

TinyML pada ESP32-CAM untuk Deteksi Objek Real-Time: Sebuah Tinjauan Literatur Sistematis

Mohamad Ardi^{1*}, Kandi Harianto²

^{1,2}Teknik Informatika, STMIK PPKIA Tarakanita Rahmawati, Tarakan, Kalimantan Utara
Email: ^{1*}mohamadardi@ppkia.ac.id, ²kandi@ppkia.ac.id

Abstrak

Perkembangan kecerdasan buatan pada perangkat tepi (*edge device*) telah mendorong munculnya paradigma *Tiny Machine Learning* (TinyML) yang memungkinkan inferensi model pembelajaran mesin berjalan langsung pada mikrokontroler dengan sumber daya terbatas, tanpa bergantung pada komputasi awan. Meskipun sejumlah studi telah menerapkan TinyML pada modul ESP32-CAM secara terpisah, belum terdapat kajian yang merangkum secara sistematis arsitektur model, teknik optimasi, dan kinerja yang dilaporkan, sehingga gambaran komprehensif mengenai capaian dan keterbatasan pendekatan tersebut masih sulit diperoleh. Penelitian ini bertujuan untuk mengkaji secara sistematis penerapan TinyML pada modul ESP32-CAM untuk keperluan deteksi objek secara *real-time*, khususnya dalam hal arsitektur model, teknik optimasi, dan kinerja sistem yang dihasilkan. Metode yang digunakan adalah *Systematic Literature Review* (SLR) dengan tahapan perencanaan, pelaksanaan pencarian pada basis data ilmiah, serta penyaringan berdasarkan kriteria inklusi dan eksklusi. Sintesis dilakukan terhadap 20 artikel terpilih yang dipublikasikan antara tahun 2021 hingga 2026. Hasil kajian menunjukkan bahwa arsitektur *Faster Objects, More Objects* (FOMO) dan model terkuantisasi berbasis *TensorFlow Lite for Microcontrollers* (TFLM) menjadi pendekatan dominan karena mampu menekan ukuran model hingga di bawah 500 KB dengan tetap menjaga akurasi deteksi di atas 90% pada kondisi lingkungan terkendali. Teknik kuantisasi INT8 dan pemangkasan jaringan (*pruning*) terbukti efektif menurunkan kebutuhan RAM dan ROM masing-masing hingga lebih dari 25% tanpa penurunan akurasi yang signifikan. Kajian ini juga mengidentifikasi keterbatasan utama berupa sensitivitas terhadap kondisi pencahayaan dan latensi inferensi yang masih bervariasi antara 300 hingga 700 milidetik. FOMO dan kuantisasi INT8 terbukti menjadi pendekatan yang paling efektif untuk diterapkan pada ESP32-CAM, meskipun tantangan sensitivitas pencahayaan dan variasi latensi inferensi masih perlu diatasi pada penerapan lapangan. Penelitian ini diharapkan dapat menjadi rujukan bagi peneliti dan praktisi dalam merancang sistem deteksi objek berbasis TinyML yang efisien pada perangkat ESP32-CAM untuk aplikasi pemantauan, robotika, dan keamanan berbiaya rendah.

Kata Kunci: Deteksi Objek, Edge AI, ESP32-CAM, Systematic Literature Review, TinyML.

TinyML on ESP32-CAM for Real-Time Object Detection: A Systematic Literature Review

Abstract

The advancement of artificial intelligence on edge devices has driven the emergence of the *Tiny Machine Learning* (TinyML) paradigm, which allows machine learning inference to run directly on resource-constrained microcontrollers without relying on cloud computing. Although a number of studies have applied TinyML on the ESP32-CAM module separately, no study has yet systematically synthesized the reported model architectures, optimization techniques, and performance outcomes, leaving a fragmented picture of the achievements and limitations of this approach. This study aims to systematically examine the application of TinyML on the ESP32-CAM module for real-time object detection purposes, particularly regarding model architecture, optimization techniques, and resulting system performance. The method used is a *Systematic Literature Review* (SLR) comprising planning, search execution across scientific databases, and screening based on inclusion and exclusion criteria. Synthesis was performed on 20 selected articles published between 2021 and 2026. The results show that the *Faster Objects, More Objects* (FOMO) architecture and quantized models based on *TensorFlow Lite for Microcontrollers* (TFLM) are the dominant approaches, as they can reduce model size to below 500 KB while maintaining detection accuracy above 90% under controlled environmental conditions. INT8 quantization and network pruning techniques have proven effective in reducing RAM and ROM requirements by more than 25% without significant accuracy degradation. This study also identifies key limitations, namely sensitivity to lighting conditions and inference latency that still varies between 300 and 700 milliseconds. FOMO and INT8 quantization proved to be the most effective approaches for ESP32-CAM, although challenges related to lighting sensitivity and variable inference latency still need to be addressed in field deployment. This research is expected to

serve as a reference for researchers and practitioners in designing efficient TinyML-based object detection systems on ESP32-CAM devices for low-cost monitoring, robotics, and security applications.

