

# Lembar Pengesahan

Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Terapan Teknik (S.Tr.T)  
di  
Politeknik Negeri Batam

Oleh:  
Zakia Putri (4242101025)

Tanggal Sidang: 16, 05, 2025

**Pembimbing I**



**Abdullah Sani, S.ST., M.Sc.**  
NIK:113119

**Penguji I**



**Yusiran, S.Si, M.T.**  
NIK:123294

**Penguji II**



**Sumantri Kurniawan  
Risandria, S.T., M.T.**  
NIK: 104030

## Pernyataan Keaslian Tugas Akhir

Saya yang bertandatangan di bawah ini menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya yang berjudul : *"Implementasi Pengenalan Kucing Dengan ESP32-CAM Menggunakan Platform Edge Impulse Untuk Akses Pintu Masuk Pada Kandang Kucing"* adalah hasil karya sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diizinkan, dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri. Semua referensi yang dikutip atau dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan saya ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Batam, 09 Mei 2025



Zakia Putri

NIM: 4242101025

# Implementasi Pengenalan Kucing Dengan ESP32-CAM Menggunakan Metode Edge Impulse Untuk Akses Pintu Masuk Pada Kandang Kucing

Zakia Putri<sup>1</sup>, Abdullah Sani<sup>2</sup>  
Teknik Elektro  
Politeknik Negeri Batam  
Batam, Indonesia  
[zakiaputri530@gmail.com](mailto:zakiaputri530@gmail.com)  
[sani@polibatam.ac.id](mailto:sani@polibatam.ac.id)

*Abstract*—Pengenalan objek merupakan salah satu tugas penting dalam bidang visi komputer yang bertujuan untuk mengidentifikasi dan menentukan lokasi objek dalam citra atau video. Penelitian ini akan mengimplementasikan sistem pengenalan kucing menggunakan ESP32-CAM dan platform Edge Impulse untuk mengendalikan akses pintu masuk pada kandang kucing. ESP32-CAM adalah modul kamera yang terintegrasi dengan mikrokontroler, sementara Edge Impulse adalah platform yang memungkinkan pengembangan model pembelajaran mesin pada perangkat edge. Platform ini memungkinkan pintu kandang untuk membuka secara otomatis ketika kucing terdeteksi, sehingga meningkatkan kenyamanan dan keamanan bagi kucing serta memudahkan pemilik dalam merawat hewan peliharaan mereka. Pengujian dilakukan untuk menganalisis kinerja sistem dalam kondisi pencahayaan dan memastikan keakuratan pengenalan kucing. Hasil yang diharapkan adalah sistem yang dapat membedakan kucing pemilik dari objek lain, serta memberikan akses pintu kandang dengan sistem yang akurat.

Kata kunci—Pengenalan Kucing, Edge Impulse, ESP32-CAM

## PENDAHULUAN

Seiring dengan banyaknya minat masyarakat terhadap hewan peliharaan terutama kucing, permintaan terhadap teknologi otomatis yang dapat memudahkan pecinta kucing dalam merawat kucing pun semakin banyak terutama pada pet shop dan dokter hewan. Hal ini dikarenakan beberapa kesulitan yang dikeluhkan oleh para pecinta kucing dalam melakukan penjagaan dan perawatan terhadap kucing peliharaannya[1].

Penelitian menunjukkan bahwa memiliki hewan peliharaan, termasuk kucing, dapat mengurangi tingkat stres pemilik hingga 37,2%[2]. Mereka yang memelihara hewan peliharaan memiliki tingkat stres yang lebih rendah dibandingkan dengan mereka yang tidak memelihara hewan.

Kesehatan kucing peliharaan sangat dipengaruhi oleh perawatan yang diberikan oleh pemiliknya[3]. Masalah timbul ketika pemilik tidak dapat memberikan perawatan yang memadai karena cuaca hujan dan kesibukan kerja, yang mengakibatkan kucing harus berlindung di rumah kosong yang lembap atau bahkan kehujanan.

Kucing memerlukan beberapa fasilitas untuk memenuhi kebutuhan mereka, termasuk kandang untuk tempat beristirahat[4]. Tanpa fasilitas ini, kucing cenderung

melakukan aktivitasnya di tempat sembarangan karena naluri alaminya yang kuat. Namun, seringkali ditemui pandangan bahwa keberadaan fasilitas ini dapat mengganggu estetika ruangan jika diletakkan di dalam rumah.

Untuk mengatasi masalah tersebut, diperlukan teknologi yang dapat mengontrol akses pintu masuk kandang kucing dengan menggunakan ESP32-CAM dan platform Edge Impulse. Konsep ini melibatkan pendeteksian kucing secara visual melalui ESP32-CAM. Setelah kucing terdeteksi, pintu akan memberikan akses masuk berdasarkan data yang telah diinput. Hal ini berguna ketika pemilik sedang berada diluar rumah. Teknologi ini sangat berguna ketika pemilik sedang pergi dari rumah[5]. Jika cuaca buruk, kucing dapat beristirahat dengan nyaman karena pintu kandang akan terbuka secara otomatis menggunakan platform Edge Impulse ketika kucing pemilik terdeteksi oleh ESP32-CAM. Dengan demikian, pemilik tidak perlu pulang ke rumah untuk memanggil kucing untuk masuk ke kandang.

## I. TINJAUAN PUSTAKA

### A. Penelitian Terdahulu

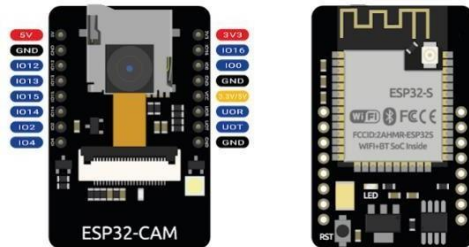
Penelitian ini dilakukan dengan mempertimbangkan hasil penelitianpenelitian terdahulu sebagai bahan perbandingan dan kajian. Hasil-hasil penelitian yang dijadikan referensi terkait dengan topik penelitian tentang teknologi otomatis untuk kontrol akses masuk pada kandang kucing agar dapat masuk ke kandangnya secara otomatis.

Berdasarkan hasil penelitian yang dilakukan oleh [6]menulis tentang Alat Kandang Kucing Otomatis Berbasis Mikrokontroler dengan Monitoring Telegram, ditemukan bahwa kucing yang mengenakan kalung dengan modul ESP 01 dapat mengirimkan sinyal ke server di kandang. Ketika kucing berada dalam jarak 20 cm hingga 50 cm dari pintu kandang, sensor ultrasonik akan mendeteksi keberadaan objek dan memberikan sinyal kepada motor servo untuk membuka pintu.

Berdasarkan hasil kajian dari[5] menulis tentang Prototipe Sistem Kendali Jarak Jauh Pada Pakan Dan Pintu Kandang Kucing, Sistem ini mengintegrasikan berbagai komponen, seperti motor servo, sensor infrared, NodeMCU, dan aplikasi Blynk, untuk memungkinkan pengendalian pintu kandang dan pemberian pakan secara jarak jauh.

