak melakukan pemrosesan *machine learning*, melainkan hanya berperan sebagai media visualisasi data hasil inferensi.

Sistem monitoring menerima data hasil pemrosesan dari ESP32-CAM melalui jaringan Wi-Fi dalam bentuk data numerik dan status logis, sehingga tidak melibatkan pengiriman citra mentah. Pendekatan ini sejalan dengan prinsip *edge computing* yang menempatkan beban komputasi utama pada perangkat edge, sekaligus menjaga efisiensi sistem dan perlindungan privasi pengunjung. Informasi yang ditampilkan pada sistem monitoring selanjutnya dapat dimanfaatkan oleh

pengelola coffeeshop untuk memantau kondisi kapasitas ruang dan antrian secara lebih mudah dan terstruktur.

Arsitektur sistem secara konseptual dapat digambarkan sebagai berikut:



Gambar 3.2 Diagram Arsitektur Sistem

Penjelasan diagram:

1. *Lingkungan Café*: Area nyata tempat pengunjung berada dan aktivitas terjadi.
2. *ESP32-CAM*: Perangkat utama yang melakukan pengambilan citra dan inferensi *TinyML* secara *on-device*.
3. *People Counting & Queue Detection*: Modul logika yang mengolah hasil deteksi menjadi informasi jumlah orang dan kondisi antrian.
4. *Jaringan Wi-Fi*: Media komunikasi data jika hasil perlu dikirim ke sistem *monitoring*.
5. *Sistem Monitoring*: Digunakan untuk menampilkan informasi secara visual, bersifat opsional dan tidak mempengaruhi proses inferensi inti.

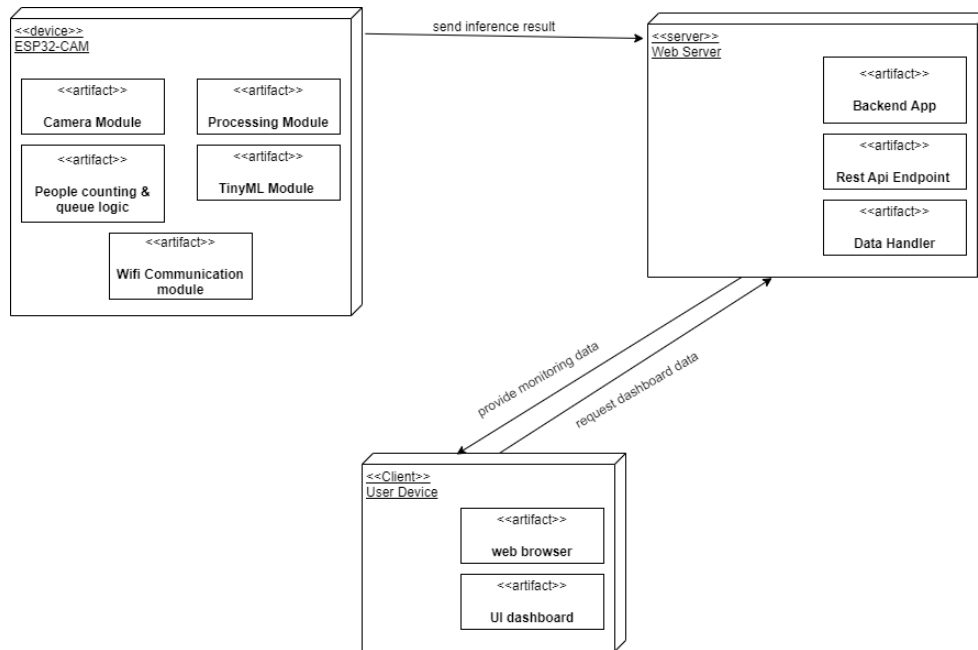
Diagram ini nantinya dapat Anda gambar ulang dalam bentuk diagram blok sistem menggunakan tools seperti draw.io, PowerPoint, atau Figma. Jenis diagram yang paling sesuai untuk menggambarkan perancangan sistem pada tahap ini adalah

UML Diagram Jenis *Component Diagram*. Alasan pemilihan *Component Diagram* adalah sebagai berikut:

1. Diagram ini berfokus pada komponen sistem dan hubungan antar komponen, bukan pada detail algoritma.
2. Cocok digunakan pada tahap metodologi untuk menjelaskan arsitektur sistem secara struktural.
3. Mampu merepresentasikan integrasi antara perangkat keras (ESP32-CAM), perangkat lunak (model *TinyML*), serta sistem komunikasi (Wi-Fi dan *dashboard*).

Selain *Component Diagram*, diagram ini juga memiliki karakteristik menyerupai Diagram Arsitektur Sistem, yang lazim digunakan dalam penelitian *IoT* dan sistem cerdas. Diagram ini tidak dimaksudkan untuk menjelaskan urutan eksekusi secara detail (seperti *Sequence Diagram*), melainkan memberikan gambaran menyeluruh tentang bagaimana sistem dibangun dan saling terhubung.

Untuk melengkapi gambaran perancangan sistem dari sisi lokasi eksekusi dan distribusi komponen, digunakan *Deployment Diagram* yang menunjukkan pemetaan perangkat, server, dan antarmuka pengguna dalam sistem monitoring.

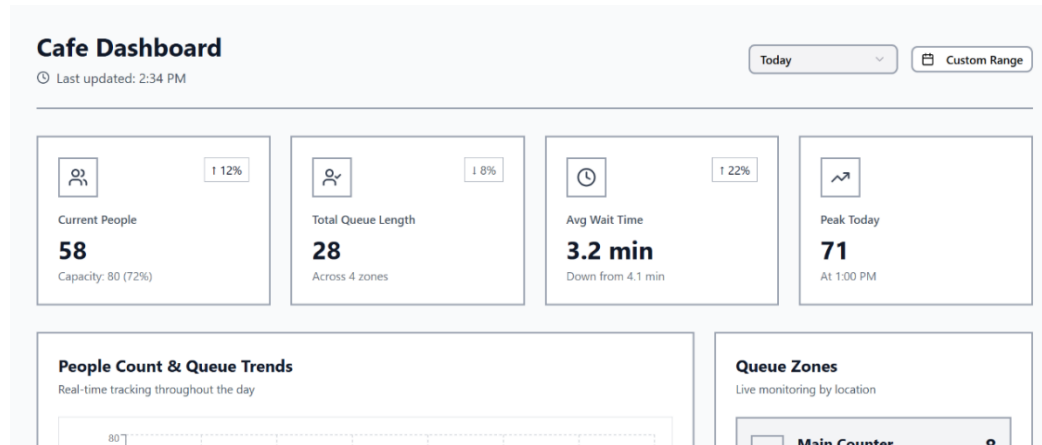


Gambar 3.3 Deployment Diagram Sistem

Deployment Diagram pada Gambar 3.3 menggambarkan pemetaan lokasi eksekusi komponen sistem people counting dan deteksi antrian berbasis Tiny Machine Learning (TinyML). Proses inferensi dan pengolahan citra dijalankan sepenuhnya pada perangkat ESP32-CAM sebagai *edge device*, sehingga tidak memerlukan pengiriman data citra mentah ke server. Hasil inferensi berupa jumlah pengunjung dan status antrian selanjutnya dikirimkan melalui jaringan Wi-Fi ke web server untuk ditampilkan pada sistem monitoring berbasis web. Pengguna mengakses sistem monitoring melalui peramban (*web browser*) pada perangkat klien untuk memantau kondisi ruang secara *real-time*. Deployment diagram ini menegaskan penerapan arsitektur *edge computing* yang menempatkan beban komputasi utama pada perangkat tepi, sekaligus memastikan efisiensi komunikasi, latensi rendah, dan perlindungan privasi data.

Sebagai bagian dari perancangan sistem monitoring, dirancang pula rancangan antarmuka berbasis web dalam bentuk wireframe dashboard. Wireframe

ini digunakan untuk menggambarkan tata letak dan komponen utama antarmuka sistem monitoring yang menampilkan hasil people counting dan deteksi antrian. Perancangan wireframe dilakukan untuk memberikan gambaran konseptual mengenai struktur tampilan sistem sebelum tahap implementasi, tanpa membahas aspek visual atau teknis secara detail.



Gambar 3.4 Wireframe Dashboard Sistem

Wireframe dashboard web pada Gambar 3.4 menunjukkan rancangan antarmuka sistem monitoring yang menyajikan informasi utama berupa jumlah pengunjung saat ini, panjang antrian, waktu tunggu rata-rata, serta kondisi puncak kunjungan. Informasi tersebut ditampilkan dalam bentuk panel ringkas agar mudah dipahami oleh pengguna. Selain itu, wireframe juga menampilkan area visualisasi tren jumlah pengunjung dan distribusi antrian berdasarkan zona, yang bertujuan membantu pengelola coffeeshop dalam memantau dinamika ruang secara real-time. Rancangan antarmuka ini difokuskan pada keterbacaan informasi dan kemudahan penggunaan sebagai dasar pendukung pengambilan keputusan manajerial.

Pada sistem ini, Firebase Realtime Database digunakan sebagai media penyimpanan data hasil inferensi dari ESP32-CAM serta data status yang ditampilkan pada dashboard monitoring. Oleh karena itu diperlukan perancangan

struktur database yang terorganisasi agar setiap data hasil deteksi dapat disimpan, diproses, dan ditampilkan dengan konsisten. Tabel berikut merupakan rancangan struktur database yang digunakan dalam penelitian ini.

Tabel 3.1 Rancangan Sistem Database

Nama Field	Tipe Data	Deskripsi
people_inside	Integer	Jumlah orang yang terdeteksi berada di dalam area ruangan pada waktu tertentu.
meja_terisi	Integer	Jumlah meja yang sedang digunakan (berdasarkan kondisi real-time dari area café).
meja_tersedia	Integer	Jumlah meja yang masih tersedia dan dapat ditempati pengunjung.
total_meja	Integer	Jumlah total meja yang didaftarkan di sistem.
occupancy_pct	Integer	Persentase tingkat keterisian area berdasarkan jumlah meja terisi.
queue_status	String	Status antrean berdasarkan logika antrian sistem, misalnya: “SEPI”, “SEDANG”, atau “RAMAI”.
uptime_sec	Integer	Total lama perangkat aktif sejak booting terakhir dalam satuan detik.
timestamp_ms	Integer	Waktu pengiriman data dalam format UNIX epoch millisecond.
device_id	String	Identitas perangkat ESP32-CAM yang mengirimkan data. (Opsional, tapi sangat disarankan).
frame_id	Integer	Nomor urutan frame sebagai penanda setiap cycle inferensi. (Opsional, mendukung debugging).
manual_override	Boolean	Digunakan bila ada intervensi manual dari operator (opsional).

Dengan perancangan sistem yang jelas dan terstruktur ini, penelitian memiliki fondasi metodologis yang kuat untuk tahap implementasi, pengujian, dan evaluasi sistem pada subbab selanjutnya.

3.5. Pengumpulan Data

Pengumpulan data merupakan tahap penting dalam penelitian ini karena kualitas data sangat memengaruhi kinerja model *people counting* dan deteksi

antrean yang dikembangkan. Data yang digunakan dalam penelitian ini terdiri atas dataset publik dan data kustom yang dikumpulkan dari lingkungan *indoor* yang merepresentasikan kondisi *coffeeshop*. Pendekatan ini bertujuan untuk memperoleh keragaman data sekaligus mengurangi *domain gap* antara data pelatihan dan kondisi implementasi nyata.

Dataset publik dimanfaatkan sebagai sumber data awal untuk melatih model agar mampu mengenali objek manusia dalam berbagai pose dan kondisi visual. Dataset yang digunakan meliputi dataset berskala besar yang umum digunakan dalam penelitian visi komputer, seperti COCO dan *CrowdHuman*, yang menyediakan anotasi manusia dalam bentuk *bounding box*. Penggunaan dataset publik bertujuan untuk meningkatkan kemampuan generalisasi model pada tahap awal pelatihan (Lin et al., 2014; Shao et al., 2018).

Selain dataset publik, penelitian ini juga menggunakan data kustom yang dikumpulkan langsung dari lingkungan penelitian. Pengambilan data kustom dilakukan di The Gade Café pada area *Indoor*, dengan menggunakan kamera ESP32-CAM atau perangkat kamera lain yang memiliki karakteristik serupa. Data dikumpulkan dalam bentuk citra atau potongan *frame video* dengan mempertimbangkan variasi kondisi operasional, seperti perbedaan tingkat kepadatan pengunjung, perubahan pencahayaan, serta variasi sudut pandang kamera. Pendekatan ini sejalan dengan penelitian sebelumnya yang menekankan pentingnya data kontekstual untuk meningkatkan akurasi sistem visi komputer pada lingkungan *indoor* (Zhang et al., 2021).