Keywords: Object Detection, Edge AI, ESP32-CAM, Systematic Literature Review, TinyML.

I. PENDAHULUAN

Kemajuan teknologi informasi yang berlangsung secara masif pada saat ini telah memicu transformasi mendasar di berbagai aspek kehidupan manusia [1], di mana beragam inovasi digital telah diterapkan secara luas pada sektor-sektor strategis seperti telekomunikasi, transportasi, kesehatan, pendidikan, hingga pertahanan dan keamanan nasional [2], [3], sejalan dengan meluasnya digitalisasi yang turut memicu pertumbuhan volume data secara eksponensial sehingga data tersebut bergeser menjadi aset penting yang perlu diolah menjadi informasi, pola, dan pengetahuan baru untuk mendukung ketepatan pengambilan keputusan strategis [4]. Meskipun demikian, karakteristik data modern yang masif, dinamis, dan fluktuatif menimbulkan tantangan tersendiri dalam hal prediktabilitas, sementara kebutuhan operasional di berbagai sektor industri menuntut ketersediaan informasi yang akurat dan dapat diakses secara *real-time* guna menjamin efisiensi serta peningkatan kualitas layanan secara berkelanjutan [5].

Kebutuhan akan pengolahan data secara cepat dan *real-time* tersebut mendorong berkembangnya paradigma komputasi cerdas yang menempatkan proses analisis lebih dekat dengan sumber data melalui perangkat *edge*. Seiring dengan pertumbuhan jumlah perangkat *Internet of Things* (IoT) yang diproyeksikan mencapai puluhan miliar unit, kebutuhan akan kecerdasan buatan yang mampu berjalan secara mandiri pada perangkat tepi semakin meningkat karena pengiriman seluruh data ke *server* atau *cloud* tidak selalu efisien maupun memungkinkan [6]. Paradigma ini dikenal sebagai TinyML, yaitu cabang pembelajaran mesin yang dirancang khusus untuk berjalan pada mikrokontroler dan perangkat *edge* berdaya rendah dengan keterbatasan memori serta daya komputasi yang signifikan dibandingkan komputer atau server konvensional.

Mikrokontroler ESP32-CAM menjadi salah satu platform yang paling banyak digunakan dalam riset TinyML untuk visi komputer karena menggabungkan unit pemrosesan dua inti berkecepatan tinggi, modul Wi-Fi terintegrasi, dan kamera dalam satu papan berbiaya rendah. Sejumlah riset terdahulu telah menunjukkan keberhasilan penerapan deteksi objek berbasis ESP32-CAM, salah satunya pada konteks kontes robot SAR Indonesia yang memanfaatkan platform *Edge Impulse* dengan arsitektur *Faster Objects, More Objects* (FOMO) untuk membedakan boneka korban dan boneka biasa dengan tujuan menyelamatkan boneka korban dan membawanya ke zona aman [7]. Riset tersebut melaporkan bahwa FOMO merupakan algoritma deteksi objek yang dirancang agar dapat berjalan secara efisien pada perangkat dengan keterbatasan sumber daya seperti mikrokontroler ESP32-CAM.

Selain pada konteks robotika pencarian dan penyelamatan, ESP32-CAM juga telah diuji dalam berbagai skenario deteksi visual lain, seperti deteksi orang untuk sistem keamanan tanpa koneksi *cloud* yang memanfaatkan model *MobileNet*

terkuantisasi yang berjalan langsung pada CPU ESP32 sehingga memungkinkan deteksi orang secara *real-time* pada sisi *edge* tanpa dependensi *cloud* [8], meskipun rata-rata waktu inferensi yang dilaporkan pada studi tersebut masih relatif tinggi yaitu sekitar 1018 milidetik per bingkai akibat keterbatasan daya komputasi ESP32-CAM. Performa inferensi pada chip ini bervariasi antarstudi tergantung arsitektur model dan resolusi citra yang digunakan [9], mengingat kerangka kerja TFLM dirancang dengan pendekatan interpreter ringan yang meminimalkan overhead runtime pada perangkat beresolusi memori terbatas.

Penelitian lain pada konteks deteksi kerusakan jalan turut memperkuat bukti kelayakan ESP32-CAM untuk tugas visi komputer, di mana penggunaan algoritma YOLOv8 berbasis ESP32-CAM pada deteksi lubang jalan menunjukkan sistem dapat mencapai F1-Score sebesar 0,73 dengan rata-rata waktu inferensi antara 300-350 milidetik, meskipun demikian ESP32-CAM juga memiliki keterbatasan seperti kapasitas memori dan daya pemrosesan yang terbatas, sehingga menjadi tantangan ketika digunakan untuk pengolahan citra yang intensif seperti deteksi objek berbasis *deep learning* [10].

