



SISTEM DETEKSI KERETAKAN TELUR AYAM BERBASIS *ARTIFICIAL INTELLIGENCE* MENGGUNAKAN ESP32-CAM

TUGAS AKHIR



**Program Studi
S1 TEKNIK KOMPUTER**

**UNIVERSITAS
Dinamika**

Oleh:

Cut Salsabila Amani

21410200004

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2025

**SISTEM DETEKSI KERETAKAN TELUR AYAM BERBASIS *ARTIFICIAL*
INTELLIGENCE MENGGUNAKAN ESP32-CAM**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana Teknik Komputer**



UNIVERSITAS
Dinamika

Oleh:

Nama : Cut Salsabila Amani
NIM : 21410200004
Program Studi : S1 Teknik Komputer

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA**

2025

TUGAS AKHIR

SISTEM DETEKSI KERETAKAN TELUR AYAM BERBASIS *ARTIFICIAL INTELLIGENCE* MENGGUNAKAN ESP32-CAM

Dipersiapkan dan disusun oleh:

Cut Salsabila Amani

NIM: 21410200004

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 22 Januari 2025

Susunan Dewan Pembahas

Pembimbing:

I. Harianto, S.Kom., M.Eng.

NIDN: 0722087701



cn=Harianto Harianto,
o=Universitas Dinamika,
ou=Prodi S1 Teknik Komputer,
email=hari@dinamika.ac.id, c=ID
2025.01.24 10:28:01 +07'00'

II. Weny Indah Kusumawati, S.Kom., M.MT.

NIDN: 0721047201



cn=Weny Indah Kusumawati,
o=Undika, ou=Prodi S1 TK -
FTI,
email=weny@dinamika.ac.id,
c=ID
2025.01.23 16:47:06 +07'00'

Pembahas:

Heri Pratikno, M.T.

NIDN: 0716117302



Digitally signed by Heri Pratikno, M.T.
DN: cn=Heri Pratikno, M.T.,
o=Universitas Dinamika, ou=S1
Teknik Komputer,
email=heri@dinamika.ac.id, c=ID
Date: 2025.01.24 11:17:06 +07'00'
Adobe Acrobat version: 11.0.23

Tugas Akhir ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana



Fakultas Teknologi dan Informatika
UNIVERSITAS

Dinamika

Dr. Anjik Sukmaaji, S.Kom., M.Eng.

NIDN: 0731057301

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA



UNIVERSITAS
“Follow The Flow”
Dinamika



Dengan banyak terima kasih, Saya persembahkan laporan ini kepada orang tua dan semua pihak yang terlibat dan mendukung saya selama proses ini.

UNIVERSITAS
Dinamika

PERNYATAAN
PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama : **Cut Salsabila Amani**
NIM : **21410200004**
Program Studi : **S1 Teknik Komputer**
Fakultas : **Fakultas Teknologi dan Informatika**
Jenis Karya : **Laporan Tugas Akhir**
Judul Karya : **SISTEM DETEKSI KERETAKAN TELUR AYAM
BERBASIS *ARTIFICIAL INTELLIGENCE*
MENGUNAKAN ESP32-CAM**

Menyatakan dengan sesungguhnya bahwa :

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

Demikian surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 14 Januari 2025



Cut Salsabila Amani
NIM : 21410200004

ABSTRAK

Peternakan ayam membutuhkan sistem pemantauan yang efisien untuk mengetahui kondisi telur, terutama yang menetas atau retak. Penelitian ini merancang sistem deteksi keretakan telur ayam berbasis AI menggunakan ESP32-CAM. Sistem ini mengintegrasikan mikrokontroler dengan model deteksi objek untuk mendeteksi perubahan kondisi telur secara otomatis. Tiga model AI digunakan dalam pengembangan, yaitu FOMO (*Faster Object Detection for Mobile Oriented*), YOLO (*You Only Look Once*) dan SSD (*Single Shot Detector*), yang dilatih menggunakan platform *Edge Impulse*. Model YOLO menunjukkan kinerja terbaik pada *Epoch* ke-6 dengan *Learning Rate* 0.1, mAP 0.27, *Precision* 57.0% dan *Recall* 0.42. Model SSD mencapai performa terbaik pada *Epoch* ke-12 dengan *Learning Rate* 0.02, mAP 0.55, *Precision* 66.5% dan *Recall* 0.64. Sementara itu, Model FOMO mencatat *Precision* 0.79, *Recall* 0.84 dan *F1-Score* 0.82 pada *Epoch* ke-9 dengan *Learning Rate* 0.003. Pengujian dilakukan dalam dua kategori: satu butir dan dua butir telur. Pada satu butir telur, akurasi rata-rata adalah 99% untuk kondisi retak dan 96% untuk kondisi tidak retak. Pada dua butir telur dengan empat kondisi berbeda, akurasi rata-rata adalah: (1) telur 1 retak, telur 2 tidak retak: 99%; (2) telur 1 tidak retak, telur 2 retak: 98%; (3) kedua telur retak: 98%; dan (4) kedua telur tidak retak: 93%. Hasil menunjukkan bahwa Model FOMO lebih efisien dalam penggunaan sumber daya dibandingkan model lain, sehingga cocok untuk perangkat IoT seperti ESP32-CAM. Sistem ini membantu peternak memantau kondisi telur secara otomatis, mengurangi risiko kelalaian dan meningkatkan keberhasilan penetasan.