Proses pengumpulan data dilakukan pada beberapa rentang waktu operasional, termasuk pagi sampai siang dan sore sampai malam, untuk

memperoleh variasi jumlah pengunjung dan pola antrean. Data yang diperoleh kemudian diseleksi untuk memastikan kualitas visual yang memadai dan relevan dengan tujuan penelitian. *Frame* yang buram, terlalu gelap, atau tidak menampilkan aktivitas manusia secara jelas dieliminasi dari dataset.

Selanjutnya, data yang telah terkumpul melalui proses anotasi, yaitu pemberian label pada objek manusia dalam citra menggunakan *bounding box*. Proses anotasi dilakukan secara manual dengan bantuan perangkat lunak anotasi citra, sehingga posisi dan jumlah individu pada setiap *frame* dapat digunakan sebagai *ground truth* dalam pelatihan dan evaluasi model. Anotasi yang konsisten dan akurat sangat penting untuk memastikan model dapat mempelajari representasi visual manusia dengan baik (Goodfellow et al., 2016).

Dengan mengombinasikan dataset publik dan data kustom *indoor*, tahap pengumpulan data diharapkan dapat menghasilkan dataset yang representatif terhadap kondisi nyata *coffeeshop*. Dataset ini selanjutnya digunakan pada tahap pra-pemrosesan dan pelatihan model untuk mendukung pengembangan sistem *people counting* dan deteksi antrean berbasis *TinyML*.

3.6. Pra-pemrosesan Data

Pra-pemrosesan data merupakan tahap yang bertujuan menyiapkan data mentah agar sesuai dengan kebutuhan pelatihan model *machine learning* dan dapat meningkatkan kinerja serta stabilitas model. Pada penelitian ini, pra-pemrosesan dilakukan terhadap data citra yang diperoleh dari dataset publik dan data kustom *indoor coffeeshop*. Tahapan pra-pemrosesan dirancang untuk menyesuaikan karakteristik data dengan keterbatasan perangkat ESP32-CAM serta kebutuhan model *TinyML*.

Tahap awal pra-pemrosesan adalah penyesuaian ukuran citra (*image resizing*). Seluruh citra diubah ke resolusi tertentu yang lebih kecil dan konsisten agar sesuai dengan *input* model serta mengurangi beban komputasi. Penyesuaian ukuran citra merupakan praktik umum dalam visi komputer karena ukuran input yang seragam memudahkan proses pelatihan dan inferensi, terutama pada perangkat dengan sumber daya terbatas (Goodfellow et al., 2021).

Tahap berikutnya adalah normalisasi nilai piksel, di mana nilai intensitas piksel citra disesuaikan ke rentang tertentu, seperti $[0,1]$ atau $[-1,1]$. Normalisasi bertujuan untuk mempercepat proses konvergensi selama pelatihan model dan mengurangi dominasi nilai piksel tertentu yang dapat mempengaruhi proses pembelajaran. Proses ini penting untuk memastikan model dapat mempelajari pola visual secara lebih stabil dan konsisten.

Selanjutnya dilakukan augmentasi data (*data augmentation*) untuk meningkatkan keragaman data latih. Augmentasi dilakukan dengan menerapkan transformasi ringan pada citra, seperti perubahan tingkat kecerahan (*brightness*), kontras, rotasi kecil, dan *horizontal flipping*. Teknik ini bertujuan untuk meningkatkan kemampuan generalisasi model terhadap variasi kondisi lingkungan, khususnya pencahayaan *indoor* yang sering berubah-ubah pada *coffeeshop*. Penelitian sebelumnya menunjukkan bahwa augmentasi data dapat mengurangi risiko *overfitting* dan meningkatkan ketahanan model terhadap kondisi visual yang tidak ideal (Shorten & Khoshgoftaar, 2020).

Setelah proses augmentasi, data kemudian melalui tahap seleksi dan penyeimbangan dataset. Pada tahap ini, distribusi jumlah citra dengan variasi kepadatan manusia diperhatikan agar dataset tidak didominasi oleh kondisi tertentu,

seperti kondisi sepi atau kondisi ramai saja. Penyeimbangan dataset dilakukan untuk memastikan model dapat belajar secara proporsional dari berbagai skenario yang mungkin terjadi dalam operasional *coffeeshop*.

Tahap akhir pra-pemrosesan adalah pembagian dataset ke dalam tiga bagian utama, yaitu data latih (*training set*), data validasi (*validation set*), dan data uji (*testing set*). Data latih digunakan untuk melatih model, data validasi untuk memantau performa model selama pelatihan, dan data uji untuk mengevaluasi kinerja model secara objektif. Pembagian dataset ini merupakan prosedur standar dalam pengembangan model *machine learning* untuk memastikan hasil evaluasi tidak bias (Bishop, 2021).

Dengan melalui tahapan pra-pemrosesan tersebut, data yang digunakan dalam penelitian ini diharapkan memiliki kualitas yang baik, representatif, dan sesuai untuk mendukung pelatihan serta implementasi model *TinyML* pada sistem *people counting* dan deteksi antrean berbasis ESP32-CAM.

3.7. Perancangan dan Pelatihan Model

Perancangan dan pelatihan model merupakan tahap metodologis yang bertujuan menghasilkan model *machine learning* yang mampu mendeteksi keberadaan manusia secara andal, sekaligus sesuai dengan keterbatasan komputasi perangkat ESP32-CAM. Pada tahap ini ditentukan arsitektur model, skema pelatihan, serta parameter yang digunakan, tanpa menyajikan hasil performa numerik yang akan dibahas pada bab selanjutnya.

Pemilihan arsitektur model dilakukan dengan mempertimbangkan kebutuhan deteksi manusia pada citra beresolusi rendah hingga menengah, serta efisiensi ukuran dan kompleksitas komputasi agar kompatibel dengan pendekatan *TinyML*.

Oleh karena itu, arsitektur *Convolutional Neural Network* (CNN) ringan dipilih sebagai dasar, dengan karakteristik jumlah parameter yang relatif kecil dan kemampuan ekstraksi fitur spasial yang baik. Arsitektur ringan seperti MobileNet atau CNN sederhana umum digunakan pada sistem edge karena menawarkan keseimbangan antara akurasi dan efisiensi komputasi (Howard et al., 2017).

Pada tahap perancangan, ditentukan pula spesifikasi *input* model, meliputi ukuran citra masukan dan jumlah kanal warna. Penyesuaian ini penting untuk memastikan kesesuaian antara data hasil pra-pemrosesan dan kebutuhan model. Selain itu, lapisan-lapisan utama CNN seperti *convolution layer*, *activation function*, dan *pooling layer* dirancang secara minimalis agar model tetap ringan namun mampu mengekstraksi fitur penting dari citra manusia.

Proses pelatihan model dilakukan menggunakan dataset yang telah melalui tahap pra-pemrosesan. Data latih digunakan untuk memperbarui bobot model melalui proses optimasi, sementara data validasi digunakan untuk memantau kinerja model selama pelatihan dan mencegah terjadinya *overfitting*. Metode optimasi yang digunakan mengikuti praktik umum dalam pelatihan model *deep learning*, dengan pemilihan *loss function* yang sesuai untuk tugas deteksi objek dan *optimizer* yang stabil untuk konvergensi model (Goodfellow et al., 2022).

Pelatihan model dilakukan pada lingkungan komputasi dengan sumber daya memadai (misalnya komputer atau *cloud environment*), sehingga proses pelatihan tidak dibatasi oleh keterbatasan perangkat target. Model yang dihasilkan pada tahap ini merupakan model dasar (*baseline model*) yang selanjutnya akan dioptimasi pada tahap berikutnya agar dapat diimplementasikan menggunakan pendekatan *TinyML*.

Untuk menjaga konsistensi metodologi, parameter pelatihan ditentukan sebelum proses pelatihan dimulai. Parameter tersebut meliputi jumlah *epoch*, *learning rate*, ukuran *batch*, serta jenis *optimizer*. Penentuan parameter model dilakukan dengan mengacu pada hasil penelitian terkini serta praktik umum dalam pengembangan convolutional neural networks (CNN) ringan untuk aplikasi visi komputer pada perangkat dengan keterbatasan sumber daya, sehingga konfigurasi parameter yang dipilih mampu menyeimbangkan kinerja model dan efisiensi komputasi (Howard et al., 2019; Banbury et al., 2021).

Dengan tahapan perancangan dan pelatihan model ini, diperoleh model dasar yang siap untuk memasuki tahap optimasi dan mekanisme implementasi *TinyML* pada subbab berikutnya. Tahap selanjutnya akan berfokus pada penyesuaian model agar dapat dijalankan secara efisien pada perangkat ESP32-CAM.

3.8. Optimasi Model dan Mekanisme Implementasi TinyML

Tahap optimasi model dan mekanisme implementasi *TinyML* bertujuan memastikan model *machine learning* yang telah dilatih dapat dijalankan secara efisien pada perangkat dengan sumber daya terbatas, khususnya ESP32-CAM. Pada tahap ini, fokus utama adalah pada prosedur teknis untuk menurunkan kompleksitas model, menyesuaikan format model dengan *runtime TinyML*, serta mengintegrasikannya ke dalam sistem *IoT* berbasis *edge computing*, tanpa membahas hasil kinerja atau evaluasi numerik.

Optimasi model dilakukan melalui penerapan teknik *quantization*, yaitu proses konversi representasi bobot dan aktivasi model dari format presisi tinggi (misalnya *floating-point*) ke format numerik yang lebih ringkas. *Quantization* dipilih karena mampu mengurangi ukuran model dan kebutuhan komputasi,

sehingga model lebih sesuai untuk dijalankan pada *mikrokontroler*. Teknik ini umum digunakan pada sistem *TinyML* karena memberikan efisiensi memori dan kecepatan inferensi yang lebih baik dengan penurunan akurasi yang relatif kecil (Jacob et al., 2018; Banbury et al., 2021).

Setelah proses optimasi, model dikonversi ke dalam format *TensorFlow Lite* (*TFLite*). Format ini dirancang khusus untuk mendukung inferensi pada perangkat *edge*. Selanjutnya, model *TFLite* diadaptasi menggunakan *TensorFlow Lite for Microcontrollers* (*TFLite Micro*), yang memungkinkan model dijalankan tanpa sistem operasi dan dengan alokasi memori statis. Tahapan ini penting untuk memastikan model dapat diintegrasikan langsung ke dalam *firmware* ESP32-CAM.

Mekanisme implementasi *TinyML* pada ESP32-CAM dilakukan dengan mengintegrasikan model *TFLite Micro* ke dalam lingkungan pengembangan mikrokontroler. Model yang telah dikonversi diubah ke dalam bentuk *array* statis agar dapat disimpan di memori program. Selanjutnya, *interpreter* *TFLite Micro* diinisialisasi dengan menyediakan *tensor arena*, yaitu area memori yang digunakan untuk menyimpan *input*, *output*, dan tensor perantara selama proses inferensi. Penentuan ukuran *tensor arena* dilakukan secara hati-hati agar sesuai dengan kapasitas memori perangkat.

Dalam proses inferensi, citra yang diambil oleh kamera ESP32-CAM terlebih dahulu melalui tahapan pra-pemrosesan sederhana sebelum dimasukkan sebagai input ke model *TinyML*. Model kemudian menghasilkan keluaran berupa informasi deteksi manusia yang selanjutnya diproses oleh modul logika sistem untuk mendukung fungsi *people counting* dan deteksi antrean. Seluruh proses inferensi dilakukan secara *on-device*, sehingga sistem tidak bergantung pada server eksternal

dan mampu beroperasi dengan latensi rendah serta menjaga privasi data visual (Warden & Situnayake, 2019).

Dengan mekanisme optimasi dan implementasi tersebut, sistem *TinyML* dapat dijalankan secara stabil pada perangkat ESP32-CAM. Tahap ini menjadi penghubung antara proses pelatihan model dan penerapan sistem secara nyata, serta memastikan bahwa model yang dikembangkan dapat berfungsi sesuai dengan kebutuhan penelitian dalam lingkungan operasional *coffeeshop*.

3.9 Perancangan Sistem People Counting dan Deteksi Antrean

Perancangan sistem *people counting* dan deteksi antrean bertujuan untuk mendefinisikan logika operasional yang mengolah keluaran inferensi *TinyML* menjadi informasi bermakna bagi manajemen ruang *coffeeshop*. Pada tahap ini, fokus perancangan diarahkan pada penentuan alur pemrosesan data visual, mekanisme penghitungan jumlah pengunjung, serta aturan deteksi kondisi antrean berbasis *region of interest* (ROI). Seluruh proses dirancang agar berjalan secara *on-device* pada ESP32-CAM dengan latensi rendah dan konsumsi sumber daya minimal.