Berbagai studi [11], [12], [13], [14] yang tersebar pada rentang waktu dan platform yang berbeda-beda tersebut belum dirangkum secara sistematis untuk memberikan gambaran komprehensif mengenai arsitektur model, teknik optimasi, serta kinerja sistem deteksi objek berbasis TinyML pada ESP32-CAM. Beberapa pertanyaan mendasar masih belum terjawab secara terintegrasi, misalnya sejauh mana *trade-off* antara ukuran model dan akurasi dapat ditoleransi pada aplikasi nyata, serta bagaimana kondisi lingkungan memengaruhi konsistensi kinerja sistem di lapangan dibandingkan saat pelatihan.

Oleh karena itu, penelitian ini menggunakan pendekatan *Systematic Literature Review* (SLR) untuk mensintesis temuan-temuan tersebut sehingga dapat memberikan peta jalan yang jelas bagi peneliti maupun praktisi yang ingin mengembangkan sistem serupa. Kontribusi utama penelitian ini adalah menyediakan rangkuman komparatif arsitektur model, teknik optimasi, dan metrik kinerja dari berbagai studi terkini, sekaligus mengidentifikasi celah penelitian yang dapat menjadi arah pengembangan pada masa mendatang.

II. METODOLOGI PENELITIAN

A. Desain Penelitian dan Pendekatan SLR

Penelitian ini menggunakan pendekatan SLR yang dilaksanakan melalui tiga tahap utama, yaitu tahap perencanaan (*planning*), tahap pelaksanaan (*conducting*), dan tahap pelaporan (*reporting*), mengikuti kerangka kerja yang umum digunakan dalam kajian sistematis di bidang teknik informatika dan rekayasa perangkat lunak. Tahap perencanaan meliputi perumusan pertanyaan penelitian (*research question*), penentuan strategi pencarian, serta penetapan kriteria inklusi dan eksklusi. Tahap pelaksanaan meliputi proses pencarian

literatur, penyaringan artikel, penilaian kualitas, serta ekstraksi data. Tahap pelaporan meliputi sintesis dan penyajian hasil kajian dalam bentuk naratif dan tabular.

Pertanyaan penelitian yang diajukan dalam kajian ini adalah: (1) arsitektur model TinyML apa yang paling banyak digunakan pada ESP32-CAM untuk deteksi objek; (2) teknik optimasi apa yang digunakan untuk menyesuaikan model dengan keterbatasan memori dan daya komputasi ESP32-CAM; dan (3) bagaimana kinerja sistem yang dihasilkan dari segi akurasi, ukuran model, dan latensi inferensi.

B. Strategi Pencarian dan Sumber Data

Pencarian literatur dilakukan pada basis data ilmiah bereputasi, meliputi Google Scholar, IEEE Xplore, Springer Link, ScienceDirect, MDPI, dan ResearchGate, dengan menggunakan kombinasi kata kunci seperti TinyML, ESP32-CAM, *object detection*, *edge AI*, dan *embedded vision*. Rentang waktu publikasi yang dipertimbangkan adalah tahun 2021 hingga 2026 untuk memastikan kebaruan dan relevansi teknologi yang dikaji, mengingat perkembangan platform TinyML yang sangat pesat dalam beberapa tahun terakhir.

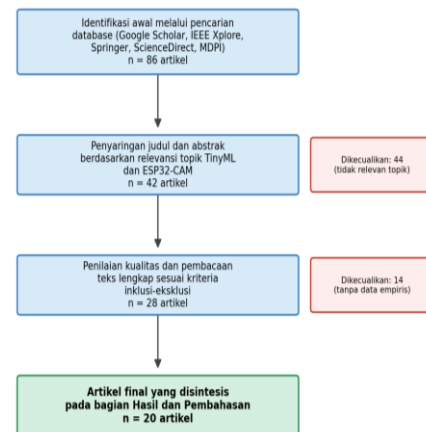
C. Kriteria Inklusi dan Eksklusi

Kriteria inklusi yang diterapkan meliputi: artikel membahas implementasi TinyML atau model pembelajaran mesin pada mikrokontroler berbasis kamera (khususnya ESP32-CAM); artikel melaporkan data kinerja kuantitatif berupa akurasi, ukuran model, atau latensi; serta artikel ditulis dalam bahasa Indonesia atau Inggris dan dipublikasikan pada jurnal, prosiding, atau repositori ilmiah yang dapat diakses penuh. Kriteria eksklusi meliputi artikel yang hanya membahas teori TinyML secara umum tanpa implementasi pada perangkat kamera, artikel duplikat, serta artikel yang tidak melaporkan hasil pengujian empiris.

D. Proses Seleksi dan Ekstraksi Data

Proses seleksi awal menghasilkan 86 artikel yang relevan dengan kata kunci pencarian. Setelah penyaringan judul dan abstrak, didapatkan 42 artikel yang berpotensi sesuai. Selanjutnya dilakukan penilaian kualitas dan pembacaan teks lengkap, yang menghasilkan 28 artikel yang lolos pada tahap tersebut, sebelum akhirnya disaring kembali berdasarkan kelengkapan data kinerja kuantitatif sehingga menghasilkan 20 artikel final yang memenuhi seluruh kriteria inklusi dan dijadikan dasar sintesis pada bagian hasil dan pembahasan. Data yang diekstraksi dari setiap artikel meliputi nama penulis, tahun publikasi, platform mikrokontroler yang digunakan, arsitektur model TinyML, teknik optimasi yang diterapkan, serta metrik kinerja yang dilaporkan seperti akurasi, ukuran model, dan waktu inferensi.