Kata Kunci: Telur Ayam, *Edge Impulse*, *Object Detection*, FOMO, *Prototype*

KATA PENGANTAR

Segala puji dan syukur kehadirat Tuhan Yang Maha Esa atas segala rahmat, nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan laporan Tugas Akhir dengan judul "Sistem Deteksi Keretakan Telur Ayam Berbasis *Artificial Intelligence* Menggunakan ESP32-CAM". Penyusunan laporan Tugas Akhir ini tidak terlepas dari bantuan dan kontribusi berbagai pihak yang telah memberikan saran, kritik, nasihat dan masukan. Oleh karena itu, rasa terima kasih yang sebesar-besarnya disampaikan kepada:

1. Allah SWT, karena atas rahmat dan hidayah-Nya penulis dapat menyelesaikan laporan Tugas Akhir ini.
2. Seluruh keluarga atas dukungan, cinta dan dorongan yang terus-menerus diberikan selama proses penyelesaian laporan Tugas Akhir ini.
3. Bapak Dr. Anjik Sukmaaji, S.Kom., M.Eng., selaku Dekan Fakultas Teknologi dan Informatika Universitas Dinamika.
4. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer.
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku dosen pembahas atas masukan dan sarannya sehingga penulis dapat menyelesaikan Laporan Tugas Akhir ini.
6. Bapak Harianto, S.Kom., M.Eng., yang memegang peran sebagai dosen pembimbing 1, telah banyak memberikan masukan, wawasan dan solusi kepada peneliti selama pengerjaan Tugas Akhir dan pembuatan Laporan Tugas Akhir.
7. Ibu Weny Indah Kusumawati, S.Kom., M.MT., yang memegang peran sebagai dosen pembimbing 2, telah memberikan bimbingan, dukungan, serta motivasi kepada penulis sepanjang proses pengerjaan Tugas Akhir.
8. Laboran S1 Teknik Komputer yang membantu memberikan saran dan masukan bagi penulis, sehingga dapat menyelesaikan pembuatan Tugas Akhir ini.
9. Seluruh Teman-teman S1 Teknik Komputer khususnya Angkatan 2021 yang telah membantu penulis dalam memberi dukungan dan semangat.

10. Seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan bantuan dan dukungan untuk menyelesaikan Tugas Akhir ini.

Semoga segala dukungan, bimbingan dan nasihat dari semua pihak dalam proses pengerjaan Tugas Akhir ini mendapat balasan yang baik dari Tuhan Yang Maha Esa. Penulis sangat menyadari bahwa masih ada kekurangan dalam proses pengerjaan Tugas Akhir ini, maka dari itu penulis mengharapkan kritik dan saran yang membangun dari semua pihak. Semoga laporan Tugas Akhir ini diterima dengan baik dan bermanfaat bagi penulis dan semua pihak yang terlibat.



UNIVERSITAS
Dinamika

DAFTAR ISI

Halaman

ABSTRAK	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	1
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	2
BAB II LANDASAN TEORI.....	3
2.1 Ternak Ayam.....	3
2.2 Mesin Penetas Telur.....	3
2.3 <i>Artificial Intelligence</i>	5
2.3.1 <i>Machine Learning</i>	5
2.3.2 <i>Edge Impulse</i>	6
2.3.3 FOMO.....	9
2.4 ESP32-CAM.....	10
2.5 Arduino IDE.....	11
2.6 OLED 0.96” 128x64.....	12
2.7 Lampu LED (<i>Light Emitting Diode</i>).....	13
BAB III METODOLOGI PENELITIAN.....	14
3.1 Studi Literatur.....	14
3.2 Alat dan Bahan.....	14
3.3 Pengambilan Dataset.....	15
3.4 Pre-Prosessing Dataset.....	16
3.5 Desain Sistem.....	18
3.5.1 Diagram Blok Sistem.....	18

3.5.2 Perancangan <i>Software</i>	19
3.5.3 Perancangan <i>Hardware</i>	20
3.5.4 Perancangan Mekanik.....	20
3.6 Metode Deteksi FOMO.....	21
BAB IV HASIL DAN PEMBAHASAN.....	23
4.1 Validasi Proses <i>Training Data</i>	23
4.1.1 Langkah Validasi Proses <i>Training Data</i>	23
4.1.2 Hasil Validasi Proses <i>Training Data</i>	24
4.2 Pengujian Proses Deteksi Objek.....	26
4.2.1 Langkah Pengujian Proses Deteksi Objek.....	27
4.2.2 Hasil Pengujian Proses Deteksi Objek.....	27
4.3 Kesimpulan Pengujian Deteksi Objek.....	30
BAB V PENUTUP.....	35
5.1 Kesimpulan.....	35
5.2 Saran.....	36
DAFTAR PUSTAKA.....	37
LAMPIRAN 38	



DAFTAR GAMBAR

	Halaman
Gambar 2.1 Ternak Ayam.....	3
Gambar 2.2 Mesin Penetas Telur	4
Gambar 2.3 Telur Ayam	4
Gambar 2.4 Logo <i>Edge Impulse</i>	6
Gambar 2.5 Tampilan Website <i>Edge Impulse</i>	7
Gambar 2.6 Labeling Objek Deteksi FOMO	9
Gambar 2.7 Modul ESP32-CAM.....	10
Gambar 2.8 ESP32-CAM-MB	11
Gambar 2.9 Arduino IDE	11
Gambar 2.10 OLED 0.96” 128x64	13
Gambar 2.11 LED	13
Gambar 3.1 Satu Telur Retak	15
Gambar 3.2 Satu Telur Tidak Retak	15
Gambar 3.3 Dua Telur Retak Kanan	16
Gambar 3.4 Labeling Dataset.....	17
Gambar 3.5 Ekstraksi <i>Features</i>	17
Gambar 3.6 <i>Training Settings</i>	17
Gambar 3.7 Diagram Blok Sistem	18
Gambar 3.8 <i>Flowchart</i> Perancangan Sistem <i>Software</i>	19
Gambar 3.9 <i>Flowchart</i> Perancangan Sistem <i>Hardware</i>	20
Gambar 3.10 Perancangan Mekanik	21
Gambar 3.11 <i>Prototype</i> Penetasan Telur	21
Gambar 4.1 Model-model pada <i>Edge Impulse</i>	23
Gambar 4.2 Proses Pengujian Deteksi Objek dengan <i>Prototype</i>	27