Secara umum, sistem menerima masukan berupa citra dari kamera ESP32-CAM yang kemudian diproses oleh model *TinyML* untuk menghasilkan keluaran deteksi manusia. Keluaran ini selanjutnya diolah oleh modul logika sistem untuk menentukan jumlah orang yang berada dalam area tertentu serta status antrean. Pendekatan ini memungkinkan sistem menghasilkan informasi *real-time* tanpa ketergantungan pada pemrosesan eksternal.

ROI ditetapkan sebagai area spesifik pada bidang pandang kamera yang merepresentasikan lokasi antrean, seperti area di depan kasir atau titik pelayanan.

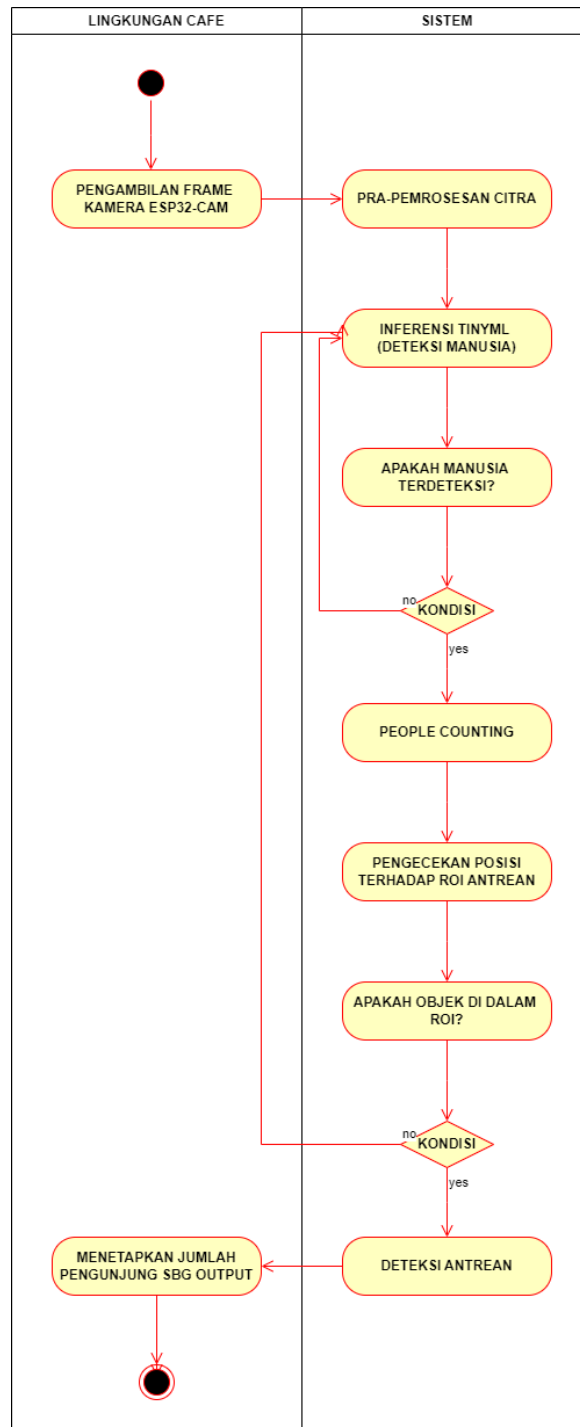
Penentuan ROI dilakukan pada tahap perancangan sistem dan bersifat statis selama pengujian, dengan tujuan membatasi area analisis agar sistem lebih fokus dan efisien. Dengan adanya ROI, hanya objek manusia yang berada di dalam area tersebut yang diperhitungkan dalam proses deteksi antrean, sehingga mengurangi kesalahan akibat pergerakan pengunjung di area lain.

Logika *people counting* dirancang untuk menghitung jumlah individu yang terdeteksi dalam satu frame citra. Sistem memanfaatkan keluaran model *TinyML* berupa informasi keberadaan manusia pada citra untuk menentukan jumlah pengunjung pada suatu waktu. Penghitungan dilakukan dengan menghitung jumlah objek manusia yang terdeteksi dan memenuhi kriteria tertentu, seperti tingkat kepercayaan (*confidence threshold*) dan posisi objek pada bidang pandang kamera. Mekanisme ini menghasilkan nilai jumlah pengunjung yang diperbarui secara berkala sesuai dengan frame yang diproses.

Deteksi antrean dirancang sebagai perluasan dari proses *people counting* dengan memanfaatkan ROI. Sistem mengevaluasi jumlah individu yang berada di dalam ROI antrean dan membandingkannya dengan ambang batas tertentu yang telah ditetapkan. Jika jumlah individu dalam ROI melebihi ambang batas tersebut, sistem mengklasifikasikan kondisi sebagai “terdapat antrean”, sedangkan jika berada di bawah ambang batas, sistem mengklasifikasikan kondisi sebagai “tidak terdapat antrean”. Pendekatan berbasis aturan (*rule-based*) ini dipilih karena sederhana, efisien, dan sesuai untuk implementasi pada perangkat dengan keterbatasan sumber daya.

Alur kerja sistem *people counting* dan deteksi antrean direpresentasikan menggunakan UML *Activity Diagram*, karena diagram ini mampu menggambarkan

urutan aktivitas dan keputusan logis secara jelas. Rangkaian alur diagram yang digunakan dalam perancangan sistem adalah sebagai berikut:



Gambar 3.5 Activity Diagram Perancangan Sistem

Alur ini dirancang berjalan secara berulang untuk setiap *frame* yang diproses, sehingga sistem mampu memberikan pembaruan informasi secara *real-time*.

Keluaran dari sistem *people counting* dan deteksi antrian berupa data numerik jumlah pengunjung serta status antrian yang bersifat kategorikal. Informasi ini selanjutnya dapat ditampilkan atau dikirimkan ke sistem *monitoring* sebagai bagian dari sistem *IoT* untuk mendukung pengambilan keputusan dalam manajemen ruang *coffeeshop*.

Dengan perancangan logika dan alur sistem tersebut, sistem diharapkan mampu memberikan informasi yang relevan, akurat, dan efisien mengenai kondisi ruang dan antrian, sekaligus mempertahankan kesederhanaan implementasi yang sesuai dengan keterbatasan perangkat ESP32-CAM. Hasil inferensi sistem selanjutnya dikirimkan ke sistem monitoring berbasis web untuk ditampilkan dalam bentuk dashboard

3.10. Metode Pengujian Sistem

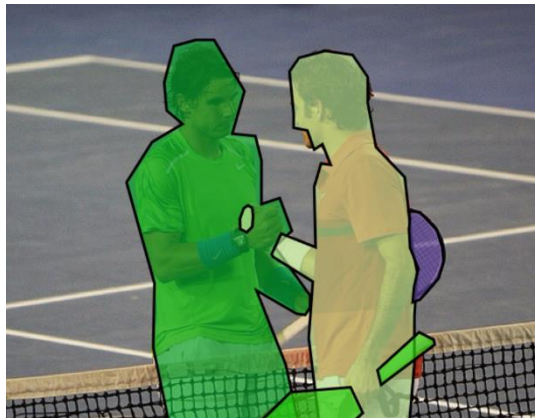
Metode pengujian sistem bertujuan untuk memastikan bahwa sistem *people counting* dan deteksi antrian berbasis *Tiny Machine Learning* (TinyML) pada ESP32-CAM berfungsi sesuai dengan tujuan perancangan. Pengujian dilakukan secara terstruktur untuk menilai keandalan sistem dalam mendeteksi manusia, menghitung jumlah pengunjung, serta mengidentifikasi kondisi antrian pada lingkungan indoor *coffeeshop*. Seluruh pengujian dirancang untuk mengevaluasi aspek fungsional dan kinerja sistem tanpa membahas hasil kuantitatif yang akan disajikan pada bab selanjutnya.

Data yang digunakan dalam pengujian sistem berupa citra dan cuplikan video manusia yang diperoleh dari dataset publik dan sumber data terbuka, yang merepresentasikan kondisi lingkungan indoor dengan karakteristik serupa *coffeeshop*. Data ini digunakan sebagai objek uji untuk mensimulasikan proses

inferensi sistem pada ESP32-CAM. Adapun data dan gambar yang digunakan meliputi:

1. Dataset *COCO* (*Common Objects in Context*)

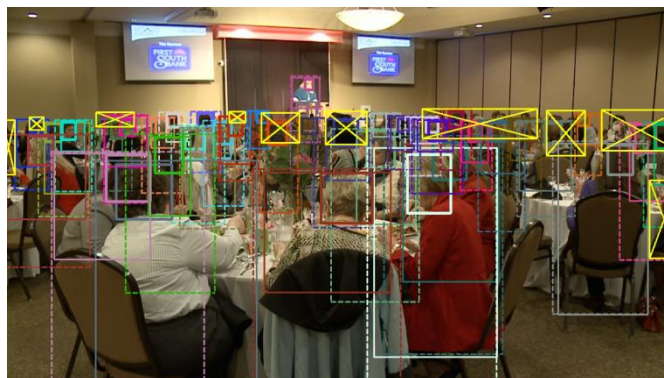
Dataset COCO digunakan sebagai sumber data utama untuk pengujian deteksi manusia (*person detection*). Data yang digunakan difokuskan pada kelas *person* dengan latar belakang indoor.



Gambar 3.6 Contoh citra manusia dari dataset *COCO*

2. Dataset *CrowdHuman* (*subset indoor*)

Dataset CrowdHuman digunakan untuk menguji kemampuan sistem dalam mendeteksi manusia pada kondisi kepadatan rendah hingga sedang. Data ini membantu mengevaluasi performa *people counting* ketika terdapat lebih dari satu individu dalam satu frame.



Gambar 3.7 Contoh citra uji dari dataset *CrowdHuman* dengan variasi jumlah individu.

3. Cuplikan video *indoor* publik (*open-source*)

Data berupa cuplikan video *indoor* yang diperoleh dari sumber terbuka digunakan untuk pengujian *continuous inference*. Video diekstraksi menjadi rangkaian frame dan diproses secara berurutan untuk mensimulasikan kondisi *real-time*.



Gambar 3.8 Contoh frame hasil ekstraksi video indoor yang digunakan dalam pengujian sistem.

Pengujian sistem diawali dengan pengujian fungsional, yaitu pengujian untuk memastikan setiap komponen sistem bekerja sesuai dengan alur yang telah dirancang. Pada tahap ini, dilakukan verifikasi terhadap proses pengambilan citra kamera, inferensi *TinyML* menggunakan data citra uji, logika *people counting*, serta mekanisme deteksi antrean berbasis *Region of Interest (ROI)*. Pengujian fungsional bertujuan memastikan bahwa sistem mampu menghasilkan keluaran berupa jumlah pengunjung dan status antrean pada setiap siklus pemrosesan.

Selanjutnya dilakukan pengujian akurasi deteksi manusia menggunakan citra uji yang telah dianotasi secara manual sebagai *ground truth*. Data citra dari dataset publik digunakan untuk membandingkan hasil deteksi sistem dengan kondisi aktual pada gambar. Tahap ini bertujuan memastikan bahwa model *TinyML* mampu

mengenali objek manusia secara konsisten sebelum digunakan dalam proses *people counting* dan deteksi antrean.

Pengujian *people counting* dilakukan menggunakan citra dan frame video yang mengandung jumlah individu yang bervariasi. Pengujian ini difokuskan pada evaluasi ketepatan sistem dalam menghitung jumlah orang yang terdeteksi pada setiap frame dengan membandingkannya terhadap *ground truth* berdasarkan anotasi manual.

Pengujian deteksi antrean dilakukan dengan menggunakan data citra dan video yang telah ditentukan area ROI antreannya. Sistem diuji untuk mengklasifikasikan kondisi antrean berdasarkan jumlah individu yang berada di dalam ROI, meliputi kondisi tidak ada antrean, antrean pendek, dan antrean padat.

Pengujian kinerja sistem mencakup pengujian waktu proses (*latency*) dengan menggunakan data citra dan video uji yang sama. Waktu pemrosesan diukur mulai dari pengambilan citra hingga sistem menghasilkan keluaran. Pengujian ini bertujuan memastikan sistem mampu beroperasi mendekati *real-time* pada perangkat ESP32-CAM.