## B. ESP32CAM

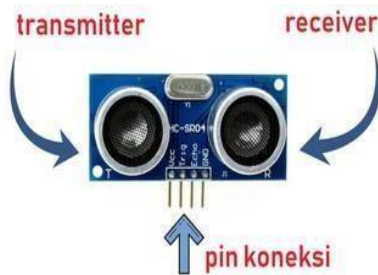
ESP32 Cam adalah mode komunikasi nirkabel berbasis antenna yang didasarkan pada Bluetooth dan WI-FI, dan itu adalah PCB berdasarkan mikroprosesor ESP32. Modul ini dapat digunakan untuk berbagai tujuan, seperti memantau CCTV, menampilkan grafis, dan sebagainya. Dengan cara ini, modul ESP32 Cam dapat digunakan untuk mengirimkan video dan juga dapat digunakan sebagai modul wifi untuk menerima data[7].



Gambar 1 ESP32CAM

## C. Sensor Ultrasonik

Sensor ultrasonik adalah jenis sensor yang mengubah sinyal fisik (bunyi) menjadi sinyal digital. Alat ini akan menghasilkan suara ultrasonik[8] dengan frekuensi 40 kHz ketika osilator ditempatkan pada benda yang akan diuji. Sensor ultrasonik umumnya digunakan untuk banyak jenis pengujian non-pembusukan, seperti aplikasi inspeksi sarung tangan. Secara umum, perangkat ini menampilkan sidik jari ultrasonic pada sasaran yang menunjukkan adanya keterlambatan sensor. Sistem kemudian menyesuaikan waktu yang diperlukan untuk pencarian gelombang untuk kembali ke sensor dan mendeteksi sasaran dengan menggunakan kecepatan permukaan media. Selain itu, generator ultrasonik didukung oleh jenis kristal dengan sifat piezoelektrik.

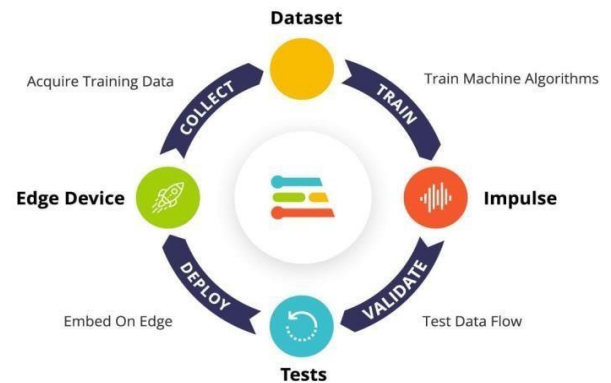


Gambar 2 Sensor Ultrasonik

## D. Edge Impulse

Platform operasi pembelajaran mesin berbasis cloud (MLOps) yang disebut Edge Impulse digunakan untuk mengembangkan sistem embedded dan edge ML (TinyML) yang dapat didistribusikan ke banyak target perangkat keras. Gaya kerja TinyML saat ini ditandai dengan perangkat lunak yang terfragmentasi dan penyebaran keras optimasi heterogen, yang menghasilkan model pembelajaran mesin yang terbatas dan tidak memuaskan. Kami memperkenalkan Edge Impulse, platform latihan MLOps yang dirancang untuk skala sistem TinyML. Edge Impulse mengatasi

tantangan ini dan meningkatkan desain stack TinyML dengan mengintegrasikan beberapa permutasi lunak dan keras untuk menciptakan stack keras yang fleksibel dan cocok untuk berbagai sistem heterogen. Edge Impulse melaporkan 118.185 proyek dari 50.953 start-up pada bulan Oktober 2022[9].



Gambar 3 Platform Edge Impulse

## E. FOMO (Fast Object More Object)

*Faster Object, More Objects (FOMO)* adalah platform deteksi objek yang diperkenalkan oleh Edge Impulse dan disesuaikan untuk digunakan oleh Edge edge yang sangat tahan lama. Untuk memulai, kurang dari 200K RAM diperlukan, yang setara dengan 30 kali lebih banyak RAM daripada SSD. Ini menggunakan teknik *Background Subtraction (BGS)* untuk menyesuaikan kontras antara objek latar depan dan latar belakang. Asumsi ini adalah bahwa objek ke bawah akan naik sementara objek ke atas akan jatuh. Bahkan jika platform saraf yang lebih baru menghasilkan hasil yang lebih akurat, mereka masih membutuhkan lebih banyak waktu komputer. Pemilihan platform BGS tergantung pada daya komputer yang tersedia di lokasi tepi. Ini membuatnya menjadi alat yang sangat berguna untuk perangkat edge[10], model FOMO (Faster Objects, More Objects) tersedia dalam dua versi, yaitu dengan nilai alpha 0.1 dan alpha 0.35. Model FOMO dengan alpha 0.1 memiliki ukuran model yang lebih kecil, yaitu kurang dari 100kB, sehingga cocok digunakan pada perangkat dengan sumber daya terbatas. Meskipun demikian, model ini memiliki akurasi yang lebih rendah dan lebih sesuai untuk mendeteksi objek yang berukuran besar dan memiliki bentuk yang jelas. Model FOMO dengan alpha 0.35 memiliki ukuran yang lebih besar, yaitu diatas 100kB. Model tersebut menawarkan akurasi yang lebih tinggi karena memiliki kemampuan untuk mendeteksi objek yang lebih kecil.

## F. Tiny ML

Tiny Machine Learning (TinyML) adalah bidang penelitian yang berkembang pesat yang berfokus pada penerapan teknik pembelajaran mesin ke sistem kecil dan menengah. TinyML menggunakan Unit MikroKontroler (MCU) yang beroperasi pada daya rendah (1 mW), memiliki memori variabel, dan logika diskret. Dengan mengumpulkan data secara lokal, TinyML mengurangi konsumsi energi dibandingkan komunikasi, memperpanjang umur baterai perangkat IoT, dan menggunakan teknologi komunikasi berdaya rendah seperti LPWAN. Perangkat inferensi ML juga meningkatkan keamanan data dan privasi pengguna karena data tidak perlu disimpan secara online[11].

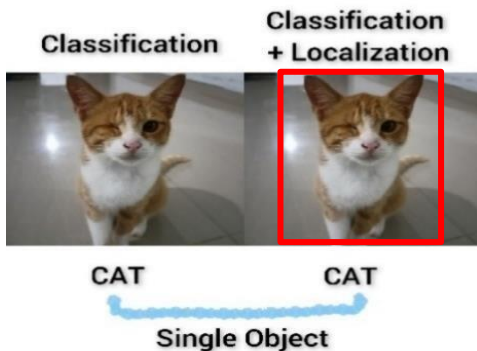
### G. . Quantization

Kuantisasi adalah proses mengubah nilai kontinu atau analog menjadi nilai diskrit atau digital. Dalam konteks pembelajaran mesin dan pemrosesan sinyal, kuantisasi biasanya mengacu pada mengurangi presisi model atau data agar lebih efisien dijalankan pada perangkat dengan sumber daya terbatas, seperti mikroprosesor atau MCU. Misalnya, model neural network yang awalnya menggunakan angka floating-point 32-bit dapat dikuantisasi menjadi angka fixed-point 8-bit. Meskipun ini bisa sedikit menurunkan akurasi model, kuantisasi memungkinkan model berjalan lebih efisien pada perangkat keras yang memiliki keterbatasan dalam hal daya, memori, dan komputasi[12].

### H. Object Detection

Objek deteksi merupakan tugas penting dalam bidang visi komputer yang bertujuan untuk mengenali dan menentukan posisi objek dalam gambar atau video. Tujuan deteksi objek adalah untuk mendeteksi objek apa pun dari satu atau lebih kategori yang diketahui, seperti orang, ponsel, atau wanita dalam gambar. Secara umum, hanya ada beberapa objek kecil yang digambarkan dalam gambar namun, ada banyak lokasi dan ukuran besar yang dapat muncul dan perlu diselidiki[13].