Tahapan seleksi literatur dalam penelitian ini diilustrasikan sebagai diagram alir pada Gambar 1, yang menggambarkan proses penyaringan dari 86 artikel awal menjadi 20 artikel akhir mengikuti prinsip dasar *Preferred Reporting Items for Systematic Reviews and Meta-Analyses* (PRISMA) yang diadaptasi untuk kajian pada bidang rekayasa sistem tertanam.



Gambar 1. Diagram Alir Seleksi Artikel SLR

Diagram alir seleksi disusun mengikuti prinsip PRISMA, mencakup tahap identifikasi, penyaringan, kelayakan, dan inklusi. Setiap tahap menyaring artikel berdasarkan relevansi judul, abstrak, dan isi penuh terhadap pertanyaan penelitian yang telah ditetapkan, sehingga memastikan transparansi dan keterulangan (*reproducibility*) dari proses seleksi yang dilakukan.

III. HASIL DAN PEMBAHASAN

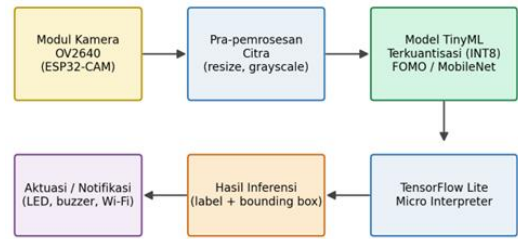
A. Arsitektur Model TinyML yang Digunakan

Hasil sintesis terhadap 20 artikel terpilih menunjukkan bahwa arsitektur FOMO merupakan pendekatan yang paling banyak diadopsi untuk deteksi objek pada ESP32-CAM, khususnya pada konteks robotika dan kontes bersifat kompetitif. Hal ini sejalan dengan temuan bahwa FOMO dirancang khusus agar mampu berjalan secara efisien pada perangkat dengan sumber daya terbatas seperti ESP32-CAM, dan terbukti efektif pada studi deteksi boneka korban dengan tingkat keberhasilan model yang tinggi pada proses pelatihan [7]. Pembahasan berikut menyoroti studi-studi yang paling representatif dari korpus 20 artikel yang disintesis; pola arsitektur, teknik optimasi, dan kecenderungan kinerja yang serupa juga teridentifikasi secara konsisten pada artikel-artikel lain dalam korpus tersebut, meskipun tidak seluruhnya diuraikan satu per satu agar pembahasan tetap ringkas.

Selain FOMO, sejumlah penelitian juga mengadopsi model berbasis *MobileNet* yang dikuantisasi menggunakan *TensorFlow Lite for Microcontrollers* (TFLM), terutama untuk tugas deteksi keberadaan manusia. Salah satu studi menerapkan arsitektur MobileNetV2 terkuantisasi ($\alpha = 0,1$) pada ESP32-CAM yang diintegrasikan dengan protokol MQTT untuk pengiriman notifikasi deteksi secara langsung, dengan kebutuhan RAM puncak sebesar 485,4 KB dan memori flash sebesar 102,1 KB, meskipun rata-rata waktu inferensinya mencapai 1018 milidetik per bingkai sehingga hanya sesuai untuk pengawasan dasar pada laju sekitar satu bingkai per detik [8].

Pada studi lain dengan kompleksitas objek yang lebih tinggi seperti deteksi kerusakan permukaan jalan, arsitektur YOLO versi terbaru juga mulai diadaptasi pada ESP32-CAM meskipun harus melalui proses kompresi model yang

signifikan, mengingat keterbatasan kapasitas memori dan daya pemrosesan menjadi tantangan tersendiri ketika digunakan untuk tugas pengolahan citra yang intensif berbasis *deep learning* [10]. Gambar 2 menunjukkan arsitektur umum sistem TinyML pada ESP32-CAM yang disintesis dari berbagai studi yang dikaji, mulai dari akuisisi citra hingga aktuasi keluaran.



Gambar 2. Arsitektur Sistem TinyML pada ESP32-CAM

Sebagaimana terlihat pada Gambar 2, alur pemrosesan dimulai dari modul kamera OV2640 yang terintegrasi pada ESP32-CAM, dilanjutkan dengan pra-pemrosesan citra seperti pengubahan ukuran (*resize*) dan konversi ke skala abu-abu untuk mengurangi beban komputasi, kemudian dilakukan inferensi menggunakan model TinyML yang telah dikuantisasi melalui interpreter *TensorFlow Lite Micro*, sebelum akhirnya hasil deteksi digunakan untuk memicu aktuasi atau notifikasi sesuai kebutuhan aplikasi.