Selain itu, dilakukan pengujian penggunaan sumber daya perangkat dengan mengamati konsumsi memori dan stabilitas sistem selama inferensi berlangsung menggunakan data uji secara berulang. Pengujian ini bertujuan memastikan implementasi *TinyML* dapat berjalan stabil pada perangkat dengan keterbatasan sumber daya.

Sebagai pelengkap, dilakukan pengujian pada variasi kondisi lingkungan menggunakan citra dan video dengan perbedaan pencahayaan dan tingkat

kepadatan pengunjung. Pengujian ini bertujuan mengevaluasi ketahanan sistem terhadap perubahan kondisi visual yang umum terjadi pada *coffeeshop*.

Dengan menerapkan metode pengujian yang terstruktur dan menggunakan data citra serta video uji yang representatif, sistem *people counting* dan deteksi antrian yang dikembangkan dapat dievaluasi secara objektif dan komprehensif. Hasil dari seluruh pengujian ini selanjutnya akan dianalisis dan dibahas pada BAB IV untuk menilai efektivitas dan kinerja sistem secara keseluruhan.

3.11. Metode Evaluasi dan Analisis Data

Metode evaluasi dan analisis data digunakan untuk menilai kinerja sistem *people counting* dan deteksi antrian berbasis *Tiny Machine Learning (TinyML)* yang telah diuji pada tahap sebelumnya. Evaluasi dilakukan secara kuantitatif dengan menggunakan metrik yang umum digunakan dalam penelitian visi komputer dan sistem cerdas, sehingga hasil pengujian dapat dianalisis secara objektif dan terukur. Tahap ini berfungsi sebagai jembatan antara proses pengujian sistem dan pembahasan hasil yang akan disajikan pada BAB IV.

Evaluasi kinerja sistem diawali dengan evaluasi deteksi manusia, yang bertujuan menilai kemampuan model *TinyML* dalam mengenali objek manusia pada citra. Evaluasi dilakukan dengan membandingkan hasil deteksi sistem terhadap *ground truth* yang diperoleh dari anotasi manual. Metrik evaluasi yang digunakan meliputi *precision* dan *recall*, yang masing-masing mengukur tingkat ketepatan deteksi dan kemampuan sistem dalam mendeteksi seluruh objek manusia yang ada. Penggunaan metrik ini umum diterapkan dalam penelitian deteksi objek untuk menilai keseimbangan antara kesalahan deteksi dan kelengkapan deteksi (Powers, 2020).

Selanjutnya dilakukan evaluasi *people counting*, yang difokuskan pada tingkat kesalahan sistem dalam menghitung jumlah pengunjung. Evaluasi ini dilakukan dengan membandingkan jumlah pengunjung yang dihasilkan sistem dengan jumlah aktual berdasarkan *ground truth*. Untuk mengukur tingkat kesalahan perhitungan, digunakan metrik *Mean Absolute Error (MAE)*, karena metrik ini mampu menggambarkan selisih rata-rata antara nilai prediksi dan nilai aktual secara intuitif. MAE banyak digunakan dalam penelitian *counting* karena mudah diinterpretasikan dan tidak terlalu sensitif terhadap nilai ekstrem (Chaudhry et al., 2020).

Evaluasi berikutnya adalah evaluasi deteksi antrean, yang bertujuan menilai ketepatan sistem dalam mengklasifikasikan kondisi antrean. Evaluasi dilakukan dengan membandingkan status antrean yang dihasilkan sistem dengan kondisi antrean aktual berdasarkan pengamatan manual. Analisis dilakukan dengan menghitung tingkat kesesuaian antara keluaran sistem dan *ground truth* untuk memastikan bahwa aturan logika deteksi antrean dapat bekerja secara konsisten.

Selain evaluasi akurasi, dilakukan pula analisis kinerja sistem yang mencakup waktu proses dan stabilitas sistem. Analisis waktu proses bertujuan menilai kemampuan sistem dalam melakukan inferensi secara efisien pada perangkat ESP32-CAM. Meskipun nilai numerik waktu proses tidak dibahas pada bab ini, metode pengukurannya dijelaskan sebagai dasar evaluasi performa sistem secara keseluruhan. Pendekatan ini sejalan dengan penelitian *edge computing* yang menekankan pentingnya efisiensi pemrosesan pada perangkat berdaya rendah (Shi et al., 2020).

Hasil dari seluruh metrik evaluasi kemudian dianalisis secara deskriptif untuk mengidentifikasi kekuatan dan keterbatasan sistem. Analisis ini mencakup interpretasi terhadap pola kesalahan, pengaruh kondisi lingkungan, serta konsistensi kinerja sistem pada berbagai skenario pengujian. Pendekatan analisis deskriptif kuantitatif digunakan untuk memberikan gambaran menyeluruh mengenai performa sistem tanpa menarik kesimpulan akhir, yang akan disajikan pada bab selanjutnya.

Dengan metode evaluasi dan analisis data tersebut, penelitian ini diharapkan mampu menilai kinerja sistem *people counting* dan deteksi antrean secara objektif, terukur, dan relevan dengan tujuan penelitian, serta menjadi dasar yang kuat untuk pembahasan hasil dan penarikan kesimpulan.

3.12. Alat dan Bahan Penelitian

Alat dan bahan penelitian digunakan untuk mendukung proses perancangan, pengembangan, pengujian, serta evaluasi sistem penghitungan orang dan deteksi antrean berbasis *Tiny Machine Learning (TinyML)*. Pemilihan alat dan bahan disesuaikan dengan kebutuhan penelitian serta keterbatasan sumber daya perangkat yang menjadi objek penelitian, khususnya ESP32-CAM sebagai perangkat sistem utama.

Alat penelitian yang digunakan meliputi perangkat keras dan perangkat lunak yang mendukung proses pengolahan data, model pelatihan, serta sistem implementasi *TinyML*. Perangkat keras digunakan sebagai media pengambilan data dan eksekusi sistem, sedangkan perangkat lunak digunakan untuk pengembangan model, pemrograman perangkat, dan analisis data.

Perangkat keras yang digunakan dalam penelitian ini antara kamera lain *IoT* ESP32-CAM sebagai perangkat utama akuisisi data visual dan inferensi *TinyML*, serta komputer atau laptop yang digunakan untuk proses pelatihan model dan pemrograman perangkat. Selain itu, jaringan Wi-Fi digunakan sebagai media komunikasi data untuk mendukung sistem *monitoring*.

Perangkat lunak yang digunakan meliputi lingkungan pengembangan pembelajaran mesin untuk pelatihan dan konversi model, serta perangkat lunak pemrograman *mikrokontroler* untuk mengunggah *firmware* ke ESP32-CAM. Perangkat lunak pendukung juga digunakan untuk anotasi data dan analisis hasil pengujian.

Bahan penelitian yang digunakan berupa data dan sumber pendukung yang diperlukan untuk pelatihan dan sistem pemancar. Data utama dalam penelitian ini adalah dataset citra manusia yang diperoleh dari dataset publik dan data kustom *indoor coffeeshop*. Dataset publik digunakan untuk melatih kemampuan dasar model dalam mendeteksi manusia, sedangkan data kustom digunakan untuk menyesuaikan model dengan kondisi lingkungan penelitian.

Selain data citra, bahan penelitian juga mencakup parameter dan konfigurasi sistem yang digunakan dalam proses pelatihan, optimasi, dan implementasi model *TinyML*. Bahan-bahan ini berperan penting dalam memastikan sistem dapat diuji dan dievaluasi secara konsisten. Untuk memperjelas komponen yang digunakan, ringkasan alat dan bahan penelitian disajikan pada Tabel 3.6.

Tabel 3.2 Alat dan Bahan Penelitian

Kategori	Nama	Fungsi
Perangkat keras	ESP32-CAM	Akuisisi citra dan inferensi <i>TinyML</i>

Perangkat keras	Komputer/Laptop	Pelatihan model dan pemrograman
Perangkat lunak	TensorFlow / TFLite	Pelatihan dan konversi model
Perangkat lunak	Arduino IDE / Platform sejenis	Implementasi firmware ESP32-CAM
Bahan data	Dataset publik & data kustom	Pelatihan dan evaluasi model
Jaringan	Wi-Fi	Komunikasi data <i>monitoring</i>

Tabel ini memberikan gambaran umum mengenai alat dan bahan yang digunakan dalam penelitian, tanpa mengungkapkan spesifikasi teknis secara berlebihan, sehingga tetap sesuai dengan ruang lingkup metodologi penelitian.

3.13. Etika dan Keamanan Data

Aspek etika dan keamanan data merupakan bagian penting dalam penelitian yang melibatkan pengambilan data visual, khususnya pada lingkungan publik seperti *coffeeshop*. Pada penelitian ini, penerapan sistem penghitungan orang dan deteksi antrean berbasis kamera *IoT* dirancang dengan memperhatikan pr

Dalam aspek etika penelitian, sistem dirancang untuk tidak melakukan proses pengenalan wajah (*face recognition*) atau identifikasi pribadi lainnya. Data citra yang digunakan hanya dimanfaatkan sebagai masukan untuk mendeteksi keberadaan *privacy by design*, di mana perlindungan privasi telah dipertimbangkan sejak tahap perancangan sistem.

Terkait keamanan data, penelitian ini menerapkan prinsip pemrosesan data secara pada perangkat melalui pendekatan *TinyML*. Seluruh proses inferensi dilakukan langsung pada perangkat ESP32-CAM, sehingga citra mentah tidak perlu dikirimkan ke server eksternal atau sistem *cloud*.

Apabila hasil keluaran sistem dikirimkan ke sistem pemantauan melalui jaringan *IoT*, data yang dikirimkan dibatasi pada informasi non-pribadi, seperti

jumlah pengunjung dan status antrean. Informasi tersebut tidak mengandung data sensitif dan tidak dapat digunakan secara langsung

Selain aspek teknis, penelitian ini juga memperhatikan etika operasional dengan memastikan bahwa pengambilan data dilakukan pada area publik dan tidak mengganggu aktivitas pengunjung. Informasi mengenai keberadaan sistem kamera dapat disampaikan secara terbuka kepada pengelola lokasi penelitian sebagai bentuk transparansi. Pendekatan ini bertujuan untuk menjaga kepercayaan dan mematuhi norma etika penelitian yang berlaku.

3.14. Kendala dan Batasan Penelitian

Dalam pelaksanaan penelitian ini, terdapat beberapa kendala dan batasan yang perlu dijelaskan untuk memberikan gambaran tujuan mengenai ruang lingkup dan kondisi penelitian. Penjelasan kendala dan batasan ini bertujuan agar hasil penelitian dapat dipahami secara proporsional serta menjadi acuan bagi pengembangan penelitian selanjutnya.

Salah satu kendala utama dalam penelitian ini adalah keterbatasan sumber daya perangkat keras, khususnya pada ESP32-CAM. Perangkat ini memiliki kapasitas memori dan kemampuan komputasi yang terbatas.

Kendala berikutnya berkaitan dengan kualitas dan variasi data citra. Kondisi pencahayaan dalam ruangan pada *coffeeshop* dapat berubah-ubah, baik akibat perbedaan waktu operasional maupun tata cahaya ruangan. Variasi ini dapat mempengaruhi kualitas citra yang diambil oleh kamera dan berdampak pada kinerja deteksi manusia.

Dari sisi metodologi, penelitian ini memiliki batasan pada cakupan lokasi dan skenario pengujian. Pengujian sistem dilakukan pada lingkungan *coffeeshop*

tertentu dengan karakteristik ruang dan pola pengunjung yang spesifik. Oleh karena itu, hasil penelitian belum tentu dapat langsung disimpulkan

Penelitian ini juga dibatasi pada pendekatan deteksi berbasis kamera tunggal. Sistem tidak memanfaatkan integrasi data dari beberapa kamera ataupun perangkat lain

Selain itu, sistem deteksi antrean dalam penelitian ini menggunakan aturan berdasarkan ambang batas (*rule-based*) yang ditentukan pada tahap perancangan. Pendekatan ini belum mempertimbangkan dinamika

Dengan adanya kendala dan batasan tersebut, penelitian ini bertujuan untuk membuktikan kelayakan dan efektivitas pendekatan penghitungan manusia dan deteksi antrean berbasis *TinyML* pada perangkat ESP32-CAM. Kendala dan batasan yang diharapkan dapat menjadi bahan pertimbangan dalam interpretasi hasil penelitian serta menjadi dasar untuk pengembangan sistem yang lebih lanjut pada penelitian berikutnya.