Setiap deteksi disertai dengan beberapa jenis informasi pose. Ini bisa sederhana lokasi objek, lokasi dan skala, kotak batas, atau topeng segmentasi. Dalam skenario lain, informasi tersebut lebih rinci dan memiliki parameter transformasi linear atau non-linear. Misalnya, detektor wajah mungkin mendeteksi lokasi mata, hidung, dan mulut selain kontur wajah[14].



Gambar 4 Objek Deteksi

### I. Arsitektur Edge Impulse

Pada gambar 5 menunjukkan arsitektur platform Edge Impulse , yang mendukung proses kerja menyeluruh untuk pengembangan TinyML. Prosesnya dimulai dari akuisisi data, yaitu pengambilan data sensor secara otomatis dari perangkat lalu pra-pemrosesan dan interpretasi, kemudian pelatihan model machine learning di cloud. Setelah model siap, Edge Impulse memudahkan deploy model ke edge ( perangkat dengan sumber daya terbatas ). Selain itu, platform ini menyediakan antarmuka visual dan integrasi langsung dengan beragam perangkat keras, sehingga eksperimen dapat dilakukan dengan cepat dan efisien[15].

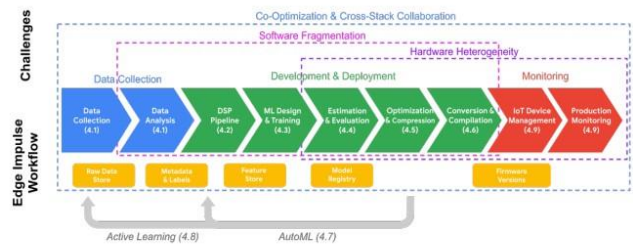


Figure 1. The challenges associated with the ML Workflow and features of Edge Impulse that solve those challenges.

Gambar 5. Arsitektur Edge Impulse

### J. Alur Kerja Platform Edge Impulse

Pada gambar 6 di bawah menunjukkan alur kerja dari TinyML

#### 1. Collect Images ( Mengumpulkan Gambar )

Tahapan pertama adalah mengumpulkan data berupa gambar lingkungan sekitar sesuai dengan gambar apa yang akan di teliti menggunakan sensor seperti kamera. Data ini sangat penting sebagai bahan dasar pembelajaran bagi model machine learning yang akan dikembangkan

#### 2. Label Images ( Memberi Label Gambar )

Setelah data dikumpulkan, setiap gambar harus diberi label sesuai dengan kategori yang diinginkan. Misalnya, gambar kucing1 diberi label “kucing1”, dan gambar kucing2 diberi label “Kucing2”. Pemberian label ini bertujuan agar model dapat belajar membedakan masing-masing kelas dengan benar.

#### 3. Train Model ( Melatih Model )

Data yang sudah dilabeli akan digunakan untuk melatih model machine learning. Proses pelatihan ini dilakukan agar model dapat mengenali pola-pola dalam data dan mampu membuat prediksi yang akurat terhadap data baru.

#### 4. Export Model ( Mengekspor Model )

Setelah proses pelatihan selesai, model yang sudah jadi akan diekspor kedalam bentuk file yang sesuai agar dapat digunakan pada perangkat edge. Model ini biasanya sudah dioptimalkan agar ukurannya kecil dan efisien digunakan di perangkat dengan sumber daya terbatas.

#### 5. Run Model ( Menjalankan Model )

Tahap terakhir adalah menjalankan model pada perangkat seperti Arduino atau ESP32. Model ini akan langsung digunakan untuk mengenali data baru secara real-time tanpa perlu mengandalkan koneksi internet atau server eksternal.

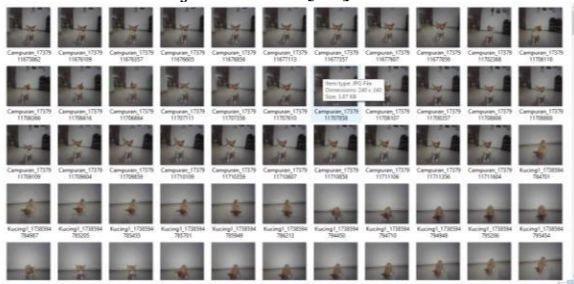


Gambar 6 Alur Kerja TinyML

## II. TAHAP PELAKSANAAN

### A. Dataset Images

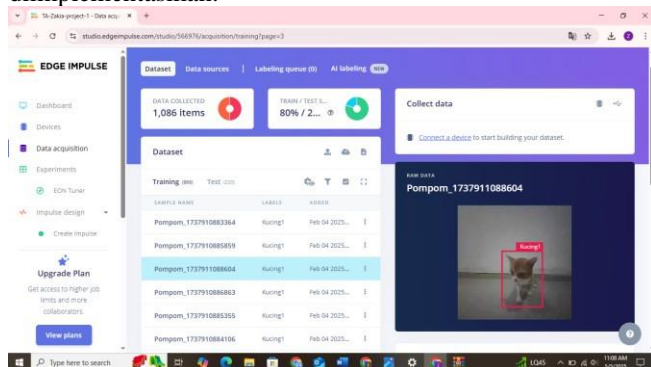
Dataset images merupakan kumpulan besar data gambar yang telah diberi label dan diklasifikasikan, yang digunakan sebagai sumber pelatihan dan pengujian system computer untuk memahami objek tersebut[16].



Gambar 7 Dataset Images

### B. Alur Kerja Sistem

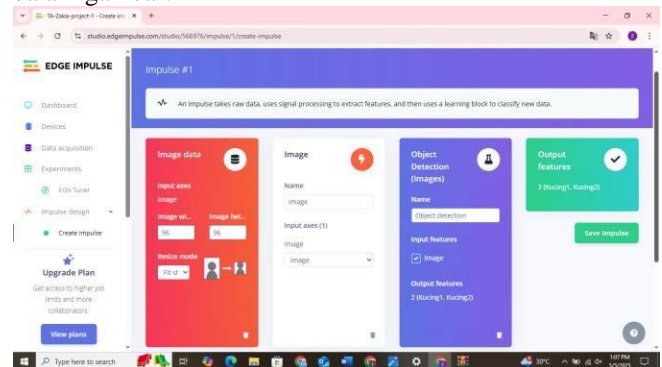
Pengumpulan dataset merupakan tahap awal dalam proses pelatihan model. Dataset ini diperoleh dengan mengambil gambar dua jenis objek, yaitu kucing1 dan kucing2, menggunakan kamera ESP32-CAM yang telah diprogram dengan fitur Collect Images for Edge Impulse. Total gambar yang berhasil dikumpulkan adalah sebanyak 1.086 gambar, yang terdiri dari 543 gambar kucing1 dan 543 gambar kucing2. Data yang telah terkumpul kemudian dibagi menjadi dua bagian, yaitu data pelatihan (training data) dan data pengujian (testing data) dengan rasio 80:20. Setelah itu, gambar-gambar tersebut diunggah ke platform Edge Impulse, kemudian dilanjutkan dengan proses pelabelan melalui fitur labeling queue. Tujuan pelabelan ini adalah untuk mengidentifikasi dan mengelompokkan masing-masing gambar ke dalam kelas yang sesuai, sehingga model dapat melakukan deteksi objek secara akurat saat diimplementasikan.