DAFTAR TABEL

	Halaman
Tabel 2.1 Karakteristik OLED 0.96”	12
Tabel 3.1 Alat yang digunakan.....	14
Tabel 3.2 Bahan yang digunakan	14
Tabel 4.1 Hasil Validasi Proses <i>Training</i> Model YOLO	24
Tabel 4.2 Hasil Validasi Proses <i>Training</i> Model SSD.....	25
Tabel 4.3 Hasil Validasi Proses <i>Training</i> Model FOMO	25
Tabel 4.4 Hasil Pengujian Deteksi Satu Butir Telur Retak	28
Tabel 4.5 Hasil Pengujian Deteksi Satu Butir Telur Tidak Retak	28
Tabel 4.6 Hasil Deteksi Kondisi Telur 1 Retak dan Telur 2 Tidak Retak	29
Tabel 4.7 Hasil Deteksi Kondisi Telur 1 Tidak Retak dan Telur 2 Retak	29
Tabel 4.8 Hasil Deteksi Kondisi Telur 1 Retak dan Telur 2 Retak.....	30
Tabel 4.9 Hasil Deteksi Kondisi Telur 1 Tidak Retak dan Telur 2 Tidak Retak ...	30
Tabel 4.10 Kesimpulan Kondisi Telur 1 Retak dan Telur 2 Tidak Retak.....	31
Tabel 4.11 Kesimpulan Kondisi Telur 1 Tidak Retak dan Telur 2 Retak.....	31
Tabel 4.12 Kesimpulan Kondisi Telur 1 Retak dan Telur 2 Retak.....	32
Tabel 4.13 Kesimpulan Kondisi Telur 1 Tidak Retak dan Telur 2 Tidak Retak....	33
Tabel 4.14 Kesimpulan Pengujian Deteksi Dua Butir Telur	33

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Foto Alat	38
Lampiran 2 Tabel Hasil Pengujian Deteksi Satu Butir Telur Retak.....	39
Lampiran 3 Tabel Hasil Pengujian Deteksi Satu Butir Telur Tidak Retak.....	41
Lampiran 4 Tabel Hasil Deteksi Kondisi Telur 1 Retak dan Telur 2 Tidak Retak	43
Lampiran 5 Tabel Hasil Deteksi Kondisi Telur 1 Tidak Retak dan Telur 2 Retak	45
Lampiran 6 Tabel Hasil Deteksi Kondisi Telur 1 Retak dan Telur 2 Retak	47
Lampiran 7 Tabel Hasil Deteksi Kondisi Telur 1 Tidak Retak dan Telur 2 Tidak Retak	49
Lampiran 8 Kesimpulan Kondisi Telur 1 Retak dan Telur 2 Tidak Retak	51
Lampiran 9 Kesimpulan Kondisi Telur 1 Tidak Retak dan Telur 2 Retak	53
Lampiran 10 Kesimpulan Kondisi Telur 1 Retak dan Telur 2 Retak	55
Lampiran 11 Kesimpulan Kondisi Telur 1 Tidak Retak dan Telur 2 Tidak Retak	57
Lampiran 12 Program Pengambilan Dataset.....	59
Lampiran 13 Program Pengujian Deteksi Objek	60
Lampiran 14 Kartu Konsultasi Bimbingan Tugas Akhir	68
Lampiran 15 Bukti Originalitas	69
Lampiran 16 Biodata Penulis	75

BAB I

PENDAHULUAN

1.1 Latar Belakang

Peternakan adalah salah satu sektor penting yang terlibat dalam pemeliharaan hewan untuk menghasilkan sumber pangan, salah satunya adalah peternakan ayam. Dalam pengembangbiakan ayam dibutuhkan pengecekan berkala terhadap telur yang akan menetas. Salah satu masalah yang dihadapi adalah tidak diketahuinya kapan telur akan menetas. Anak ayam yang baru menetas perlu dipindahkan ke tempat lain dengan suhu yang lebih rendah sebagai pencegahan dini dari risiko kematian anak ayam.

Dari permasalahan tersebut diperlukan pemantauan otomatis sebagai penanda apabila ada telur yang menetas. Dibutuhkan teknologi yang dapat mendeteksi perubahan yang terjadi pada telur dengan cara mengintegrasikan mikrokontroler dengan teknologi AI (*Artificial Intelligence*). Harapannya setelah telur menetas, sistem akan memberi tanda agar anak ayam yang baru menetas segera dipindahkan ke tempat khusus seperti *brooder*. Meskipun teknologi ini memiliki potensi besar, terdapat beberapa tantangan dalam penerapannya, salah satunya adalah memastikan keakuratan deteksi pada perubahan bentuk saat penetasan terjadi.

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem deteksi perubahan telur yang menetas menggunakan mikrokontroler yang diintegrasikan dengan teknologi AI. Sistem ini dirancang untuk membantu peternak ayam dalam memantau telur ayam. Dengan sistem ini, diharapkan dapat mengurangi risiko kerugian bagi peternak ayam yang disebabkan oleh kelalaian peternak dalam merawat anak ayam.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka disampaikan rumusan masalah sebagai berikut:

1. Bagaimana sistem dapat mendeteksi adanya telur yang retak?
2. Bagaimana sistem dapat memberi tanda atau indikator bahwa ada telur yang retak?