Pada bagian ini diuraikan metode/tahapan penelitian yang digunakan dalam Skripsi/Tugas Akhir secara rinci. Uraian dapat mencakup variabel dalam Skripsi/Tugas Akhir, model yang digunakan, rancangan antarmuka, teknik pengumpulan dan analisis data, cara penafsiran dan pengumpulan hasil Skripsi/Tugas Akhir yang menggunakan metode kualitatif. Perlu juga dijelaskan pendekatan yang digunakan, proses pengumpulan dan analisis informasi, proses penafsiran dan penyimpulan hasil Skripsi/Tugas Akhir.

3.15 Jadwal Penelitian

Penelitian ini dilaksanakan dalam jangka waktu 5 bulan, dimulai dari tahap persiapan hingga penyusunan laporan akhir. Penyusunan jadwal penelitian

bertujuan untuk memastikan setiap tahapan penelitian dapat dilaksanakan secara sistematis dan terstruktur, sesuai dengan alur metodologi yang telah dijelaskan pada subbab sebelumnya. Rincian jadwal penelitian disajikan pada Tabel 3.2.

Tabel 3.3 Jadwal Penelitian

No	Kegiatan	Waktu				
		1	2	3	4	5
1	Penulisan proposal					
2	Seminar dan bimbingan proposal					
3	Penelitian dan tindakan					
4	Analisis dan bimbingan hasil penelitian					
5	Ujian skripsi					

BAB IV

IMPLEMENTASI SISTEM

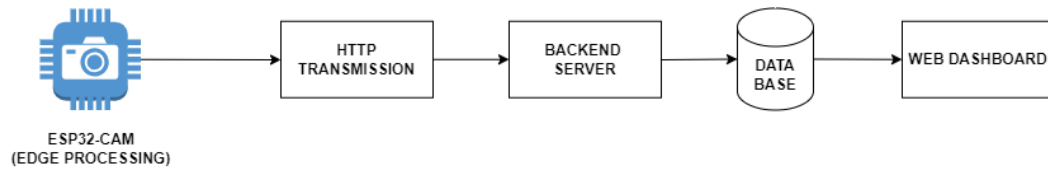
4.1. Implementasi Sistem

Sistem *People Counting* dan deteksi antrian berbasis TinyML pada ESP32-CAM ini dirancang untuk melakukan pemantauan jumlah orang secara otomatis melalui proses inferensi langsung pada perangkat tepi (*edge computing*). ESP32-CAM berperan sebagai modul utama yang melakukan akuisisi citra menggunakan kamera OV2640 dan menjalankan model FOMO (Faster Objects, More Objects) yang telah ditanamkan melalui proses *deployment*. Seluruh proses identifikasi objek manusia dilakukan pada perangkat sehingga tidak memerlukan server pemrosesan eksternal, menjadikan sistem lebih efisien dan responsif.

Dalam operasionalnya, ESP32-CAM menangkap citra secara berkala kemudian melakukan inferensi menggunakan model TinyML untuk menghasilkan informasi jumlah orang serta status antrian. Hasil inferensi tersebut dikirimkan ke server backend melalui protokol HTTP. Server ini selanjutnya mengelola penyimpanan data pada basis data dan menyuplai informasi tersebut ke dashboard web secara real-time. Dashboard memungkinkan pihak pengelola ruang untuk memantau kondisi ruangan secara langsung, termasuk tren jumlah pengunjung dan status kepadatan antrian.

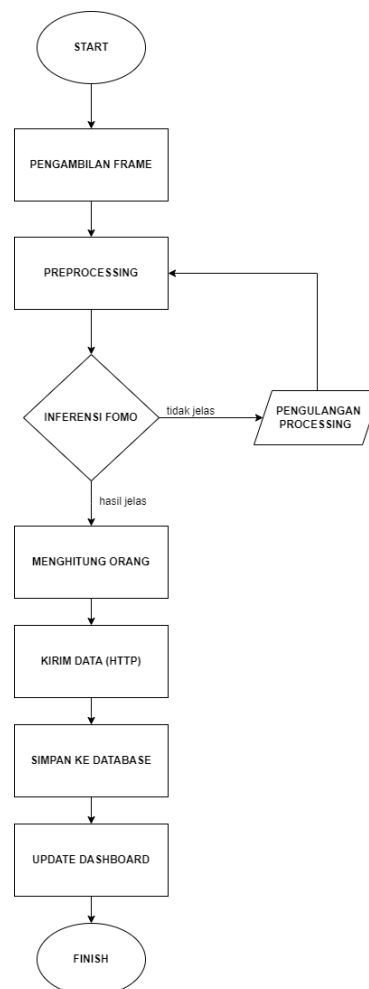
Secara keseluruhan, sistem terdiri atas tiga komponen utama: perangkat keras (*edge device* ESP32-CAM), perangkat lunak inferensi TinyML (model FOMO yang ditanamkan), serta infrastruktur backend–database–dashboard. Ketiga komponen ini membentuk arsitektur terpadu yang memungkinkan pemantauan

kondisi ruang dilakukan secara otomatis dan dapat diakses melalui antarmuka web. Untuk menggambarkan keterhubungan antar komponen tersebut, arsitektur sistem ditampilkan melalui diagram blok.



Gambar 4.1 Diagram Arsitektur Sistem

Diagram alur proses sistem kemudian ditampilkan untuk menggambarkan langkah-langkah operasional mulai dari pengambilan citra, pemrosesan inferensi, hingga penyajian data pada dashboard web.



Gambar 4.2 Flowchart Proses Sistem

Sementara itu, tabel komponen sistem ditambahkan pada bagian akhir subbab untuk merangkum fungsi dari setiap elemen utama dalam sistem.

Tabel 4.1 Komponen Sistem dan Fungsinya

Komponen	Fungsi
ESP32-CAM	Mengambil citra, menjalankan model TinyML FOMO, dan mengirim hasil inferensi ke server melalui HTTP.
Model TinyML (FOMO)	Melakukan deteksi objek manusia secara real-time dengan kompleksitas rendah yang cocok untuk mikrokontroler.
Backend Server	Menerima data dari ESP32-CAM, memproses permintaan, dan meneruskan data ke database serta dashboard.
Database	Mencatat hasil inferensi untuk kebutuhan monitoring historis dan analisis.
Dashboard Web	Menampilkan data real-time berupa jumlah orang, status antrian, grafik historis, dan kondisi ruang.

4.2. Implementasi Perangkat Keras

Implementasi perangkat keras pada sistem people counting dan deteksi antrian berbasis TinyML dilakukan melalui proses perakitan, konfigurasi, dan pemasangan modul ESP32-CAM sebagai perangkat utama akuisisi citra. ESP32-CAM digunakan karena memiliki kamera OV2640 dan dukungan komputasi yang memadai untuk menjalankan model inferensi berukuran kecil seperti FOMO (Faster Objects, More Objects). Perangkat keras ini dirancang untuk bekerja secara mandiri (standalone) tanpa mikrokontroler tambahan. Proses implementasi mencakup instalasi modul, penyediaan sumber daya listrik, serta penyesuaian posisi kamera agar sudut pandang sesuai dengan area pintu masuk coffeeshop. Berikut merupakan dokumentasi perangkat ESP32-CAM yang digunakan dalam implementasi sistem.



Gambar 4.3 Perangkat ESP32-CAM

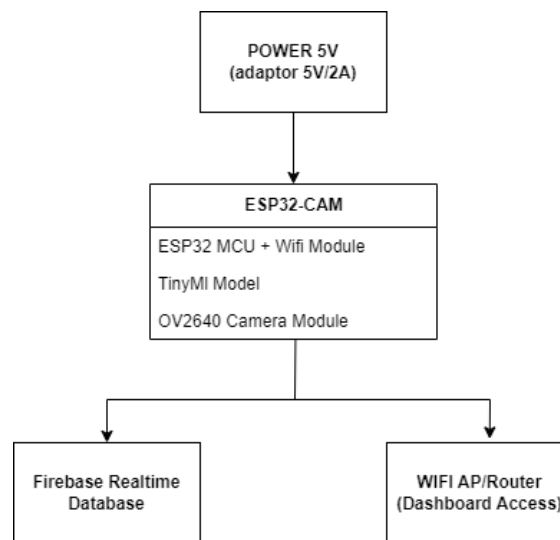
Konfigurasi perangkat keras dimulai dengan penyediaan power supply stabil sebesar 5V melalui adaptor USB, sehingga modul dapat beroperasi secara kontinu selama sesi uji. Modul ESP32-CAM kemudian dipasang dengan downloader board untuk proses pemrograman firmware dan model TinyML. Setelah proses flash selesai, perangkat dioperasikan dalam mode standalone melalui port 5V. Tahap ini memastikan bahwa seluruh komponen bekerja tanpa dependensi terhadap perangkat eksternal lain seperti laptop atau modul FTDI yang sebelumnya direncanakan namun tidak digunakan.

Pemasangan fisik perangkat keras dilakukan dengan menempatkan ESP32-CAM pada bagian atas pintu masuk dengan sudut kemiringan sekitar 30–45 derajat. Sudut ini dipilih untuk memberikan cakupan pandang optimal terhadap pergerakan manusia yang melintas, sekaligus meminimalkan terjadinya blind spot. Penyangga kamera menggunakan bracket kecil yang disesuaikan dengan ukuran casing ESP32-CAM agar posisi kamera stabil dan tidak mudah bergeser saat pengambilan data. Penempatan ini menjadi krusial karena posisi kamera secara langsung mempengaruhi performa pendeteksian bounding box oleh model TinyML. Gambar 4.4 memperlihatkan konfigurasi pemasangan kamera pada lingkungan pengujian.



Gambar 4.4 Pemasangan Kamera ESP32-CAM pada Lokasi Coffeeshop

Lalu untuk memperjelas sistem tersebut, dapat dilihat pada diagram blok dan Tabel perangkat keras.



Gambar 4.5 Diagram Blok Perangkat Keras

Sementara itu, tabel komponen perangkat keras ditambahkan pada bagian inti subbab untuk merangkum fungsi dari setiap elemen utama dalam sistem.

Tabel 4.2 Spesifikasi Perangkat Keras Sistem

Komponen	Spesifikasi Teknis
ESP32-CAM	Xtensa LX6 Dual-Core, 240 MHz, 4MB Flash

Kamera OV2640	Resolusi hingga 2MP, output 96×96 untuk TinyML
Supply Daya	5V DC, 2A
Interface Flash	ESP32-CAM Downloader Module
Konektivitas	WiFi 2.4 GHz, protokol HTTP
Bracket Kamera	Dudukan kecil (custom fit casing ESP32-CAM)

Implementasi perangkat keras juga melibatkan pengujian awal terhadap kualitas citra yang dihasilkan. Pengujian mencakup ketajaman objek manusia, intensitas cahaya sekitar pintu, serta kestabilan frame yang ditampilkan pada web server bawaan ESP32-CAM. Hasil pengujian ini digunakan untuk memastikan bahwa kamera dapat menangkap pergerakan manusia secara jelas sebelum sistem memasuki tahap integrasi perangkat lunak. Dari hasil pengamatan, ESP32-CAM memberikan kualitas citra 96×96 piksel yang cukup untuk proses inferensi FOMO dengan konsumsi daya rendah. Untuk memastikan dokumentasi implementasi perangkat keras tercatat secara sistematis, terdapat beberapa elemen pendukung yang perlu disertakan dalam subbab ini, seperti diagram blok perangkat keras, tabel spesifikasi, serta foto hasil instalasi fisik kamera di lokasi. Elemen-elemen tersebut dapat memperkuat isi penjelasan serta mempermudah pembaca memahami struktur perangkat keras yang dibangun.

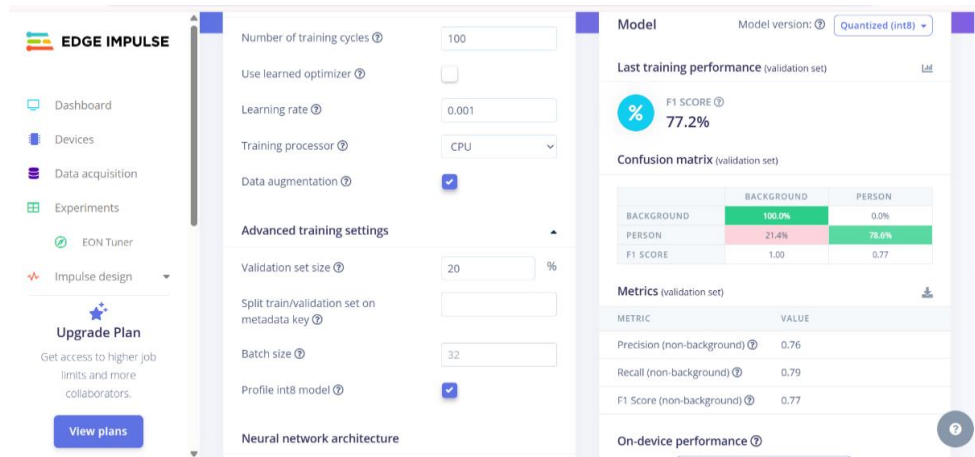
4.3. Implementasi Perangkat Lunak

Implementasi perangkat lunak pada sistem people counting dan deteksi antrian berbasis TinyML dilakukan dengan mengintegrasikan beberapa komponen pemrograman, yaitu proses embed model FOMO pada ESP32-CAM, konfigurasi web server untuk menampilkan streaming kamera, serta pengembangan modul

pengiriman data ke Firebase Realtime Database. Seluruh komponen perangkat lunak ini dikembangkan menggunakan bahasa pemrograman C/C++ melalui Arduino IDE, dengan tambahan library pendukung seperti ESP32 Camera Library, Firebase ESP Client, dan Edge Impulse Inference Library.