B. Teknik Optimasi Model

Teknik optimasi yang paling konsisten ditemukan pada literatur yang dikaji adalah kuantisasi INT8 yang mengubah representasi numerik model dari *floating point* menjadi bilangan bulat 8-bit, sehingga ukuran model dan kebutuhan komputasi dapat ditekan secara signifikan tanpa kehilangan akurasi yang berarti. Studi penerapan pengenalan suara pada ESP32 melaporkan bahwa proses kompilasi dan kuantisasi model menghasilkan penurunan penggunaan RAM sebesar 37% dan ROM sebesar 27% [15] dibandingkan model sebelum dioptimasi.

Teknik lain yang turut diidentifikasi dalam kajian ini adalah pemangkasan jaringan (*pruning*) dan penggunaan *compiler* khusus seperti *EON Compiler* pada platform *Edge Impulse*, yang secara konsisten digunakan dalam berbagai studi untuk mengoptimalkan model sebelum di-deploy ke perangkat keras dengan sumber daya terbatas. Pendekatan tropikalisasi data turut ditemukan pada riset lokal, di mana model *anomaly detection* berbasis *Edge AI* yang diuji secara simulasi mencapai akurasi deteksi anomali sebesar 89,4% dengan ukuran model 142 KB dan rata-rata latensi inferensi sekitar 28 milidetik [16] setelah melalui proses kuantisasi INT8 dan augmentasi *noise* tropis.

C. Kinerja Sistem: Akurasi, Ukuran Model, dan Latensi

Sintesis data kinerja dari seluruh artikel yang dikaji menunjukkan variasi akurasi deteksi objek yang cukup tinggi, bergantung pada kompleksitas objek dan kondisi lingkungan pengujian. Pada studi deteksi objek SAR, model yang dilatih mencapai *F1-Score* pengujian training sebesar 100%, namun pada pengujian lapangan langsung dari 10 kali percobaan,

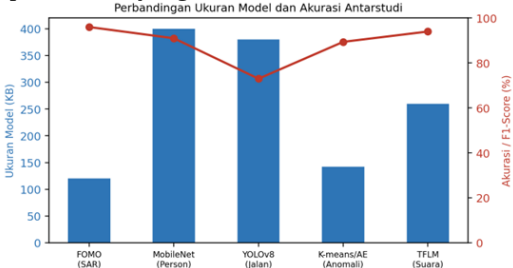
model berhasil mendeteksi boneka *dummy* dengan benar sebanyak 8 kali dan boneka korban sebanyak 9 kali [7], menunjukkan adanya penurunan kinerja saat diuji pada kondisi nyata dibandingkan kondisi pelatihan.

Perbandingan antara berbagai studi terkait ukuran model, teknik optimasi, dan kinerja yang dilaporkan dirangkum pada Tabel 1.

Tabel 1. Ringkasan Temuan SLR

Fokus Studi	Arsitektur	Optimasi	Kinerja
Deteksi boneka korban (SAR)	FOMO	Kuantisasi platform	F1 <i>training</i> 100%
Deteksi keberadaan manusia	MobileNet	Kuantisasi INT8	Ukuran ±400 KB
Deteksi kerusakan jalan	YOLOv8	Kompresi model	F1 0,73; 300-350 ms
Deteksi anomali sensor	K-means/AE	INT8 + EON	Akurasi 89,4%; 28 ms
Pengenalan suara (MFCC)	TFLM	EON <i>Compiler</i>	RAM -37%, ROM -27%

Latensi inferensi menjadi salah satu metrik yang paling bervariasi antarstudi, dengan rentang dari puluhan milidetik pada model klasifikasi sederhana hingga ratusan milidetik pada model deteksi objek visual yang lebih kompleks. Hal ini konsisten dengan karakteristik kerangka kerja TFLM yang mengutamakan overhead runtime minimal melalui pendekatan interpreter ringan [9], namun latensi aktual di lapangan tetap sangat bergantung pada kompleksitas arsitektur model dan resolusi citra yang diproses, sebagaimana ditunjukkan pada Tabel 1, sehingga performa *real-time* pada tugas deteksi objek visual yang lebih kompleks masih memerlukan optimasi lanjutan. Gambar 3 menunjukkan perbandingan ukuran model dan akurasi/F1-score pada lima kelompok studi yang representatif terhadap variasi pendekatan yang ditemukan.



Gambar 3. Perbandingan Ukuran Model dan Akurasi

Sebagaimana terlihat pada Gambar 3, tidak terdapat korelasi linear sederhana antara ukuran model dan tingkat akurasi yang dicapai. Model FOMO dengan ukuran terkecil justru mencapai akurasi pelatihan yang tinggi, sementara model YOLOv8 yang berukuran lebih besar mencatatkan F1-

Score yang relatif lebih rendah pada kondisi pengujian lapangan, mengindikasikan bahwa kompleksitas objek dan kondisi lingkungan pengujian memberikan pengaruh yang lebih besar dibandingkan ukuran model semata.