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah, maka terdapat beberapa batasan masalah dalam laporan ini, antara lain:

1. Penelitian ini berkapasitas dua telur ayam yang telah dibersihkan.
2. Pengujian melibatkan dua kondisi yaitu telur tidak retak dan telur retak.
3. Jarak deteksi telur menggunakan mikrokontroler dibatasi pada 20 cm.
4. Resolusi kamera yang digunakan 240 px x 240 px.
5. Penelitian ini dilakukan dalam kondisi pencahayaan yang merata untuk menghindari gangguan dalam proses deteksi.

1.4 Tujuan

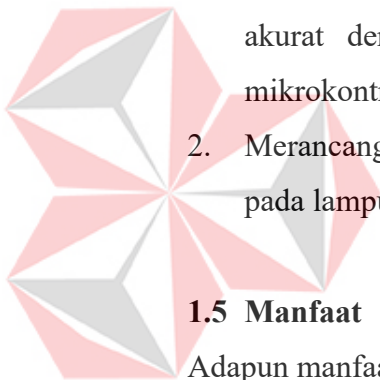
Adapun tujuan dari penelitian ini yaitu sebagai berikut:

1. Mengembangkan sistem yang dapat mendeteksi adanya telur yang retak secara akurat dengan menggunakan teknologi AI yang diintegrasikan dengan mikrokontroler.
2. Merancang sistem indikator untuk mendeteksi kondisi telur yang ditampilkan pada lampu LED.

1.5 Manfaat

Adapun manfaat dari penelitian ini, antara lain:

1. Membantu peternak ayam untuk segera menangani anak ayam yang baru menetas.
2. Mengurangi kerugian kematian anak ayam akibat kelalaian peternak yang tidak segera memindahkan anak ayam yang baru menetas.



BAB II LANDASAN TEORI

2.1 Ternak Ayam

Salah satu usaha peternakan yang ditampilkan pada Gambar 2.1 di bawah ini mempunyai hasil jual tinggi dan membantu untuk meningkatkan perekonomian masyarakat yaitu ternak ayam (Setyawan et al., 2024). Terdapat dua jenis ayam yang di ternakkan yaitu ayam pedaging (*broiler*) dan ayam petelur. Ternak ayam mempunyai peran penting dalam sumber protein hewani untuk masyarakat.



Gambar 2.1 Ternak Ayam

(Sumber: <https://kominfo.jatimprov.go.id/berita/peternak-ayam-petelur-perlu-memaksimalkan-tanggung-jawab-kesejahteraan-kandang/>)

Awal mula ternak ayam karena kebutuhan daging ayam dan juga telur terus meningkat (Parenreng et al., 2019). Oleh karena itu, banyak peternak ayam melakukan perawatan terhadap ayam agar bisa memproduksi lebih banyak ayam dan telur. Perawatannya meliputi pemberian pakan yang bergizi, menjaga kebersihan lingkungan kandang dan pemberian vaksin untuk mencegah penyakit. Selain itu, teknologi seperti mesin penetas telur juga semakin banyak digunakan untuk meningkatkan efisiensi produksi dan perkembangan ayam yang optimal.

2.2 Mesin Penetas Telur

Mesin penetas telur atau bisa disebut juga inkubator telur merupakan sebuah alat yang dirancang untuk meniru kondisi asli atau alami dimana telur ayam dierami oleh induk ayam agar menetas. Mesin ini menciptakan lingkungan yang optimal

bagi telur ayam, termasuk kelembapan, suhu dan rotasi, sehingga embrio telur dapat berkembang dengan baik.

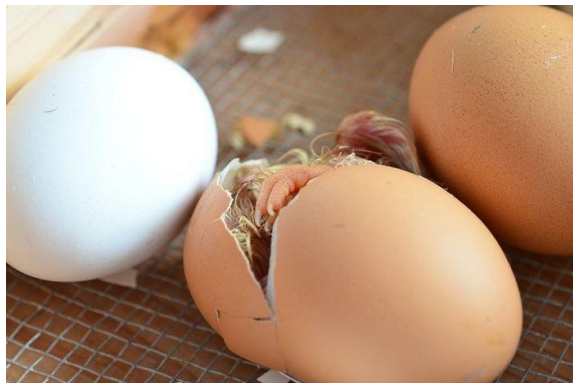
Di dalam mesin penetas telur seperti yang ditampilkan pada Gambar 2.2, terdapat lampu pijar yang digunakan untuk menghangatkan telur sehingga telur dapat menetas dengan baik (Nusyirwan et al., 2019). Lampu pijar ini menjaga suhu tetap stabil di dalam mesin penetas telur, biasanya antara 37,5°C sampai 38°C. Suhu ini optimal untuk perkembangan embrio di dalam telur.



Gambar 2.2 Mesin Penetas Telur

(Sumber: <https://www.majalahinforet.com/2018/03/mengenal-berbagai-macam-mesin-tetas.html/>)

Telur ayam yang layak ditetaskan, sebagaimana ditampilkan pada Gambar 2.3 memiliki ciri-ciri yaitu kulit telur yang sehat dan bersih dari kotoran, ukuran telur yang standar, warna cangkang cenderung agak gelap dan asal telur dari induk ayam yang sehat dan produktif.



Gambar 2.3 Telur Ayam

(Sumber: <https://bobo.grid.id/read/081738039/anak-ayam-bisa-keluar-dari-cangkang-telur-yang-keras-bagaimana-caranya?page=all>)

2.3 Artificial Intelligence

Artificial Intelligence (AI) atau kecerdasan buatan merupakan teknologi mesin atau komputer yang dapat melakukan pekerjaan yang dikerjakan dengan kecerdasan manusia. AI ini dirancang dengan melakukan tugas-tugas seperti belajar, memahami, memecahkan masalah dan mengambil keputusan.