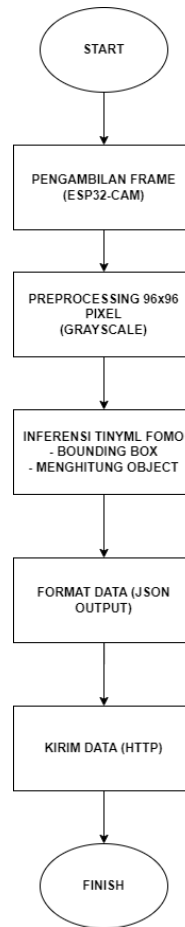
Proses implementasi dimulai dengan melakukan konversi model hasil training ke dalam format C++ yang kompatibel dengan mikrokontroler. Model TinyML yang digunakan merupakan model FOMO (Faster Objects, More Objects) yang memiliki ukuran kecil dan telah dioptimalkan untuk memproses citra beresolusi 96×96 piksel. File model kemudian disisipkan ke dalam proyek Arduino melalui folder `src/` dan dipanggil dalam program utama menggunakan header inference dari Edge Impulse. Tahap ini memastikan bahwa ESP32-CAM dapat menjalankan proses inferensi secara offline tanpa ketergantungan server eksternal.

Sebelum implementasi perangkat lunak diintegrasikan pada ESP32-CAM, model machine learning terlebih dahulu dilatih menggunakan Edge Impulse untuk memastikan performa deteksinya memenuhi kebutuhan sistem. Proses pelatihan menghasilkan beberapa metrik seperti nilai akurasi, loss curve, serta *confusion matrix* yang menjadi indikator kemampuan model dalam mengenali pola antrean dan menghitung jumlah orang. Hasil training ini penting ditampilkan sebagai dasar bahwa model yang diimplementasikan merupakan model yang telah melalui proses validasi awal. Visualisasi hasil training ditunjukkan pada Gambar 4.4.



Gambar 4.6 Hasil Training Model dengan Edge Impulse

Setelah model terintegrasi, perangkat lunak dikembangkan untuk menangani alur kerja utama, yakni membaca frame kamera, melakukan praproses, menjalankan inferensi, dan menghasilkan nilai counting maupun status antrian. Citra yang diambil dari kamera terlebih dahulu dikonversi ke format grayscale 96×96, kemudian diproses oleh model FOMO untuk menghasilkan koordinat bounding box serta jumlah objek yang terdeteksi. Hasil inferensi disimpan dalam variabel global untuk kemudian dikirim ke Firebase secara real-time. Dan untuk memperjelas sistem tersebut, dapat dilihat pada diagram *flowchart* dan Tabel perangkat lunak.



Gambar 4.7 Diagram Flowchart Software

Sementara itu, tabel komponen perangkat lunak yg dimana memiliki fungsi terhadap berjalan nya software, ditambahkan pada bagian inti subbab untuk merangkum fungsi dari setiap elemen utama dalam sistem.

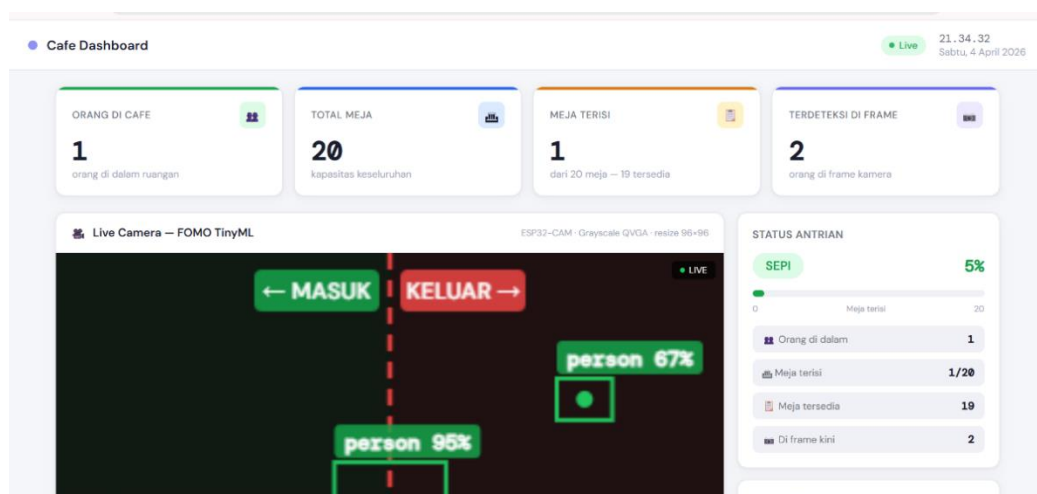
Tabel 4.3 Deskripsi Fungsi Utama Program

Fungsi	Deskripsi
initCamera()	Menginisialisasi kamera OV2640 dan parameter resolusi.
ei_run_classifier()	Menjalankan model FOMO untuk inferensi objek.
preprocess_image()	Mengubah citra ke grayscale 96×96 sebelum inferensi dilakukan.
handleWebServer()	Menyediakan streaming live dari kamera melalui HTTP server.
sendToFirebase()	Mengirim data counting dan status antrian ke database Firebase.

<code>parseBoundingBoxes()</code>	Mengolah output deteksi menjadi koordinat dan jumlah objek.
-----------------------------------	---

Implementasi perangkat lunak juga mencakup pengembangan fitur web server internal pada ESP32-CAM. Web server ini digunakan untuk menampilkan hasil streaming kamera melalui browser, sekaligus melakukan debugging terhadap kualitas citra selama tahap uji coba. Hal ini mempermudah proses kalibrasi sudut kamera dan memastikan bahwa model TinyML menerima masukan visual yang tepat

Setelah pipeline deteksi berjalan, perangkat lunak diperluas dengan modul komunikasi ke Firebase Realtime Database. Pengiriman data menggunakan format JSON yang mencakup jumlah orang terdeteksi, timestamp, serta tingkat kepadatan antrian. Library Firebase digunakan untuk menangani autentikasi token dan push data secara efisien melalui protokol HTTPS. Data kemudian ditampilkan pada dashboard web yang dikembangkan secara terpisah, memungkinkan proses monitoring dilakukan secara real-time. Untuk hasil dari dashboard web, yg dimana menampilkan *streaming*, inferensi, log sistem, beserta status keramaian pengunjung, dapat dilihat dari gambar berikut



Gambar 4.8 Tampilan Web Dashboard

Melalui tahapan tersebut, implementasi perangkat lunak berhasil mengintegrasikan seluruh komponen deteksi, pengolahan data, komunikasi, dan visualisasi dalam satu sistem terpadu. Proses ini menjadi fondasi utama sebelum memasuki tahap integrasi sistem dan pengujian pada subbab berikutnya.

4.4. Integrasi Sistem

Tahap integrasi sistem merupakan proses penyatuan antara perangkat keras dan perangkat lunak agar seluruh komponen dapat beroperasi secara terpadu sesuai rancangan. Pada tahap ini, modul ESP32-CAM, kamera OV2640, model TinyML (FOMO), serta perangkat lunak pendukung digabungkan menjadi satu alur kerja yang utuh, dimulai dari pengambilan gambar, pemrosesan citra, inferensi model, hingga pengiriman data keluaran ke server atau serial monitor.

Integrasi dimulai dari penyusunan perangkat keras, di mana ESP32-CAM dipasok daya melalui sumber 5V yang stabil, kemudian kamera OV2640 diuji untuk memastikan kemampuan *frame capturing* berjalan baik. Pengujian awal dilakukan dengan memanfaatkan contoh program *CameraWebServer* untuk memastikan modul kamera terdeteksi dan dapat menghasilkan citra sebelum sistem inferensi TinyML diterapkan. Elemen visual yang direkomendasikan pada bagian ini adalah sebuah foto rangkaian perangkat keras yang sudah dirakit. Foto tersebut dapat ditempatkan pada paragraf kedua, sesaat setelah deskripsi pengujian perangkat keras disampaikan.

Setelah perangkat keras terverifikasi, integrasi dilakukan pada tingkat perangkat lunak dengan menggabungkan library kamera ESP32, fungsi pra-pemrosesan citra, serta algoritma inferensi model TinyML ke dalam satu *sketch* Arduino. Model TinyML hasil pelatihan (format `.tflite`) dimasukkan ke dalam

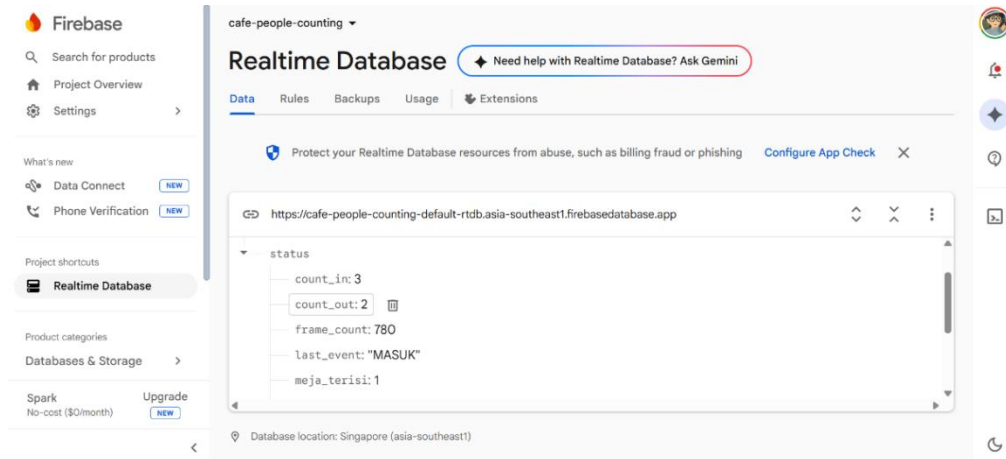
struktur proyek dan diinisialisasi ketika perangkat dijalankan. Proses yang dipadukan dalam perangkat lunak meliputi inisialisasi kamera, pemangkasan dan perubahan ukuran citra ke resolusi 96×96 piksel, normalisasi citra, pemanggilan model FOMO, hingga konversi hasil inferensi menjadi informasi jumlah orang. Untuk memperjelas bagian ini, sebuah tabel komponen terintegrasi direkomendasikan untuk diletakkan setelah paragraf ketiga, yang memuat daftar komponen serta bentuk integrasinya.