Perbedaan capaian kinerja antarstudi tersebut tidak dapat dipahami hanya dari angka yang dilaporkan, melainkan perlu ditelusuri penyebabnya. Pertama, kompleksitas tugas deteksi berperan besar: model FOMO pada studi deteksi boneka SAR menghadapi klasifikasi biner dengan latar belakang yang relatif terkendali, sedangkan model YOLOv8 pada deteksi kerusakan jalan harus menangani variasi bentuk, ukuran, dan pencahayaan objek yang jauh lebih beragam di lingkungan terbuka, sehingga F1-Score yang dicapai relatif lebih rendah meskipun arsitekturnya lebih kompleks. Kedua, kombinasi teknik optimasi yang diterapkan tidak seragam antarstudi; sebagian hanya mengandalkan kuantisasi INT8, sementara studi lain menggabungkannya dengan pruning atau *compiler* khusus seperti EON *Compiler*, sehingga efisiensi memori yang dihasilkan tidak selalu dapat dibandingkan secara langsung. Ketiga, kondisi pengujian turut menentukan hasil yang dilaporkan; studi yang diuji secara simulasi cenderung melaporkan latensi dan akurasi yang lebih baik dibandingkan studi yang diuji langsung pada perangkat keras di lapangan, sebagaimana terlihat pada kesenjangan antara F1-Score pelatihan sebesar 100% dan hasil pengujian lapangan pada studi deteksi boneka SAR. Dengan demikian, variasi hasil antarstudi lebih mencerminkan perbedaan kompleksitas tugas, kombinasi teknik optimasi, dan skenario pengujian yang digunakan, dibandingkan sekadar keunggulan satu arsitektur atas arsitektur lainnya.

Untuk memberikan gambaran lebih rinci mengenai kebutuhan sumber daya perangkat keras, Tabel 2 merangkum perbandingan kebutuhan memori (RAM dan *Flash*) dari lima studi yang dikaji beserta platform mikrokontroler yang digunakan.

Tabel 2. Perbandingan Kebutuhan Sumber Daya

Studi	Mikrokontroler	RAM (KB)	Flash (KB)
Deteksi boneka korban (SAR)	ESP32-CAM	85	120
Deteksi keberadaan manusia	ESP32-CAM	100	300
Deteksi kerusakan jalan	ESP32-CAM	180	380
Deteksi anomali sensor	Simulasi PC	32	142
Pengenalan suara (MFCC)	ESP32	63	190

D. Tantangan dan Peluang Penelitian Lanjutan

Kajian terhadap literatur yang ada juga mengidentifikasi sejumlah tantangan yang konsisten muncul, yaitu sensitivitas model terhadap variasi pencahayaan dan latar belakang, keterbatasan jumlah data pelatihan pada kondisi lapangan

yang beragam, serta *trade-off* antara akurasi dan efisiensi sumber daya. Studi mengenai analisis kinerja ESP32-CAM dalam deteksi objek menegaskan bahwa keberhasilan deteksi sangat dipengaruhi oleh pengaturan lingkungan dan karakteristik objek [10], sehingga rekomendasi teknis terkait pengaturan lingkungan dan pemilihan model menjadi penting untuk penelitian selanjutnya.

Peluang penelitian lanjutan yang dapat diidentifikasi dari kajian ini antara lain pengembangan teknik adaptasi domain agar model lebih *robust* terhadap variasi pencahayaan, integrasi *federated learning* untuk pembaruan model tanpa mengirim data mentah ke *cloud*, serta eksplorasi arsitektur jaringan saraf yang lebih ringan namun tetap mempertahankan akurasi tinggi untuk tugas deteksi objek multi-kelas. Selain itu, kombinasi TinyML dengan teknik sensor *fusion*, misalnya memadukan data visual dengan sensor inframerah atau radar jarak pendek, berpotensi meningkatkan keandalan deteksi pada kondisi pencahayaan rendah yang menjadi kelemahan utama sistem berbasis kamera tunggal.

Tabel 3 merangkum tantangan utama dan rekomendasi penelitian lanjutan yang teridentifikasi dari hasil sintesis literatur.

Tabel 3. Tantangan dan Rekomendasi Penelitian Lanjutan

Tantangan	Rekomendasi	Prioritas
Sensitivitas pencahayaan	Adaptasi domain & augmentasi	Tinggi
Latensi inferensi tinggi	<i>Pruning</i> & arsitektur ringan	Tinggi
Data lapangan terbatas	Pengumpulan data multi-lokasi	Sedang
Update model di <i>edge</i>	<i>Federated learning</i>	Sedang

E. Perbandingan dengan Tren Riset Internasional

Hasil sintesis literatur nasional dan internasional dalam kajian ini menunjukkan kesamaan arah perkembangan, yaitu pergeseran dari TinyML berbasis klasifikasi sederhana menuju *Tiny Deep Learning* (TinyDL) yang mampu menjalankan model *deep learning* yang lebih kompleks pada perangkat dengan keterbatasan sumber daya. Survei skala besar mengenai transisi dari TinyML ke TinyDL melaporkan bahwa perkembangan ini mencakup inovasi arsitektur, platform perangkat keras, teknik optimasi model, dan *toolchain* perangkat lunak yang semakin matang, mencerminkan pergeseran paradigma dari sekadar inferensi sederhana menuju kemampuan pemrosesan yang mendekati kapabilitas *deep learning* konvensional pada perangkat tepi [17].