Terdapat beberapa aplikasi pada berbagai sektor dari AI, termasuk ternak ayam. AI pada ternak ayam biasanya digunakan untuk pemantauan kesehatan ayam dengan memonitor perilaku pada ayam, yang kedua adalah otomatisasi pemberian makan dan minum dengan menjadwalkan pemberian makan dan minum, ketiga pengelolaan lingkungan kandang ayam seperti pengaturan suhu dan kelembapan, serta klasifikasi dan deteksi telur untuk memisahkan telur yang sudah menetas dan belum menetas.

Dengan kemajuan teknologi AI, industri peternakan ayam mengalami perubahan, dimana penggunaan AI memungkinkan peternak membuat keputusan yang lebih tepat dan berdasarkan data. Misalnya, teknik *computer vision* dan *machine learning* dapat digunakan untuk mendeteksi dan mengklasifikasikan gerakan dan postur ayam yang terkait dengan berbagai penyakit (Taleb et al., 2024). Hal ini menunjukkan bahwa penggunaan AI dalam sektor peternakan tidak hanya meningkatkan produktivitas dan efisiensi, tetapi juga berkontribusi terhadap kesejahteraan hewan, khususnya ayam.

2.3.1 Machine Learning

Machine Learning (ML) menjadi salah satu inovasi terbaru dalam teknologi informasi dan analisis data yang mengubah cara untuk memanfaatkan informasi dalam pengambilan keputusan. Karena ML memiliki kemampuan untuk memproses dan menganalisis data dalam jumlah besar, ML juga mengekstrak informasi berharga yang tidak diketahui manusia. Konsep dasar dari ML adalah bahwa sistem dapat “belajar” dari pengalaman, atau data yang dianalisis sebelumnya, untuk meningkatkan akurasi dan performa tertentu. Proses ini melibatkan algoritma yang dapat menyesuaikan model berdasarkan *feedback* dari hasil prediksi. Oleh karena itu, semakin banyak data yang diproses, semakin cerdas dan efisien sistem tersebut.

Machine Learning (ML) merupakan salah satu cabang dari *Artificial Intelligence* (AI) yang dapat menjadikan sistem komputer belajar dari data tanpa melakukan program eksplisit. Sistem ML dapat mengidentifikasi pola dari data, membuat prediksi dan mengambil keputusan berdasarkan data yang diberikan. ML dapat digunakan pada peternakan hewan ternak, salah satunya dapat memprediksi *output* yang bermanfaat seperti produksi telur (Neethirajan, 2020).

2.3.2 *Edge Impulse*

Edge Impulse adalah sebuah platform pengembangan AI khusus untuk perangkat IoT yang dapat memungkinkan pengguna membuat, melatih dan menyalurkan model *machine learning* langsung ke perangkat mikrokontroler. Platform ini dapat digunakan dengan berbagai sensor dan mikrokontroler, termasuk ESP32-CAM yang biasanya digunakan untuk pengenalan visual. Platform ini memungkinkan pengguna mengumpulkan data sensor, memproses data dan melatih model AI yang dapat melakukan tugas-tugas seperti deteksi objek, pengenalan suara dan klasifikasi data lainnya. Logo dari platform *Edge Impulse* dapat dilihat pada Gambar 2.4 berikut ini.



Gambar 2.4 Logo *Edge Impulse*
(Sumber: <https://edgeimpulse.com/branding/>)

Fitur-fitur utama yang ada pada platform *Edge Impulse* adalah sebagai berikut:

1. *Data Acquisition*

Mengumpulkan data sensor langsung dari perangkat yang terhubung untuk melatih model. Baik itu berupa gambar, audio, atau data sensor lainnya.

2. *Data Processing*

Menyediakan alat pengolah data bawaan, seperti konversi gambar ke *grayscale*, ekstraksi fitur audio dan analisis data sensor lain. Proses ini digunakan untuk memastikan bahwa data telah siap untuk dilatih.

3. *Model Training*

Menyediakan berbagai arsitektur model *machine learning*, seperti klasifikasi, *object detection* dan klasifikasi secara *real-time*. Dapat dilatih di *cloud interface* visual yang mudah digunakan.

4. *Impulse Design*

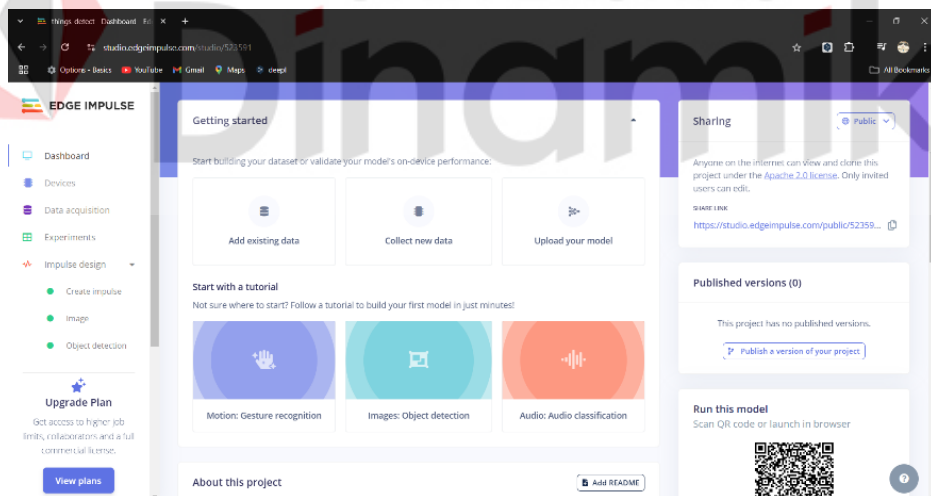
User dapat merancang *impulse*, yang berarti dapat memproses alur dari data mentah hingga prediksi akhir, mengatur blok pemrosesan dan secara opsional memodelkan blok pembelajaran.

5. *Live Classification*

User dapat melakukan klasifikasi secara langsung berdasarkan data yang diperoleh dari perangkat IoT, yang berguna untuk pengujian cepat dan demonstrasi langsung.