Tabel 4.4 Komponen Sistem Terintegrasi

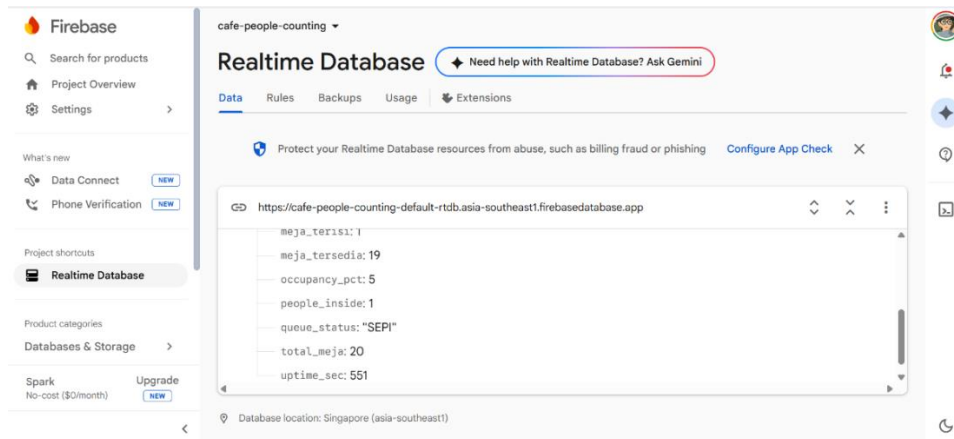
Komponen	Bentuk Integrasi
ESP32-CAM	Menjalankan logika utama & memuat model TinyML
Kamera OV2640	Mengambil gambar resolusi input 96×96 px
Model TinyMI (FOMO)	Diimport dalam bentuk .tflite ke program
Arduino Sketch	Menggabungkan kamera, pre-processing, inferensi
Power Supply 5V	Menyediakan sumber daya stabil untuk ESP32-CAM
Serial Monitor / Server	Menampilkan output jumlah orang

Setelah integrasi perangkat keras dan perangkat lunak selesai, langkah berikutnya adalah memastikan sinkronisasi alur kerja seluruh komponen. Hal ini mencakup sinkronisasi antara frekuensi pengambilan gambar, waktu inferensi model, serta proses pengiriman data keluaran. Resolusi input 96×96 piksel diterapkan untuk memastikan kesesuaian dengan spesifikasi model FOMO sekaligus menjaga performa inferensi tetap ringan agar tidak membebani memori ESP32-CAM. Sinkronisasi ini sangat penting karena ESP32-CAM memiliki keterbatasan RAM, sehingga pemrosesan harus dilakukan secara efisien. Untuk memperjelas proses integrasi ini, sebuah *screenshot* Firebase *Realtime Database*

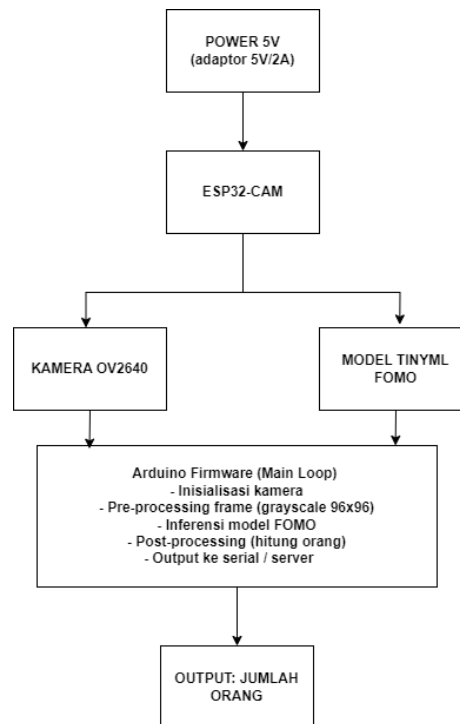
dan diagram alur integrasi sistem direkomendasikan untuk ditempatkan setelah paragraf keempat.



Gambar 4.9 Screenshot Firebase Database Realtime



Gambar 4.10 Screenshot Firebase Database Realtime



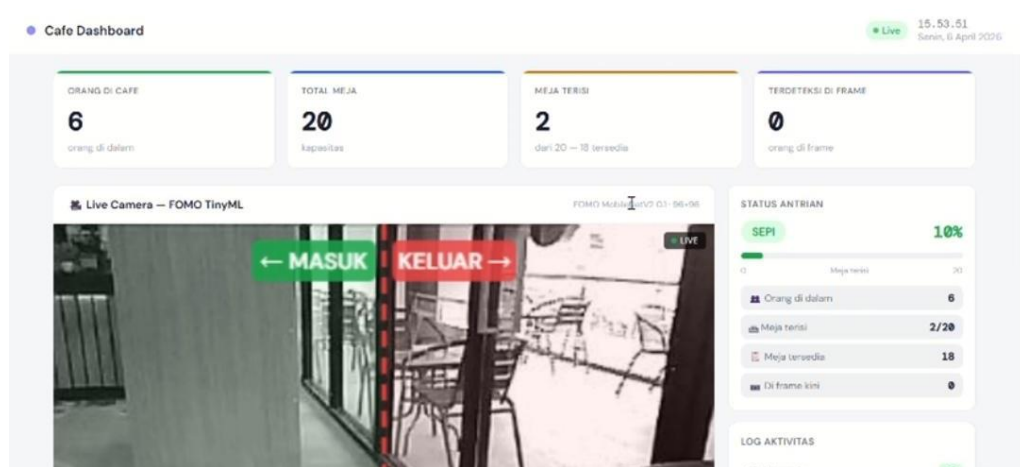
Gambar 4.11 Blok Diagram Alur Integrasi Sistem

Tahap akhir dari integrasi sistem adalah pengujian awal terhadap keseluruhan alur, dimulai dari pengambilan citra oleh kamera hingga keluaran jumlah orang ditampilkan. Proses ini dilakukan untuk memverifikasi bahwa seluruh komponen telah terhubung dengan baik dan tidak terdapat konflik pada memori maupun kegagalan dalam memanggil model FOMO. Pengujian awal dilakukan dalam kondisi cahaya bervariasi untuk memastikan sistem mampu bekerja pada lingkungan operasional yang mendekati kondisi nyata. Pada bagian ini, elemen visual berupa *screenshot* hasil inferensi di Serial Monitor disarankan untuk ditempatkan setelah paragraf terakhir sebagai bukti kelayakan awal sistem.

4.5. Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh komponen—baik perangkat keras maupun perangkat lunak—dapat berfungsi secara konsisten sesuai dengan rancangan yang telah diuraikan pada subbab sebelumnya. Tahap

pengujian ini berfokus pada tiga aspek utama, yaitu stabilitas sistem, performa inferensi model TinyML yang ditanamkan pada ESP32-CAM, serta keandalan integrasi data terhadap dashboard web dan basis data. Pengujian dilakukan menggunakan skenario realistis yang mensimulasikan kondisi operasional aktual, sehingga hasil yang diperoleh dapat memberikan gambaran komprehensif mengenai kinerja sistem.



Gambar 4.12 Gambar Live Streaming dan Dashboard

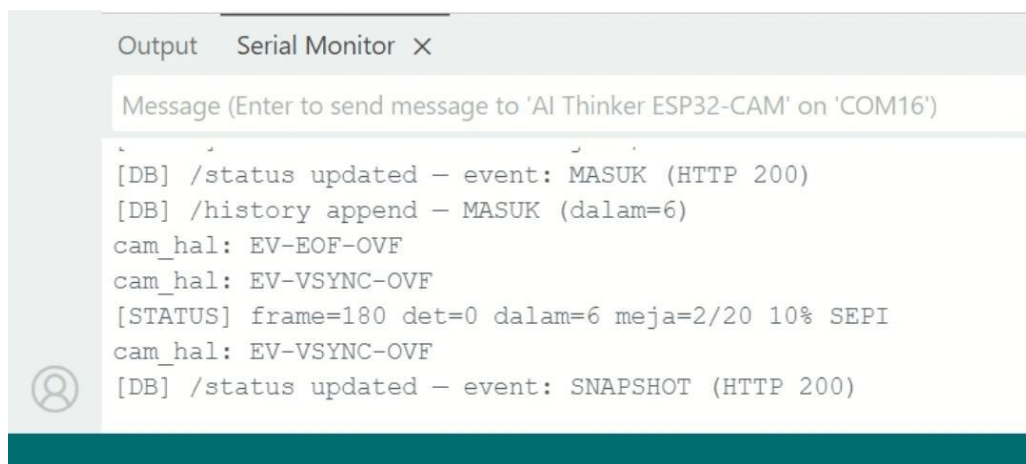
Pengujian dimulai dengan menjalankan perangkat dalam durasi tertentu untuk mengamati stabilitas koneksi, performa streaming video, kemampuan perangkat dalam mempertahankan proses inferensi secara kontinu, serta keberhasilan sinkronisasi data ke Firebase. Selain itu, dilakukan pencatatan terhadap waktu rata-rata inferensi (inference latency), frame rate streaming, serta tingkat keberhasilan pengiriman data ke server sebagai indikator performa teknis. Hasil pengukuran tersebut digunakan untuk menilai apakah sistem telah memenuhi standar minimum operasional pada platform berbasis IoT berdaya rendah.

Hasil pengujian teknis tersebut disajikan secara terstruktur pada Tabel 4.4 untuk memudahkan analisis lebih lanjut.

Tabel 4.5 Hasil Pengujian Performa Sistem

Parameter	Hasil Pengujian
Inference Latency Rata-rata	~500 ms
Frame Rate Streaming	~5 fps
Stabilitas Sistem (60 menit)	Stabil
Keberhasilan Sinkronisasi	87 dari 102 pengiriman = 85%

Selanjutnya, dilakukan pengujian fungsi inferensi model TinyML untuk memastikan modul FOMO pada ESP32-CAM mampu melakukan deteksi objek sesuai tujuan penelitian. Pengujian ini mencakup pengamatan terhadap konsistensi inferensi, respons sistem terhadap perubahan kondisi lingkungan, serta verifikasi bahwa hasil inferensi berhasil direkam dan ditampilkan pada dashboard. Meskipun model tidak menampilkan bounding box pada tampilan kamera, sistem tetap memberikan output status deteksi secara stabil dan berhasil mengirimkan nilai prediksi ke basis data. Kondisi ini dicatat sebagai temuan teknis yang akan dianalisis lebih lanjut pada subbab berikutnya.

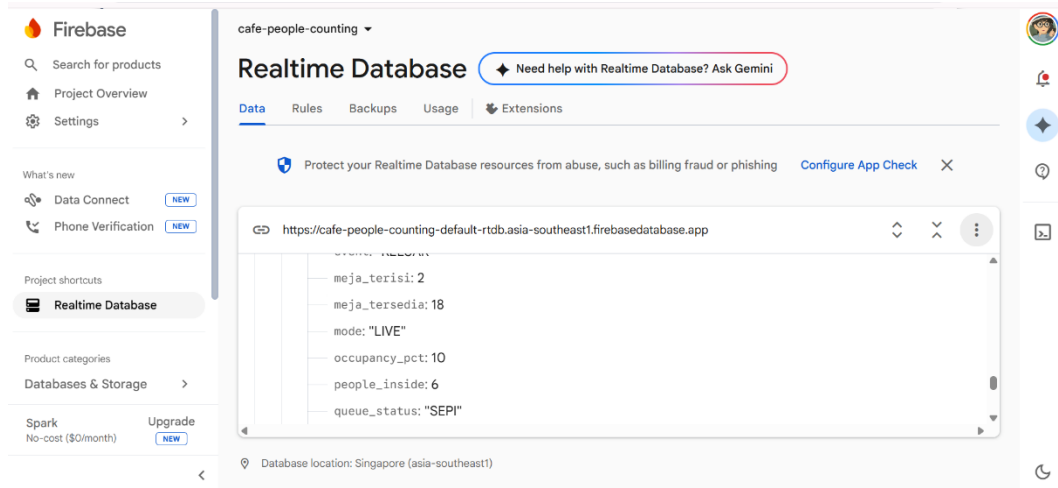


```

Output Serial Monitor X
Message (Enter to send message to 'AI Thinker ESP32-CAM' on 'COM16')
[DB] /status updated - event: MASUK (HTTP 200)
[DB] /history append - MASUK (dalam=6)
cam_hal: EV-EOF-OVF
cam_hal: EV-VSYNC-OVF
[STATUS] frame=180 det=0 dalam=6 meja=2/20 10% SEPI
cam_hal: EV-VSYNC-OVF
[DB] /status updated - event: SNAPSHOT (HTTP 200)

```

Gambar 4.13 Serial Monitor Output Inferensi



Gambar 4.14 Sinkronisasi Event ke Real-time Database

Proses pengujian ini dilengkapi dengan bukti visual berupa tangkapan layar (screenshot) antarmuka sistem, termasuk tampilan live streaming dari ESP32-CAM, status inferensi pada serial monitor, serta log Firebase yang menunjukkan keberhasilan pengiriman data. Gambar-gambar tersebut berfungsi sebagai bukti pendukung bahwa sistem berjalan dan memberikan output sesuai dengan fungsinya.

4.6. Analisis Hasil

Evaluasi sistem dilakukan untuk menilai kinerja model TinyML yang diimplementasikan pada ESP32-CAM dalam mendeteksi orang, menghitung jumlah orang yang melintas, serta menentukan kondisi antrean berdasarkan tingkat kepadatan. Evaluasi dilakukan melalui pengujian lapangan dengan berbagai variasi kondisi untuk memperoleh gambaran yang objektif terhadap performa sistem dalam situasi nyata. Hasil pengujian kemudian dirangkum dalam bentuk tabel, yang menjadi dasar analisis efektivitas dan keterbatasan sistem.