Gambar 8 Pelabelan Objek

Tahap berikutnya adalah Create Impulse, yaitu proses merancang alur pemrosesan data dari input mentah hingga menghasilkan model machine learning (ML) yang siap dilatih. Impulse ini berfungsi sebagai rangkaian kerja (workflow) yang terdiri dari beberapa blok, seperti blok pemrosesan (processing block) dan blok pembelajaran (learning block). Karena dataset yang digunakan berupa gambar, maka pada tahap ini dipilih blok pemrosesan Image, yang berguna untuk mengubah gambar mentah

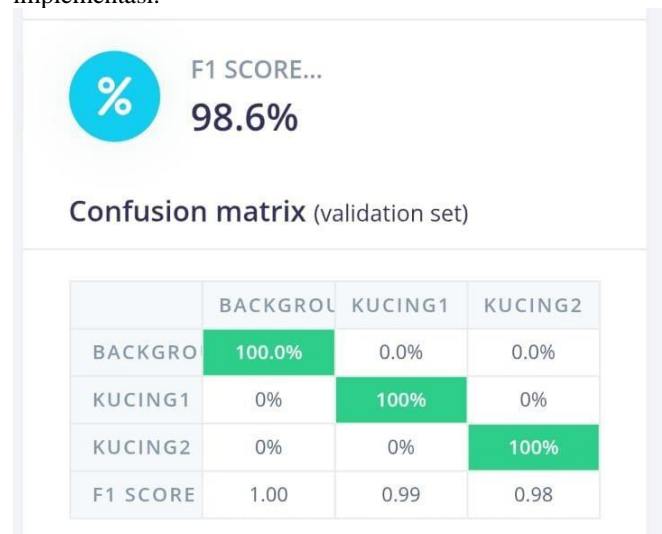
menjadi representasi fitur yang dapat diproses oleh algoritma ML. Selanjutnya, ditambahkan blok pembelajaran Object Detection (Image), yang digunakan untuk melatih model dalam mengenali dan membedakan objek berdasarkan label yang telah diberikan sebelumnya. Pemilihan blok ini disesuaikan dengan tujuan proyek, yaitu mendeteksi objek dalam gambar.



Gambar 9 Pembangunan Model

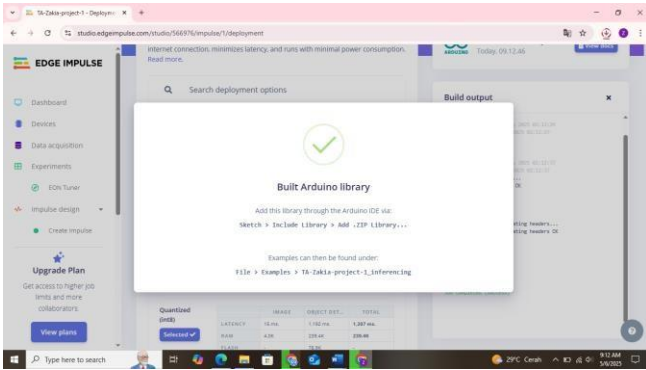
Tahap selanjutnya adalah training atau pelatihan model, yaitu proses di mana sistem belajar dari dataset yang telah dilabeli untuk membentuk pola atau model klasifikasi. Pada tahap ini, data pelatihan digunakan untuk menyesuaikan bobot-bobot dalam jaringan neural agar model mampu mengenali perbedaan antar kelas dengan akurat. Setelah proses pelatihan selesai, sistem akan menampilkan beberapa metrik evaluasi yang mencerminkan kinerja model, seperti akurasi, precision, recall, F1-score, dan loss.

Salah satu komponen penting dalam evaluasi ini adalah confusion matrix, yaitu tabel yang menunjukkan jumlah data uji yang diklasifikasikan dengan benar dan yang salah untuk setiap kelas. Dengan melihat confusion matrix, pengguna dapat mengetahui kelas mana yang sering salah diprediksi dan melakukan penyesuaian pada data atau arsitektur model[17]. Informasi ini sangat penting untuk mengoptimalkan performa model sebelum tahap implementasi.



Gambar 10 Confusion Matrix Training Output

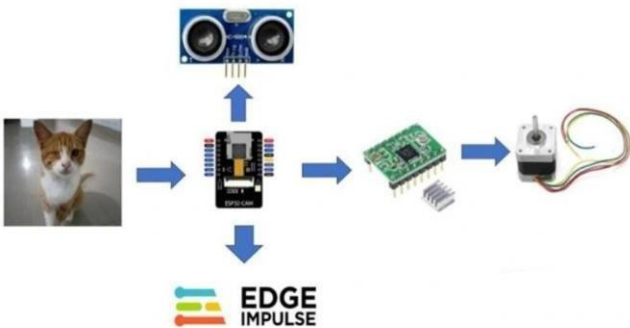
Setelah latihan model pembelajaran mesin selesai, langkah berikutnya adalah deployment, yang berarti memasukkan model ke perangkat seperti ESP32 atau Arduino. Di Edge Impulse, model dapat diunduh dalam format yang berbeda, seperti library Arduino dalam format file.zip yang siap diimpor ke Arduino IDE. Selain itu, informasi estimasi penggunaan memori dan waktu inferensi tersedia untuk memastikan bahwa model akan dijalankan langsung di perangkat, bahkan jika tidak ada koneksi internet.



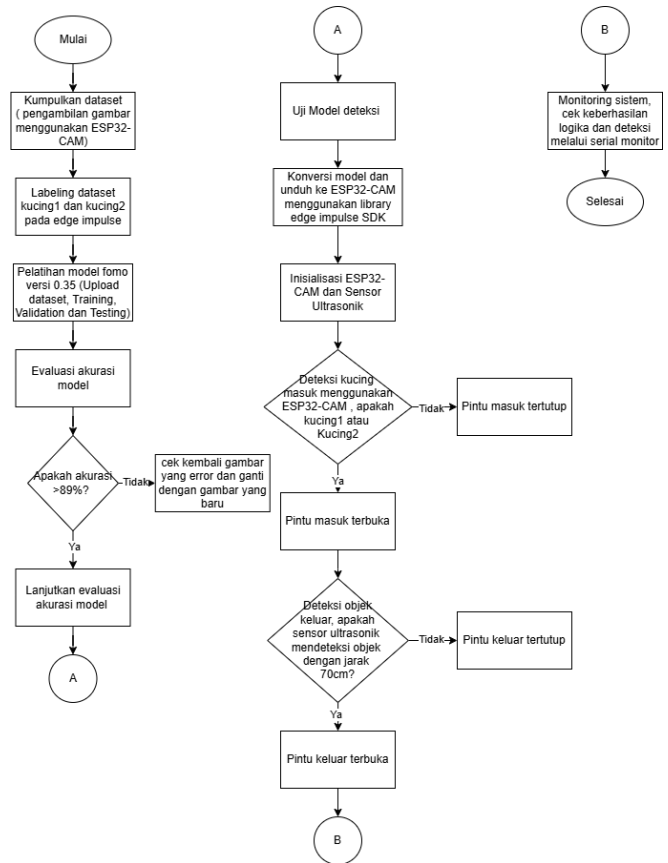
Gambar 11 Tahap Deployment

Model TinyML dari Edge Impulse (dengan dua label Kucing1 dan Kucing2) selesai di deploy pada ESP32-CAM, sistem langsung standby untuk inferensi real-time. Ketika kamera mendeteksi Kucing1 atau Kucing2, ESP32-CAM mengirimkan sinyal ke driver motor stepper A4988. Sinyal ini memungkinkan motor stepper bergerak dan membuka pintu kandang. Setelah pintu terbuka, motor berhenti sejenak selama 5 detik (delay), lalu motor stepper kembali berputar terbalik untuk menutup pintu secara otomatis.