6. *Deployment Options*

Mendukung distribusi model secara langsung ke perangkat dengan format yang kompatibel, termasuk *output* dalam bentuk *library* Arduino, *Tensorflow Lite*, atau file *firmware* yang siap pakai.



Gambar 2.5 Tampilan Website *Edge Impulse*

Gambar 2.5 menampilkan tampilan website dari *Edge Impulse*. Platform ini memungkinkan pengembangan aplikasi *machine learning* (ML) untuk perangkat *edge*, termasuk deteksi objek. Berikut ini adalah proses dari deteksi objek pada *Edge Impulse*:

1. Pengumpulan data

Mengumpulkan data yang relevan, data ini berupa gambar yang berisi objek yang dikenali. Setiap gambar harus sudah diberi label dan memiliki kelas objek.

2. *Pre-processing* data

Setelah pengumpulan data, selanjutnya *pre-processing* data, ini meliputi perubahan ukuran gambar, normalisasi dan augmentasi data untuk meningkatkan keragaman dataset.

3. Membuat proyek

User membuat proyek baru pada platform *Edge Impulse*, ini memungkinkan *user* dapat memilih jenis proyek yang sesuai, Seperti deteksi objek.

4. *Training* model

Setelah menyiapkan data, *user* dapat melatih model deteksi objek. *Edge Impulse* menyediakan berbagai algoritma yang dapat dipilih, termasuk FOMO (*Faster Object Detection for Mobile Oriented*), YOLO (*You Only Look Once*) dan SSD (*Single Shot Detector*). Hal ini dapat membuat *user* mengatur parameter pelatihan, seperti jumlah *Epoch*, ukuran *batch* dan *Learning Rate*.

5. Mengevaluasi model

Setelah *training* model, model dievaluasi berdasarkan performanya. Metrik yang digunakan untuk evaluasi deteksi objek adalah *Recall*, *Precision*, *F1-Score* dan *mean Average Precision (mAP)*.

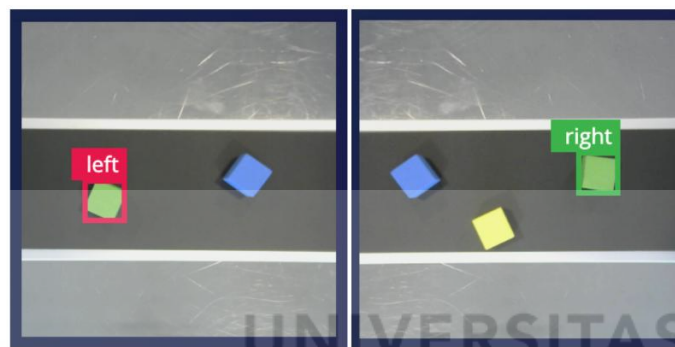
6. Mengimplementasikan pada perangkat *edge*

Selanjutnya menerapkan model yang telah di evaluasi tersebut ke perangkat *edge* dan melakukan pengujian secara langsung dalam kondisi *real-time*.

Deteksi objek menggunakan *Edge Impulse* memungkinkan pengembang untuk membangun dan mengimplementasikan deteksi objek secara efisien. Selain itu, platform ini mempermudah proses deployment model ke berbagai perangkat *edge*, sehingga dapat digunakan dalam aplikasi lain secara luas.

2.3.3 FOMO

FOMO (*Faster Object Detection for Mobile Oriented*) atau dalam website *Edge Impulse* adalah arsitektur *Deep Learning* yang dirancang untuk mendeteksi objek pada perangkat dengan sumber daya terbatas. Tidak seperti model deteksi lain yang memerlukan GPU atau TPU agar berjalan secara optimal, FOMO dioptimalkan untuk berjalan pada perangkat mikroprosesor seperti ARM Cortex-M atau ESP32. Hal ini dilakukan untuk meminimalkan jumlah operasi matematis dan parameter dalam model, sehingga secara relevan mengurangi latensi dan konsumsi daya.



Gambar 2.6 Labeling Objek Deteksi FOMO
(Sumber: <https://www.edgeimpulse.com/blog/fomo-self-attention/>)

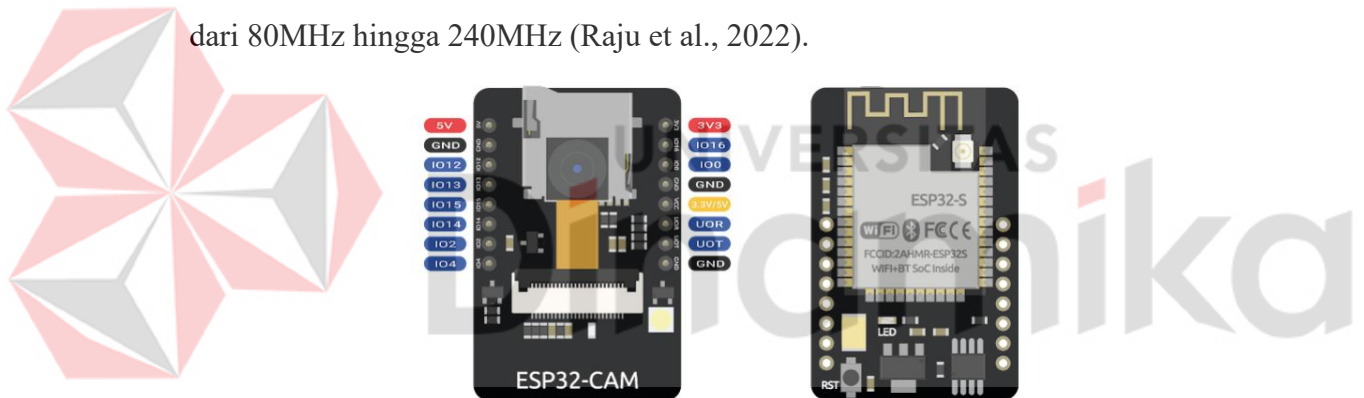
Meskipun FOMO lebih terkenal untuk deteksi objek berbasis *single-class*, model ini juga dapat diperluas untuk mendeteksi beberapa kelas objek. Namun performa *multi-class* FOMO bergantung pada penempatan *grid* dan pelabelan data pelatihan yang akurat. Contoh pelabelan dapat dilihat pada Gambar 2.6 di atas. Model ini memastikan model tetap efisien dalam membantu berbagai jenis aplikasi tanpa mengorbankan kecepatan proses. Model FOMO juga menggunakan *output grid* yang hanya mencatat probabilitas keberadaan objek tanpa memproses atribut seperti ukuran atau orientasi *bounding box*. Hal ini mengurangi kompleksitas pemrosesan tetapi tetap memberikan informasi lokasi objek yang cukup.