Tren pasar juga turut mendukung arah perkembangan ini, di mana proyeksi pertumbuhan pasar TinyML serta meningkatnya jumlah perangkat IoT yang terhubung menjadi pendorong utama investasi riset pada arsitektur model yang semakin efisien. Hal ini relevan dengan temuan dalam kajian ini bahwa ESP32-CAM, sebagai salah satu platform berbiaya rendah yang paling banyak diteliti, terus menjadi subjek eksperimen berbagai teknik kompresi dan optimasi model

terbaru, mengindikasikan bahwa platform ini akan tetap relevan untuk riset TinyML dalam beberapa tahun mendatang.

F. Implikasi Praktis bagi Pengembang Sistem

Bagi pengembang sistem yang ingin mengimplementasikan TinyML pada ESP32-CAM, hasil kajian ini memberikan beberapa implikasi praktis. Pertama, pemilihan arsitektur model perlu disesuaikan dengan kompleksitas tugas deteksi; untuk tugas deteksi objek tunggal dengan latar belakang relatif statis, arsitektur FOMO menawarkan keseimbangan terbaik antara ukuran model dan akurasi. Kedua, untuk tugas yang melibatkan variasi objek lebih kompleks seperti deteksi kerusakan infrastruktur, diperlukan kompromi antara penggunaan arsitektur yang lebih besar seperti YOLO dengan konsekuensi latensi yang lebih tinggi.

Ketiga, proses kuantisasi INT8 sebaiknya menjadi langkah standar dalam alur kerja pengembangan, mengingat konsistensi manfaatnya dalam menekan kebutuhan memori pada seluruh studi yang dikaji tanpa pengorbanan akurasi yang berarti. Keempat, pengujian pada kondisi lapangan yang merepresentasikan variasi pencahayaan, jarak, dan sudut pandang nyata perlu dilakukan secara menyeluruh sebelum sistem diterapkan secara operasional, mengingat kesenjangan kinerja antara hasil pelatihan dan pengujian lapangan yang konsisten ditemukan pada berbagai studi yang dikaji dalam penelitian ini.

Secara keseluruhan, sintesis dari berbagai studi menunjukkan bahwa keberhasilan implementasi TinyML pada ESP32-CAM tidak hanya ditentukan oleh pemilihan arsitektur model semata, melainkan juga oleh kombinasi yang tepat antara teknik optimasi, kualitas data pelatihan, dan kesesuaian kondisi pengujian dengan skenario penerapan di lapangan. Pendekatan yang holistik dengan memperhatikan ketiga aspek tersebut secara bersamaan akan menghasilkan sistem deteksi objek yang lebih andal dan siap untuk diterapkan pada skala produksi, dibandingkan dengan pendekatan yang hanya berfokus pada satu aspek optimasi saja.

IV. KESIMPULAN

Kajian sistematis terhadap 20 artikel yang membahas penerapan TinyML pada ESP32-CAM untuk deteksi objek *real-time* menunjukkan bahwa arsitektur FOMO dan model berbasis *MobileNet* yang dikuantisasi menjadi pendekatan paling dominan karena kesesuaiannya dengan keterbatasan memori dan daya komputasi perangkat. Teknik optimasi seperti kuantisasi INT8, pemangkasan jaringan, dan penggunaan *compiler* khusus seperti *EON Compiler* terbukti secara konsisten mampu menekan ukuran model dan kebutuhan memori tanpa mengorbankan akurasi secara signifikan.

Meskipun demikian, kinerja sistem pada kondisi pengujian lapangan masih menunjukkan penurunan dibandingkan kondisi pelatihan, terutama dipengaruhi oleh variasi pencahayaan dan kompleksitas objek. Penelitian mendatang disarankan untuk berfokus pada peningkatan robustitas

model terhadap kondisi lingkungan nyata serta eksplorasi teknik pembelajaran adaptif yang tetap sesuai dengan keterbatasan sumber daya pada mikrokontroler ESP32-CAM, sehingga sistem deteksi objek berbiaya rendah ini dapat diterapkan secara lebih luas pada aplikasi pemantauan, robotika, dan keamanan di lapangan.

Berbeda dari artikel-artikel sumber yang dikaji secara terpisah, kontribusi utama kajian ini terletak pada penyusunan peta jalan komparatif yang memetakan hubungan antara pilihan arsitektur, kombinasi teknik optimasi, dan kondisi pengujian terhadap kinerja yang dicapai, sebagaimana dirangkum pada Tabel 1 hingga Tabel 3, sehingga peneliti dan praktisi dapat memilih kombinasi pendekatan yang sesuai dengan kebutuhan aplikasinya tanpa perlu menelusuri satu per satu literatur yang tersebar.