Jika objek yang ditemukan bukan Kucing1 atau Kucing2, sistem tidak mengeksekusi perintah apa pun, motor tidak bergerak dan pintu tidak terbuka. Setelah itu, sensor ultrasonik HC-SR04 disambungkan ke ESP32-CAM untuk melacak jarak objek di dalam kandang. Jika sensor mendeteksi kucing pada jarak kurang dari 15 cm (tanda kucing berada di depan pintu dari dalam), ESP32-CAM memicu driver motor stepper untuk membuka pintu dari dalam, memungkinkan Kucing1 atau Kucing2 keluar.



Gambar 12 Diagram Blok Sistem



Gambar 13 Flowchart Alur Kerja Sistem

### III. HASIL DAN PEMBAHASAN

Setelah tahap pelatihan selesai, tujuan pengujian adalah untuk menilai kinerja komponen dan sistem deteksi objek. Data uji diproses secara real-time dengan dua kucing yang telah dilabeli sebelumnya (Kucing1 dan Kucing2), serta dua kucing tambahan yang belum dilatih modelnya. Untuk memastikan keakuratan dan stabilitas deteksi, model Edge Impulse diuji langsung pada objek nyata.

#### A. Objek Pelatihan

Dibawah ini merupakan sampel objek yang digunakan untuk pengujian oleh penulis, yaitu Kucing1 dan Kucing2. Penulis menggunakan 2 sampel untuk pengujian yang sudah di latih dan 2 sampel pengujian yang tidak dilatih.



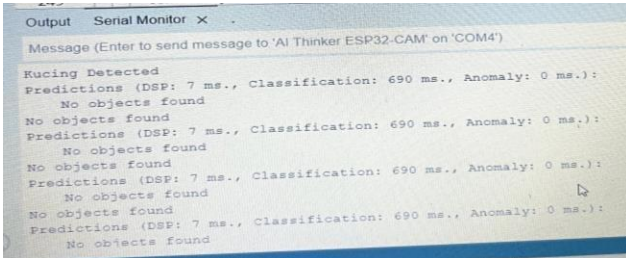
Kucing1

Kucing2

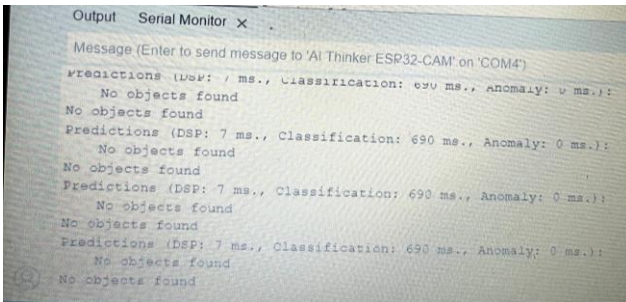


BK yang dilatih1

BK yang di latih2



Output BK yang dilatih1



Output BK yang dilatih2

Gambar 14 Sampel Uji dan Output

Dapat dilihat pada gambar diatas bahwa Kucing1 dan Kucing2 merupakan objek yang telah di latih , pada gambar BK ( Bukan Kucing ) yang dilatih1 dan BK ( Bukan Kucing ) yang dilatih2 merupakan Bukan Kucing yang telah dilatih yang juga digunakan sebagai sampel pelatihan. Proses pengambilan data memastikan setiap objek memiliki ciri khas yang berbeda seperti warna bulu, bentuk wajah, dll, agar model dapat belajar dan membedakan objek yang telah di latih dengan tepat. Selanjutnya, akurasi serta presentase keberhasilan dari setiap pengujian dapat dihitung dengan persamaan dibawah ini :

$$\text{Akurasi} = \frac{TP+TF}{TP+TN+FP+FN} \times 100\% [18]$$

$$\text{Tingkat Keberhasilan} = \frac{\text{Jumlah Deteksi Yang Benar}}{\text{Total Percobaan}} [19]$$

Istilah yang biasa dikenal dari rumus diatas yaitu sebagai berikut :

- TP (True Positive): Jumlah data yang sebenarnya positif dan juga diprediksi positif.

- FP (False Positive): Jumlah data yang sebenarnya negatif, tapi diprediksi positif.
- FN (False Negative): Jumlah data yang sebenarnya positif, tapi diprediksi negatif.
- TN (True Negative): Jumlah data yang sebenarnya negatif dan juga diprediksi negatif[20]

### B. Pengujian Kucing yang di latih dan Tidak dilatih

Empat kucing sebagai sampel digunakan untuk menguji prototipe. Setiap kucing diuji sebanyak lima belas kali. Selama proses ini, data yang dimasukkan dibandingkan dengan output yang dihasilkan oleh model yang telah dilatih. Berikut adalah ringkasan hasil tes :

Tabel 1 Confusion Matrix Pengujian Objek

Prediksi \ Aktual	Kucing1	Kucing2	BK yg dilatih1 ( No object Found )	BK yg dilatih2 ( No object Found )
Kucing1	15	0	0	0
Kucing2	4	11	0	0
BK yg dilatih1	0	0	15	0
BK yg dilatih2	0	0	0	15

Berdasarkan Confusion Matrix pada Tabel 1 dengan 15 sampel untuk masing-masing dari kelas Kucing1 , Kucing2, BK yang dilatih1 dan BK yang dilatih2 total 60 percobaan yang di uji berhasil mengklasifikasikan 56 sampel dengan benar ( 93,33%), sedangkan 4 sampel ( 6,67 % ) diidentifikasi sebagai loss/error. Tingkat akurasi keseluruhan diperkirakan sebesar 93,33 % sesuai dengan hasil perhitungan di bawah ini :

$$\text{Akurasi Kucing1} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{15 + 41}{15 + 4 + 41 + 0} = \frac{56}{60} \times 100\% = 93\%$$

$$\text{Akurasi Kucing2} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{15 + 45}{11 + 0 + 45 + 4} = \frac{56}{60} \times 100\% = 93\%$$

$$\text{Akurasi BKdilatih1} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{15 + 45}{15 + 0 + 45 + 0} = \frac{60}{60} \times 100 = 100\%$$

$$\text{Akurasi BKdilatih2} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{15 + 45}{15 + 0 + 45 + 0} = \frac{60}{60} \times 100 = 100\%$$

$$\text{Total Akurasi} = \frac{\text{Jumlah deteksi yang benar (TP)}}{\text{Total Percobaan (N)}} = \frac{56}{60} \times 100\% = 93\%$$

$$\text{Presisi} \frac{TP}{TP+FP} = \frac{56}{60} \times 100 = 93\%$$

$$\text{Recall} \frac{TP}{TP+FN} = \frac{56}{60} \times 100 = 93\%$$

$$\text{F1 Score} \frac{2 \times \text{presisi} \times \text{Recall}}{\text{Presisi} + \text{Recall}} = \frac{2 \times 0,93\% \times 0,93\%}{0,93 + 0,93} \times 100 = 93\%$$

$$\text{Loss} = 1 - \text{Akurasi} = 1 - 0,93 = 6,67\%$$

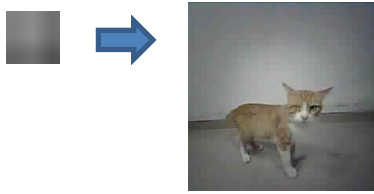
Berikut merupakan persamaan yang digunakan oleh platform :