Dalam IoT, FOMO digunakan dalam aplikasi yang memerlukan respon yang cepat, seperti deteksi gerakan dan pemantauan produksi. Kombinasi sumber daya dan pemantauan secara *real-time* ini menjadikannya lebih ideal untuk *edge AI*. Model FOMO lebih ringan dibandingkan model-model lainnya karena tidak memerlukan regresi *bounding box* dan hanya memproses *keypoints*. FOMO

termasuk bagian dari ekosistem TinyML (*Tiny Machine Learning*) yang bertujuan untuk menghadirkan kemampuan AI pada perangkat dengan sumber daya terbatas. Membantu perangkat IoT menjadi perangkat pintar dengan kemampuan analitik yang canggih tanpa memerlukan koneksi server eksternal.

2.4 ESP32-CAM

ESP32-CAM dapat digunakan sebagai modul kamera dengan ukuran kecil dan konsumsi daya yang rendah. Gambar 2.7 menunjukkan modul ESP32-CAM yang dilengkapi dengan kamera OV2640 dan slot untuk kartu TF yang mendukung penyimpanan data. ESP32-CAM digunakan umum dalam aplikasi IoT seperti CCTV, Identifikasi QR, klasifikasi dan sebagainya. Modul ini dapat diintegrasikan dengan Wi-Fi, Bluetooth dan BLE Beacon dengan sepasang CPU LX6 32-bit yang unggul. Modul ini juga terdapat penyesuaian frekuensi yang bervariasi dari 80MHz hingga 240MHz (Raju et al., 2022).



Gambar 2.7 Modul ESP32-CAM

(Sumber: <https://www.arduino.biz.id/2022/08/development-board-esp32-cam.html>)

ESP32-CAM dapat di program menggunakan board CH340 atau ESP32-CAM-MB maupun FTDI yang digunakan sebagai *interface* USB to serial. ESP32-CAM-MB atau *Downloader* ESP32-CAM Programmer CH340 Development Board yang dapat dilihat pada Gambar 2.8 merupakan modul yang memudahkan pengguna untuk memprogram dan mengatur ESP32-CAM tanpa memerlukan rangkaian kabel tambahan. Modul ini digunakan untuk komunikasi *interface* antara komputer dan ESP32-CAM, yang memungkinkan *user* dapat mengubah kode dan melakukan *debugging* secara langsung.



Gambar 2.8 ESP32-CAM-MB

(Sumber: https://www.espboards.dev/blog/flash-any-esp32-with-esp32-cam-mb/#google_vignette)

ESP32-CAM ini juga dapat terhubung dengan IC Programmer FTDI yang berfungsi sebagai USB to TTL. Menggunakan bahasa pemrograman C++ untuk mengidentifikasi ESP32-CAM menginisialisasi port FTDI dan Port ESP32-CAM hingga proses transfer data dan kompilasi data berhasil. ESP32-CAM memiliki 16 pin yang digunakan untuk *Input* dan *Output*. Setiap GPIO pada ESP32-CAM memiliki fungsi Data, CLK, CMD, CSI_MCLK, TX dan RX (Adi & Wahyu, 2022).

2.5 Arduino IDE

Arduino IDE merupakan *software* yang digunakan untuk mengedit, menulis, mengkompilasi dan mengunggah kode ke mikrokontroler. IDE dengan singkatan *Integrated Development Environment* mempunyai lingkungan pengembangan yang mudah digunakan. Di bawah ini merupakan Gambar 2.9 Logo dari Arduino IDE.



Gambar 2.9 Arduino IDE

(Sumber: <https://www.arduino.cc/en/software>)

Software ini dirancang dengan *interface* yang sederhana dan minimalis, memudahkan bagi pemula yang baru memulai belajar, serta terdapat banyak *library* yang tersedia. Arduino IDE digunakan dalam berbagai proyek *embedded systems*, seperti otomasi rumah, robotika dan proyek IoT.

2.6 OLED 0.96" 128x64

OLED 0.96" ini merupakan modul layar kecil dengan teknologi *Organic Light Emitting Diode* yang menampilkan gambar 2D 128x64 dan populer di banyak aplikasi kelistrikan. Tampilan OLED ini dapat di kontrol secara terpisah untuk menampilkan grafik, teks dan animasi. Karena setiap piksel dari OLED ini menghasilkan cahayanya sendiri, layar OLED ini dapat menampilkan warna hitam yang pekat dan rasio kontras tinggi tanpa memerlukan lampu latar (Qassim A. Al-Jarwany, 2024). Karakteristik utama OLED 0.96" 128x64 ditunjukkan pada Tabel 2.1 berikut.