REFERENSI

- [1] M. Ardi and E. T. Putri, "Sistem Pendukung Keputusan Pemeriksaan Kesehatan Menggunakan Decision Tree Classifier," *Konvergensi J. Teknol. Inf. dan Komun.*, vol. 21, no. 2, pp. 47–54, 2025.
- [2] M. Ardi and E. D. Bintari, "Systematic literature review: risiko privasi dan keamanan data pribadi dalam penggunaan artificial intelligence (AI)," *Inf. Interaktif J. Inform. dan Teknol. Inf.*, vol. 9, no. 1, pp. 23–28, Jan. 2024, doi: 10.37159/jii.v9i1.64.
- [3] M. Ardi and K. Krisdyanda, "Perancangan Aplikasi Penjualan Senapan Online dengan Metode Prototype di Toko Rayap Sport," *J. Ilm. Komputasi*, vol. 24, no. 4, pp. 595–604, Dec. 2025, doi: 10.32409/jikstik.24.4.3909.
- [4] R. Fitriyanto and M. Ardi, "Feature Selection Comparative Performance for Unsupervised Learning on Categorical Dataset," *J. Techno Nusa Mandiri*, vol. 22, no. 1, pp. 61–69, Mar. 2025, doi: 10.33480/techno.v22i1.6512.
- [5] M. Ardi, "Data Analytics," in *Teknologi Big Data : Pengantar dan Penerapan Teknologi Big Data di berbagai Bidang*, 1st ed., E. Rianty, Ed., Yogyakarta: PT. Green Pustaka Indonesia, 2024, ch. 5, pp. 53–67.
- [6] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A Comprehensive Survey on TinyML," *IEEE Access*, vol. 11, no. July, pp. 96892–96922, 2023, doi: 10.1109/ACCESS.2023.3294111.
- [7] A. Taupiq, Y. Pratama, and M. I. Bustami, "Deteksi Objek Boneka Korban pada Kontes Robot SAR Indonesia Menggunakan ESP32-cam," *Build. Informatics, Technol. Sci.*, vol. 6, no. 3, pp. 2073–2084, Dec. 2024, doi: 10.47065/bits.v6i3.5979.
- [8] A. Purnama, Indriani, B. A. Prasetyo, and A. Nugraha, "Edge-Based Person Detection Using MobileNetV2 on ESP32-CAM for Home Surveillance System," *J. Sist. Komput. dan Inform.*, vol. 7, no. 3, pp. 1171–1182, Mar. 2026, doi: 10.30865/json.v7i3.9619.
- [9] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang, P. Warden, and R. Rhodes, "TensorFlow Lite Micro: Embedded Machine Learning for TinyML Systems," in *Proc. Mach. Learn. Syst.*, vol. 3, 2021, pp. 800–811.
- [10] A. Herwandi, A. A. Ramadhan, N. T. Sunggono, and F. Ferawati, "Analisis Kinerja ESP32-CAM Dalam Mendeteksi Objek," *Bit-Tech*, vol. 7, no. 3, pp. 1014–1021, Apr. 2025, doi: 10.32877/bt.v7i3.2296.
- [11] C. M. Bhushan *et al.*, "Deploying TinyML for energy-efficient object detection and communication in low-power edge AI systems," *Sci. Rep.*, vol. 15, no. 1, Dec. 2025, doi: 10.1038/s41598-025-27818-9.

- [12] Y.-H. Chang, F.-C. Wu, and H.-W. Lin, "Design and Implementation of ESP32-Based Edge Computing for Object Detection," *Sensors*, vol. 25, no. 6, Mar. 2025, doi: 10.3390/s25061656.
- [13] V. Tsoukas, A. Gkogkidis, E. Boumpa, and A. Kakarountas, "A Review on the emerging technology of TinyML," *ACM Comput. Surv.*, vol. 56, no. 10, pp. 1–37, Oct. 2024, doi: 10.1145/3661820.
- [14] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 4, pp. 1595–1623, Apr. 2022, doi: 10.1016/j.jksuci.2021.11.019.
- [15] M. González, S. Gutiérrez, R. Espinosa, and H. Ponce, "Deploying Real-Time Speech Recognition on ESP32 Using TinyML and Edge Impulse," in *Artificial Intelligence – COMIA 2025*, Cham, Switzerland: Springer, 2025, pp. 211–230, doi: https://doi.org/10.1007/978-3-031-97907-1_17.
- [16] Iqbal and Hasriadi, "Implementasi TinyML dengan Edge AI untuk Deteksi Anomali Sensor IoT pada Kondisi Lingkungan Tropis," *J. Ilm. Sist. Inf. dan Tek. Inform.*, vol. 9, no. 1, pp. 76–82, 2026, doi: <https://doi.org/10.57093/jisti.v9i1.402>.
- [17] S. Somvanshi *et al.*, "From Tiny Machine Learning to Tiny Deep Learning: A Survey," *ACM Comput. Surv.*, vol. 58, no. 7, pp. 1–33, May 2025, doi: 10.1145/3776588.