1. Pre-Processing merupakan langkah awal dalam pengolahan data sebelum data tersebut dimasukkan kedalam model

- Normalisasi (per channel) (Menyamakan skala nilai pixel (atau fitur)

$$\chi^1 = \frac{\chi - \mu}{\sigma}$$

Di mana  $\mu, \sigma$  adalah mean & standar deviasi dataset



Gambar 15 Pixel gambar object

Contoh sebagai berikut :

Gambar diatas merupakan 3x3 pixel dari gambar asli dengan 240x240 pixel :

Hasil dari mean & standar deviasi 3x3 pixel yaitu :

Tabel 2 Hasil nilai pixel 3x3

Baris 1	124, 126, 121	142, 144, 139	112, 114, 113
Baris 2	126, 128, 123	141, 143, 138	113, 115, 114
Baris 3	98, 99, 93	109, 110, 104	83, 83, 83

Hitung mean ( $\mu$ ) :

$$\mu_R = \frac{124+142+112+126+141+113+98+109+83}{9} = \frac{1048}{9} = 116,44$$

Hitung Standar deviasi ( $\sigma$ )

$$\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

Tabel 3 Hasil perhitungan standar deviasi

x	Normalisasi x - $\mu$	$(x - \mu)^2$
124	7,56	57,15
142	25,56	653,39
112	-4,44	19,71
126	9,56	91,38
141	24,56	603,30
113	-3,44	11,83
98	-18,44	340,08
109	-7,44	55,38
83	-33,44	1118,90

Total kuadrat selisih :  $\frac{2951,12}{9} = 327,90$

Standar deviasi :  $\sigma_R = \sqrt{327,90} = 18,11$

Gunakan rumus :  $\chi^1 = \frac{\chi - \mu}{\sigma}$

Tabel 4 Hasil perhitungan channel R

x	Normalisasi $x^1 = (\chi - 116,44) / 18,11$	Hasil dalam satu channel
124	$(124 - 116,44) / 18,11$	0,42
142	$(142 - 116,44) / 18,11$	1,41
112	$(112 - 116,44) / 18,11$	-0,25
126	$(126 - 116,44) / 18,11$	0,53
141	$(141 - 116,44) / 18,11$	1,36
113	$(113 - 116,44) / 18,11$	-0,19
98	$(98 - 116,44) / 18,11$	-1,02

109	$(109 - 116,44) / 18,11$	-0,41
83	$(83 - 116,44) / 18,11$	-1,84

2. Feature Extraction adalah proses untuk mengambil informasi penting dari data mentah, agar data tersebut dapat direpresentasikan dalam bentuk yang lebih sederhana dan relevan untuk keperluan pemrosesan lanjutan atau pelatihan model ( Menangkap pola (tepi, tekstur) dari citra input )

- Convolution (2D)

$$I \times \mathcal{K}(i, j) = \sum_n I(i + m, j + n) \mathcal{K}(m, n) + b$$

Tabel 5 Contoh hasil nilai kernel

K_R	K_G	K_B
-1	0	1
-2	0	2
-1	0	1

Channel R =  $(-1 \times 124) + (0 \times 142) + (1 \times 112) + (-2 \times 126) + (0 \times 141) + (2 \times 113) + (-1 \times 98) + (0 \times 109) + (1 \times 83) = -53$

Untuk mencari chanel G dan B dapat melanjutkan perhitungan pada tabel 2 dan dikalikan dengan sesuai kernelnya

3. Aktivasi berfungsi untuk mengubah output dari setiap neuron berdasarkan input yang diterima. Contoh perhitungan :

- Sigmoid (binary) :  $\sigma(x) = \frac{1}{1+e^{-x}}$  (Mengubah nilai logit menjadi probabilitas (0-1) di kasus binary classification.)

Pada hasil ekstraksi fitur diatas kita

mendapatkan hasil -53 maka :  $\sigma(-53) =$

$$\frac{1}{1+e^{53}}$$

$$e^{53} = 1,041 \times 10^{23}$$

$$\sigma(-53) = \frac{1}{1+1,041 \times 10^{23}} = \frac{1}{1,041 \times 10^{23}} = 9,6 \times 10^{-24}$$

$$\sigma(-53) = 0,096$$

- Softmax ( multi-class/per-grid-cell) :  $\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$  (Mengubah sekumpulan skor (logit)

menjadi probabilitas multi-kelas (jumlah =1)

Softmax biasanya digunakan untuk klasifikasi 3 kelas contoh :

Misalnya logit (z) : [ 3,0, 1,0, 0,2]

Maka Langkah-langkah menghitung softmax :

$$e^{3,0} = 20,09, e^{1,0} = 2,71, e^{0,2} = 1,22$$

$$20,09 + 2,71 + 1,22 = 24,02$$

$$\hat{y}_1 = \frac{20,09}{24,02} = 0,836$$

$$\hat{y}_2 = \frac{2,71}{24,02} = 0,113$$

$$\hat{y}_3 = \frac{1,22}{24,02} = 0,051$$

- Relu merupakan fungsi yang memfilter nilai negative menjadi 0 dan hanya melewati nilai positif ke tahap berikutnya dalam jaringan saraf : (ReLU) :  $\text{ReLU}(\chi) = \max(0, x)$

$$\text{ReLU}(\chi) = \begin{cases} x, & \text{jika } x > 0 \\ 0, & \text{jika } x \leq 0 \end{cases}$$

4. Loss Function berfungsi untuk mengukur kesalahan atau selisih antara output yang diprediksi oleh model dan label sebenarnya, contoh perhitungan sebagai berikut :

- Cross - Entropy Loss ( multi -  $\frac{\text{class}}{\text{per}}$  - cell) :  $\text{LCE} = -\sum_{i=1}^n y_i \log(\hat{y}_i)$  (Mengukur jarak

antara label asli  $y$  dan prediksi  $\hat{y}$ . Biasanya digunakan untuk training klasifikasi). Untuk mendapatkan nilai  $L_{CE}$ :  $y_i$ : label asli (biasanya dapat melihat hasilnya pada saat training dataset) misalnya [1,1,0]  
 $\hat{y}_i$ : prediksi probabilitas (misalnya menggunakan hasil dari softmax diatas) [0,8, 0,1, 0,5]  
 $LCE = (1 \times \log(0,8) + 1 \times \log(0,1) + 0 \times \log(0,5)) = -( -2,5257) = 2,5257$

- Mean Squared Error (MSE) (regresi)  $LMSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$  (Digunakan di regresi atau autoencoder untuk mengukur rata-rata kuadrat selisih)  
 $LMSE = [1 - 0,8, 1 - 0,1, 0 - 0,5]^2 = [1, 0,9, -0,5]^2$   
dikuadratkan = [1, 0,81, 0,25]  
Rata-rata =  $\frac{1+0,81+0,25}{3} = 0,68$
- Mean Absolute Error (MAE):  $LMAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$   
Selisih Mutlak = [1, 0,9, -0,5]  
rata - rata =  $\frac{2,4}{3} = 0,8$

#### 5. Optimizer (biasanya menggunakan Adam)

Update parameter  $\theta$  :

$$\text{Momentum pertama } m_t = \beta_1 m_{t-1} + (1 - \beta_1) G_t$$

$$\text{Momentum kedua (varians)} v_t = \beta_2 v_{t-1} + (1 - \beta_2) G_t^2$$

$$\text{Bias correction } \hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\text{Update parameter } \theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