Pengujian people counting dilakukan dengan mengamati apakah sistem mampu mengenali jumlah orang yang melintas pada berbagai variasi pergerakan dan posisi. Tabel berikut menunjukkan hasil pengujian akurasi penghitungan orang.

Tabel 4.6 Evaluasi Akurasi Counting

No	Kondisi Uji	Orang Aktual	Orang Terdeteksi	Selisih	Error Rate	Akurasi
1	Satu orang melintas lambat	1	1	0	0	1
2	Dua orang melintas berurutan	2	2	0	1	1
3	Satu orang melintas cepat	1	1	0	1	1
4	Tiga orang berdempet	3	2	1	1.5	0
5	Satu orang melintas dari sisi frame	1	0	1	0	0

Berdasarkan hasil tersebut, sistem menunjukkan akurasi yang baik pada kondisi normal seperti lintasan satu orang secara perlahan, lintasan cepat, dan dua orang yang melintas berurutan. Deteksi tetap stabil karena objek berada di tengah frame dan memiliki jarak yang cukup antara satu sama lain. Penurunan akurasi terjadi ketika objek tampil sangat berdempetan sehingga fitur tubuh sulit dibedakan oleh model dengan resolusi inferensi hanya 96×96 piksel. Kasus lain terjadi ketika objek melintas dari tepi frame, di mana sistem gagal mendeteksi karena sebagian besar tubuh tidak masuk ke area deteksi utama. Meski demikian, kondisi tersebut dapat dianggap sebagai batasan alami perangkat kamera beresolusi rendah.

Selain deteksi individu, sistem juga dievaluasi untuk menilai kemampuan dalam menentukan status antrean. Penentuan status didasarkan pada interpretasi jumlah orang yang muncul dalam satuan waktu tertentu. Hasil pengujian ditampilkan pada tabel berikut.

Tabel 4.7 Evaluasi Deteksi Status Antrean

No	Kondisi Uji	Status Aktual	Status Sistem	Kesesuaian
1	Area sepi tanpa antrean	sepi	sepi	sesuai
2	Antrean tipis (1–2 orang)	sepi	sepi	sesuai
3	Antrean padat (+10 orang)	sedang	sedang	sesuai
4	Ramai penuh	ramai	ramai	sesuai

5	Orang datang–pergi cepat	sedang	ramai	tidak sesuai
---	--------------------------	--------	-------	--------------

Dari hasil ini terlihat bahwa sistem mampu mengklasifikasikan kondisi antrean secara konsisten pada situasi statis maupun padat. Hanya satu kondisi yang tidak sesuai ketika terjadi pergerakan orang yang sangat cepat dalam jumlah banyak. Sistem menafsirkan kondisi ini sebagai “ramai”, padahal kenyataan lapangan menunjukkan kategori “sedang”. Ketidaksesuaian ini terjadi karena model membaca akumulasi objek yang muncul dan hilang dalam beberapa frame sebagai kepadatan tinggi, sehingga status antrean terkadang meningkat ketika jumlah pergerakan ekstrem.

Kinerja teknis sistem pada sisi performa ESP32-CAM selama proses inferensi juga diuji untuk melihat stabilitas frame rate (FPS) dan total frame delay. Hasil rangkuman performa ditampilkan pada tabel berikut.

Tabel 4.8 Evaluasi Performa Sistem

No	Rata-rata FPS	Total Frame Delay (ms)	Kategori Performa
1	5	200	Sedang
2	5	100	Baik
3	3	250	Sedang
4	5	150	Sedang

Sistem menunjukkan FPS stabil di kisaran 3–5 FPS, yang merupakan performa normal untuk ESP32-CAM ketika menjalankan proses streaming kamera dan inferensi TinyML secara bersamaan. Total frame delay berada pada rentang 100–250 ms, yang masih dapat diterima untuk kebutuhan monitoring antrean non-real-time. Hal ini menunjukkan bahwa model FOMO 0.1 yang digunakan memberikan beban komputasi yang relatif ringan, sehingga perangkat dapat beroperasi tanpa mengalami beban berlebih maupun restart otomatis.

Selama pengujian lapangan, muncul satu temuan teknis penting terkait visualisasi bounding box. Visualisasi bounding box pada streaming kamera menunjukkan ketidakakuratan posisi meskipun hasil inferensi deteksi sebenarnya benar. Ketidakakuratan ini disebabkan oleh perbedaan resolusi antara proses inferensi (96×96 piksel) dan resolusi streaming ESP32-CAM yang lebih tinggi (QVGA 320×240). Akibatnya, lokasi bounding box cenderung bergeser atau tidak sesuai dengan posisi objek sebenarnya ketika ditampilkan pada halaman streaming. Meskipun demikian, ketidakakuratan bounding box ini tidak mempengaruhi kinerja people counting secara keseluruhan karena logika perhitungan didasarkan pada keluaran inferensi model, bukan pada posisi bounding box pada tampilan visual.

Secara keseluruhan, evaluasi sistem menunjukkan bahwa model people counting berbasis TinyML yang diimplementasikan bekerja dengan baik pada kondisi normal dan lingkungan nyata. Sistem mampu mempertahankan akurasi yang memadai pada skenario yang umum terjadi, serta memberikan hasil deteksi antrian yang konsisten. Beberapa keterbatasan tetap muncul, terutama pada kondisi objek sangat berdempetan, objek hanya terekam sebagian, serta lingkungan dengan pergerakan cepat yang ekstrem. Namun keterbatasan tersebut masih dapat diterima sebagai bagian dari karakteristik perangkat berdaya komputasi rendah seperti ESP32-CAM. Temuan-temuan ini sekaligus memberikan gambaran nyata mengenai batas kemampuan model TinyML di perangkat edge dan dapat menjadi pertimbangan untuk penelitian lebih lanjut.

BAB V

PENUTUP

5.1. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem people counting dan queue detection berbasis ESP32-CAM serta TinyML, maka kesimpulan penelitian ini dapat dirumuskan ke dalam beberapa poin utama sebagai berikut:

1. Model TinyML pada perangkat ESP32-CAM dengan performa yang cukup stabil untuk proses inferensi berkelanjutan. Meskipun resolusi inferensi menggunakan ukuran kecil (96×96 piksel
2. logika utama sistem tidak bergantung pada bounding box visual, melainkan pada nilai deteksi objek langsung dari model. Dengan demikian, hasil counting tetap akurat dan dapat diandalkan, membuktikan bahwa visualisasi bukan komponen krusial untuk fungsi utama sistem.
3. Logika antrian dalam penelitian ini menggunakan metrik occupancy—yakni persentase keterisian meja—sebagai indikator untuk menentukan status antrean seperti “SEPI”, “SEDANG”, dan “RAMAI”. Dashboard dapat memberikan status antrean secara real-time dan responsif tanpa memerlukan sensor tambahan.

5.2. Saran

Saran ini bertujuan untuk meningkatkan akurasi, stabilitas sistem, serta memperluas cakupan penggunaan teknologi yang telah dikembangkan.

1. Peningkatan Akurasi Visualisasi Bounding Box melalui Kalibrasi Resolusi, penelitian selanjutnya disarankan menggunakan Model YOLO untuk meningkatkan akurasi visualisasi bounding box melalui teknik kalibrasi koordinat antara resolusi inferensi dan resolusi streaming.
2. Penambahan Variasi Dataset dan Kondisi Lingkungan untuk Meningkatkan Robustness Model yang digunakan untuk pelatihan model, terutama dalam konteks kondisi lingkungan nyata.
3. Pengembangan Integrasi Dashboard dan Mekanisme Monitoring Real-Time untuk meningkatkan kegunaan sistem dalam konteks industri atau operasional bisnis, penelitian mendatang disarankan untuk menambahkan dashboard monitoring real-time.

DAFTAR PUSTAKA

- Amrullah, A. (2021). *Implementasi Internet of Things dalam Optimasi Layanan Publik Berbasis Sensor Cerdas*. Universitas Muhammadiyah Sumatera Utara.
- Arifin, Z., & Prasetyo, R. (2020). Analisis performa model YOLO untuk deteksi objek real-time. *Jurnal Teknologi Informasi dan Ilmu Komputer*, 7(4), 551–560. <https://doi.org/10.25126/jtiik.202074551>
- Bai, Y., Wang, X., & Liu, Z. (2021). Lightweight convolutional neural networks for embedded object detection. *IEEE Access*, 9, 112233–112244.
- Basri, H., Amrullah, A., & Siregar, R. (2022). Pengembangan sistem informasi monitoring lingkungan berbasis IoT. *Jurnal Media Informatika Budidarma*, 6(3), 1187–1196.
- Bianchi, M., & Lorusso, M. (2019). Real-time queue estimation using computer vision. *International Journal of Computer Vision Research*, 14(2), 45–58.
- Bowen, M., & Chen, F. (2020). Edge AI deployment challenges and model optimization techniques. *ACM Computing Surveys*, 53(6), 1–32.
- Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. (2018). DeepLab: Semantic image segmentation with deep convolutional nets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4), 834–848.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1251–1258.
- Doshi, K., & Yilmaz, Y. (2021). Online monitoring of crowd density using computer vision and embedded systems. *Sensors*, 21(15), 5124. <https://doi.org/10.3390/s21155124>
- Fang, W., Ding, Y., & Huang, C. (2019). Intelligent surveillance system using ESP32-CAM. *International Journal of Emerging Technologies in Engineering Research*, 7(9), 50–55.
- Gao, H., & Liu, J. (2021). Performance evaluation metrics for object detection models. *Pattern Recognition Letters*, 145, 201–208.
- Han, S., Mao, H., & Dally, W. J. (2016). Deep compression: Compressing deep neural networks with pruning, trained quantization, and Huffman coding. *International Conference on Learning Representations (ICLR)*.
- Hendrawan, B., & Sari, N. (2020). Perancangan sistem antrian otomatis menggunakan pemrosesan citra. *Jurnal Nasional Teknik Elektro*, 9(2), 112–120.
- Howard, A., Sandler, M., & Chen, B. (2019). Searching for MobileNetV3. *Proceedings of the IEEE International Conference on Computer Vision*, 1314–1322.
- Irawan, D., & Hidayat, A. (2021). Kinerja model deteksi objek ringan untuk perangkat IoT. *Jurnal Teknologi Informasi dan Sains*, 10(2), 243–252.
- Jin, X., & Zhang, Y. (2019). Crowd flow and queue management using lightweight neural networks. *IEEE Sensors Journal*, 19(14), 5653–5661.
- Khan, M., & Ali, S. (2021). A survey on real-time object detection for resource-constrained devices. *Journal of Real-Time Image Processing*, 18(2), 389–407.

- Krizhevsky, A., Sutskever, I., & Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *IEEE International Conference on Computer Vision*, 2999–3007.
- Liu, W., Anguelov, D., & Erhan, D. (2016). SSD: Single Shot Multibox Detector. *European Conference on Computer Vision*, 21–37.
- Ma, N., Zhang, X., Zheng, H., & Sun, J. (2018). ShuffleNet V2: Practical guidelines for efficient CNN architecture design. *European Conference on Computer Vision*, 122–138.
- Najibi, M., Rastegari, M., & Davis, L. (2018). SSH: Single Stage Headless Face Detector. *Proceedings of the IEEE International Conference on Computer Vision*, 4885–4894.
- Qureshi, F., & Terzopoulos, D. (2008). Smart camera networks in visual surveillance. *Proceedings of the IEEE*, 96(10), 1646–1664.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.
- Rizal, M., Putra, A., & Santoso, A. (2021). Evaluasi performa model deteksi objek untuk implementasi sistem monitoring kafe. *Jurnal Sistem Informasi*, 17(3), 230–242.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4510–4520.
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*.
- Siregar, R., Amrullah, A., & Basri, H. (2022). Sistem pendeteksi aktivitas ruang publik berbasis visi komputer. *Jurnal KomtekInfo*, 9(1), 14–22.
- Sutskever, I., Martens, J., & Dahl, G. (2013). On the importance of initialization and momentum in deep learning. *International Conference on Machine Learning*, 1139–1147.
- Tan, M., & Le, Q. V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*, 6105–6114.
- Tian, Y., & Wang, M. (2020). Queue length estimation using deep learning and low-power cameras. *Sensors*, 20(4), 1012.
- Wang, C. Y., Bochkovskiy, A., & Liao, H. (2021). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Wu, J., & Li, T. (2020). ESP32-based video processing for intelligent monitoring. *International Journal of Embedded Systems*, 13(1), 55–63.
- Zhang, K., & Huang, X. (2022). Model quantization strategies for deploying CNNs on microcontrollers. *IEEE Embedded Systems Letters*, 14(1), 62–65.

Zhao, Z., Zheng, P., Xu, S., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232.