Tabel 6 Contoh nilai dari parameter

Parameter	Nilai
$\theta$	0,5
$G_t$	0,2
$m_t$	0
$v_t$	0
$\beta_1$	0,9
$\beta_2$	0,999
$\alpha$	0,01
$\epsilon$	10 <sup>-8</sup>
$t$	1

$$\text{hitung } m_t = \beta_1 m_{t-1} + (1 - \beta_1) G_t = 0,9 \times 0 + 0,1 \times 0,2 = 0,02$$

$$\text{hitung } v_t = \beta_2 v_{t-1} + (1 - \beta_2) G_t^2 = 0,999 \times 0 + 0,001 \times (0,2)^2 = 0,0004$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} = \frac{0,02}{1 - 0,9^1} = \frac{0,02}{0,1} = 0,2$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} = \frac{0,0004}{1 - 0,999^1} = \frac{0,0004}{0,001} = 0,4$$

$$\text{Update parameter } \theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

$$0,5 - 0,01 \times \frac{0,2}{\sqrt{0,4} + 10^{-8}} = 0,5 -$$

$$0,01 \times$$

$$\frac{0,2}{0,2 + 10^{-8}} = 0,5 - 0,01 = 0,49$$

- $IoU$  (object detection) =  $\frac{\text{overlap area}}{\text{union area}}$

Contoh data  $IoU$  : Ground Truth Box : ( $x_1 = 1, y_1 = 1, x_2 = 4, y_2 = 4$ )

Prediksi Box : ( $x_1 = 2, y_1 = 2, x_2 = 5, y_2 = 5$ )

$x_1$  overlap =  $\max(1,2) = 2$

$y_1$  overlap =  $\max(1,2) = 2$

$x_2$  overlap =  $\min(4,5) = 4$

$y_2$  overlap =  $\min(4,5) = 4$

overlap luas =  $(4-2) \times (4-2) = 4$

Union :  $9 + 9 - 4 = 14$

$$IoU = \frac{4}{14} = 0,29$$

- $mAP$  (mean Average Precision) =

$$\frac{1}{N} \sum_{i=1}^N AP_i$$

Contoh data recall untuk dua kelas yaitu 1,0 dan 0,8

sedangkan untuk presisi yaitu, 0,8 dan 0,7 maka :

$$AP = (1,0 - 1,8) \times 0,8 = 0,2 \times 0,8 = 0,16$$

#### C. Evaluasi Hasil

Dari hasil pengujian menggunakan confusion matrix terhadap empat kelas yaitu Kucing1, Kucing2, BK yang dilatih1, dan BK yang dilatih2, dengan masing-masing kelas diuji sebanyak 15 kali, sistem menunjukkan performa yang sangat baik dalam melakukan klasifikasi. Hasil evaluasi menunjukkan bahwa model mampu mengklasifikasikan 56 dari 60 data uji dengan benar, menghasilkan akurasi sebesar 93,33%, precision 93,33%, recall 93,33%, dan F1 score sebesar 93,33%, dengan tingkat error/loss sebesar 6,67%.

Secara umum, dengan menggunakan model FOMO alpha 0.35 sistem dapat bekerja optimal dalam mengenali objek yang sudah dilatih sebelumnya (Kucing1 dan Kucing2). Hal ini disebabkan karena model hasil deploy memiliki ukuran memori sebesar 192kB, yang sesuai dan dapat dijalankan dengan baik pada konfigurasi alpha 0.35. Selain itu kesalahan klasifikasi hanya terjadi pada Kucing2 yang terdeteksi sebagai Kucing1 sebanyak 4 kali, namun tidak terdapat kesalahan pada kelas lainnya. Hal ini menunjukkan bahwa model memiliki presisi dan sensitivitas yang seimbang, serta mampu mengklasifikasikan objek yang sudah dikenal dengan tingkat keandalan tinggi.

#### KESIMPULAN

Berdasarkan hasil pengujian sistem pendeteksi kucing menggunakan ESP32-CAM dan model dari Edge Impulse, dapat disimpulkan bahwa sistem mampu mengidentifikasi dan mengklasifikasikan objek dengan akurasi yang tinggi. Dengan keberhasilan klasifikasi mencapai 93,33%, sistem ini dapat diandalkan untuk membuka pintu kandang secara otomatis ketika kucing yang telah dikenali berada di depan kamera.

Model juga menunjukkan performa yang konsisten untuk semua kelas yang diuji, baik pada objek kucing maupun bukan kucing. Tingkat kesalahan yang rendah menunjukkan bahwa sistem ini dapat digunakan untuk aplikasi nyata, khususnya dalam pengelolaan akses kandang hewan peliharaan berbasis visi komputer. Untuk meningkatkan akurasi lebih lanjut, pengumpulan data latihan tambahan dengan variasi posisi, cahaya, dan orientasi objek dapat dilakukan.

## REFERENCES

- [1] Y. Susanthi, A. Darmawan, and S. B. Suryadi, "Alat Pengenal Gerakan Tangan Menggunakan Algoritma Artificial Neural Network Berbasis ESP32 dan Platform Edge Impulse," *Techne J. Ilm. Elektrotek.*, vol. 23, no. 1, pp. 77–94, 2024, doi: 10.31358/techne.v23i1.434.
- [2] M. F. Rianti and W. Wildian, "Rancang Bangun Alat Pembersih Kotoran dan Pembersih Pakan Kucing Berbasis Modul Arduino Uno R3 Menggunakan Sensor Load Cell dan Sensor Inframerah," *J. Fis. Unand*, vol. 11, no. 2, pp. 221–227, 2022, doi: 10.25077/jfu.11.2.221-227.2022.
- [3] S. A. Wulandari and H. D. Pohan, "Pet Attachment dan Psychological Well-being pada Pemilik Kucing," vol. 3, no. 2, 2023.
- [4] D. Wagner, K. Hurley, and J. Stavisky, "Shelter housing for cats: Practical aspects of design and construction, and adaptation of existing accommodation," *J. Feline Med. Surg.*, vol. 20, no. 7, pp. 643–652, 2018, doi: 10.1177/1098612X18781390.
- [5] M. R. Heranof and D. Yendri, "Alat Kandang Kucing Otomatis Berbasis Mikrokontroler dengan Monitoring Telegram," *Chipset*, vol. 4, no. 01, pp. 71–79, 2023, doi: 10.25077/chipset.4.01.71-79.2023.
- [6] T. W. O. Putri and M. A. Darmawan, "Prototipe Sistem Kendali Jarak Jauh Pada Pakan dan Pintu Kandang Kucing," *Sutet*, vol. 12, no. 1, pp. 21–30, 2022, doi: 10.33322/sutet.v12i1.1664.
- [7] A. M. Idham, A. Candra, and A. N. Faizah, "PROTOTYPE LARANGAN PENGENDARA MELEWATI ZEBRA CROSS DI TRAFFIC LIGHT BERBASIS ESP32 CAM PROTOTYPE FOR PROHIBITING DRIVERS TO PASSAGE ZEBRA CROSS IN ESP32 CAM-BASED TRAFFIC LIGHT PENDAHULUAN Indonesia merupakan Negara yang memiliki banyak penduduk yang juga," vol. 1, no. 1, 2023.
- [8] H. Purwanto, M. Riyadi, D. W. W. Astuti, and I. W. A. W. Kusuma, "Komparasi Sensor Ultrasonik HC-SR04 Dan JSN-SR04T Untuk Aplikasi Sistem Deteksi Ketinggian Air," *J. SIMETRIS*, vol. 10, no. 2, pp. 717–724, 2019.
- [9] S. Hymel *et al.*, "Edge Impulse: An MLOps Platform for Tiny Machine Learning," 2022, [Online]. Available: <http://arxiv.org/abs/2212.03332>
- [10] M. Wibowo, R. Tullah, W. Ricesa, and P. T. Informatika, "Studi Perbandingan Algoritma YOLO dan FOMO untuk Object Detection pada Perangkat ESP32-CAM," vol. 11, no. 1, pp. 44–54, 2025.
- [11] G. Delnevo, S. Mirri, C. Prandi, and P. Manzoni, "An evaluation methodology to determine the actual limitations of a TinyML-based solution," *Internet of Things (Netherlands)*, vol. 22, p. 100729, 2023, doi: 10.1016/j.iot.2023.100729.
- [12] S. Zhuo *et al.*, *An Empirical Study of Low Precision Quantization for TinyML*, vol. 1, no. 1. Association for Computing Machinery, 2022. [Online]. Available: <http://arxiv.org/abs/2203.05492>
- [13] D. Nafis Alfarizi, R. Agung Pangestu, D. Aditya, M. Adi Setiawan, and P. Rosyani, "Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis," *J. Artif. Intel. dan Sist. Penunjang Keputusan*, vol. 1, no. 1, pp. 54–63, 2023, [Online]. Available: <https://jurnalmahasiswa.com/index.php/aidanspk>
- [14] P. Romadloni, B. Adhi Kusuma, and W. Maulana Baihaqi, "Komparasi Metode Pembelajaran Mesin Untuk Implementasi Pengambilan Keputusan Dalam Menentukan Promosi Jabatan Karyawan," *JATI (Jurnal Mhs. Tek. Inform.)*, vol. 6, no. 2, pp. 622–628, 2022, doi: 10.36040/jati.v6i2.5238.
- [15] J. R. Ziliwu, G. C. Setyawan, and H. Budiati, "Penerapan ESP32-CAM dan TinyML dalam Klasifikasi Gambar Buah dan Sayuran," *Jutisi J. Ilm. Tek. Inform. dan Sist. Inf.*, vol. 13, no. 1, p. 584, 2024, doi: 10.35889/jutisi.v13i1.1869.
- [16] A. W. Hasbullah, E. Setiawan, and A. Rachmad, "Evaluasi Keandalan Model Rekognisi Suara Burung Hama Menggunakan Platform Edge Impulse Pada Mikrokontroler Low Power," *J. Tek. Elektro dan Komput. TRIAC*, vol. 10, no. 2, pp. 69–75, 2023, doi: 10.21107/triac.v10i2.22448.
- [17] R. Nurhidayat and K. E. Dewi, "Penerapan Algoritma K-Nearest Neighbor Dan Fitur Ekstraksi N-Gram Dalam Analisis Sentimen Berbasis Aspek," *Komputa J. Ilm. Komput. dan Inform.*, vol. 12, no. 1, pp. 91–100, 2023, doi: 10.34010/komputa.v12i1.9458.
- [18] L. W. Astuti, E. P. Sari, I. Saluza, F. Faradillah, and R. Yunita, "Penentuan Tingkat Kekumuhan Permukiman Kumuh Kota Palembang Dengan Metode Algoritma K-Means Clustering Dan Algoritma Id3," *Intech*, vol. 2, no. 1, pp. 31–38, 2021, doi: 10.54895/intech.v2i1.869.
- [19] A. Sani and S. Rahmadinni, "Deteksi Gestur Tangan Berbasis Pengolahan Citra," *J. Rekayasa Elektr.*, vol. 18, no. 2, pp. 115–124, 2022, doi: 10.17529/jre.v18i2.25147.
- [20] K. Neutrino Sae B, S. Derwin R, and R. Nelci Dessy, "Kajian Machine Learning Dengan Komparasi Klasifikasi Prediksi Dataset," *J-Icon*, vol. 7, no. 1, pp. 37–49, 2019.

## ORIGINALITY REPORT

9%

SIMILARITY INDEX

7%

INTERNET SOURCES

5%

PUBLICATIONS

3%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="#">jurnalmahasiswa.com</a> Internet Source	1%
2	Submitted to Universitas Muslim Indonesia Student Paper	1%
3	<a href="#">jfu.fmipa.unand.ac.id</a> Internet Source	1%
4	<a href="#">stt-pln.e-journal.id</a> Internet Source	1%
5	Submitted to Universitas Muhammadiyah Surakarta Student Paper	1%
6	Putra Arimbawa I Gusti Ngurah, I Made Dwi Darma Artanaya, Komang Ayu Krisna Dewi, I Putu Gede Abdi Sudiatmika. "AI-Based Tourist Guide Application in Bali Using Supervised Learning Method", ARRUS Journal of Engineering and Technology, 2024 Publication	1%
7	<a href="#">journal.unhas.ac.id</a> Internet Source	1%
8	<a href="#">jurnal.umus.ac.id</a> Internet Source	1%
9	Hakon Fahmi, Sutisna. "Implementasi Data Mining Klasifikasi Gejala Penyakit TB Menggunakan Algoritma Naive Bayes pada Studi Kasus Puskesmas Pegangsaan Dua B",	<1%

- 
- |    |  |      |
|----|--|------|
| 10 | <a href="https://export.arxiv.org">export.arxiv.org</a><br>Internet Source   | <1 % |
| 11 | Abdullah Sani, Risqi Sani Pratama. "Oxidation Detection on PCB Using Image Processing", 2020 3rd International Conference on Applied Engineering (ICAE), 2020<br>Publication   | <1 % |
| 12 | Roseline Oluwaseun Ogundokun, Aiman Li, Ronke Seyi Babatunde, Chinecherem Umezuruike et al. "Enhancing Skin Cancer Detection and Classification in Dermoscopic Images through Concatenated MobileNetV2 and Xception Models", Bioengineering, 2023<br>Publication | <1 % |
| 13 | <a href="https://doaj.org">doaj.org</a><br>Internet Source   | <1 % |
| 14 | <a href="https://eprints.undip.ac.id">eprints.undip.ac.id</a><br>Internet Source   | <1 % |
| 15 | Engel Manuel Punuh. "Rancang Bangun Sensor Parkir Kendaraan Roda Empat Berbasis Mikrokontroler Arduino Uno", Jambura Journal of Electrical and Electronics Engineering, 2024<br>Publication  | <1 % |
| 16 | Doddi Yuniardi, Rani Puspita, Ridwan Ridwan. "MENINGKATKAN EFISIENSI PENGENDALIAN SUHU BOILER PADA PLTSa BURANGKENG MELALUI ANALISIS KOMPREHENSIF BERBASIS MACHINE LEARNING", Jurnal Teknik dan Science, 2023<br>Publication                                     | <1 % |
-

17

Luntungan Stephen Pieters. "Development of Automatic Waste Classification System using CNN-Based Deep Learning to Support Smart Waste Management", INOVTEK Polbeng - Seri Informatika, 2025

Publication

<1 %

---

Exclude quotes Off

Exclude matches Off

Exclude bibliography On

4242101025\_Zakia\_Putri-1746808130412

---

PAGE 1

---

PAGE 2

---

PAGE 3

---

PAGE 4

---

PAGE 5

---

PAGE 6

---

PAGE 7

---

PAGE 8

---