Tabel 2.1 Karakteristik OLED 0.96"

No.	Properti	Deskripsi
1.	Resolusi	128x64 piksel
2.	Interface	Biasanya I2C, SPI, atau paralel 8-bit
3.	Area Aktif	Sekitar 0.96 inci secara diagonal
4.	IC Penggerak	Chip pengontrol SSD1306 atau serupa
5.	Tegangan Operasi	3.3 V atau 5 V (tergantung model)
6.	Sudut Pandang	Hingga 160 derajat
7.	Suhu Operasi	-40°C hingga 70°C
8.	Kecepatan Komunikasi	Tergantung <i>interface</i> (400kHz untuk I2C, lebih tinggi untuk SPI)

Kelebihan dari OLED ini untuk ESP32-CAM adalah konsumsi daya rendah karena hanya menyalakan piksel yang aktif, kemudian menampilkan kualitas yang tinggi, serta ringkas dan mudah untuk dipasang. Tampilan dari OLED dapat dilihat pada Gambar 2.10 berikut.

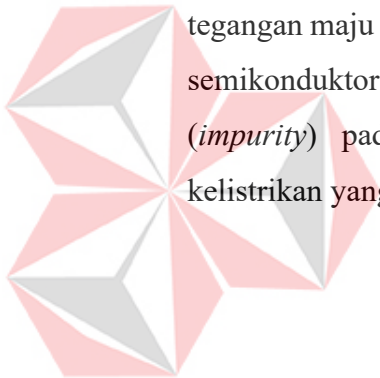


Gambar 2.10 OLED 0.96" 128x64

(Sumber: <https://electronicslovers.com/2019/01/interacting-with-an-oled-display-via-an-i2c-bus-system.html>)

2.7 Lampu LED (*Light Emitting Diode*)

Lampu LED merupakan perangkat elektronik yang menghasilkan cahaya ketika dialiri arus listrik. LED adalah jenis dioda yang terbuat dari bahan semikonduktor. Cara kerja LED adalah memancarkan cahaya hanya ketika dialiri tegangan maju (*bias forward*) diberikan dari anoda ke katoda. LED terdiri dari chip semikonduktor yang dimana semikonduktor akan ditambahkan ketidakmurnian (*impurity*) pada semikonduktor murni sehingga menghasilkan karakteristik kelistrikan yang diinginkan (Syafira et al., 2022).



Gambar 2.11 LED

(Sumber: <https://www.belajaronline.net/2020/09/pengertian-led-light-emitting-diode-dan-fungsi.html>)

Gambar 2.11 menampilkan lampu LED yang digunakan dalam penelitian ini, dipilih berdasarkan warna yang jelas, yaitu merah dan hijau. LED ini dapat menyala dan mati secara instan dan memberikan respon yang cepat serta minim konsumsi daya, sehingga sangat cocok untuk perangkat IoT yang memerlukan visual cepat dan efisien.

BAB III METODOLOGI PENELITIAN

3.1 Studi Literatur

Pada tahap ini melakukan pengumpulan dan meninjau berbagai sumber informasi terkait topik dan judul penelitian yang akan dilaksanakan. Mencari dan mempelajari literatur dan informasi terkini, terutama informasi terkait topik penelitian dalam lima tahun terakhir. Hal ini dilakukan untuk mendapatkan wawasan yang mendalam tentang dasar teori dan variabel yang menjadi fokus penelitian serta memilih metode dan algoritma yang tepat untuk diterapkan dalam penelitian ini.

3.2 Alat dan Bahan

Terdapat beberapa alat pada Tabel 3.1 dan bahan pada Tabel 3.2 yang diperlukan dalam penelitian ini, yaitu sebagai berikut:

Tabel 3.1 Alat yang digunakan

No.	Alat	Deskripsi
1.	ESP32-CAM	Sebagai alat utama untuk deteksi gambar telur dan aplikasi model AI.
2.	Kamera OV2640	Terpasang pada ESP32-CAM untuk menangkap gambar telur.
3.	LED	Untuk menampilkan hasil deteksi. LED hijau untuk penanda telur retak kiri dan LED merah untuk telur retak kanan.
4.	Laptop	Digunakan untuk memprogram, melatih model, pengujian, dan pengembangan kode menggunakan <i>Edge Impulse</i> dan Arduino IDE.
5.	Kabel Jumper	Digunakan untuk menghubungkan ESP32-CAM dengan komponen LED.
6.	<i>Power Supply</i> atau Adaptor 5 Volt	Memberikan aliran listrik pada ESP32-CAM dan komponen lainnya.
7.	Lampu	Untuk menerangi ruangan pada proses pendeteksian telur.

Tabel 3.2 Bahan yang digunakan

No.	Bahan	Deskripsi
1.	Telur Ayam	Digunakan untuk pengujian deteksi keretakan.
2.	Platform <i>Edge Impulse</i>	Digunakan untuk melatih model dan mengoptimalkan model AI dalam pendeteksian keretakan telur.

No.	Bahan	Deskripsi
3.	Resistor	Untuk arus LED agar tidak rusak.
4.	<i>Edge Impulse</i> Model AI	Model deteksi objek dari <i>Edge Impulse</i> yang telah dilatih khusus untuk mendeteksi keretakan telur.
5.	<i>Software</i> Arduino IDE	Untuk memprogram dan mengintegrasikan model dengan ESP32-CAM.
6.	Kaca dan Kayu	Sebagai bahan dasar untuk struktur fisik, seperti tata letak komponen dan tempat untuk melindungi perangkat yang digunakan serta membuat sistem berjalan dengan stabil.

3.3 Pengambilan Dataset

Dataset pada penelitian ini dikumpulkan secara pribadi menggunakan mikrokontroler ESP32-CAM. Dataset ini terdiri dari satu butir telur dan dua butir telur untuk penelitian. Gambar 3.1 menunjukkan satu butir telur yang retak dan Gambar 3.2 menunjukkan satu butir telur yang tidak retak.



Gambar 3.1 Satu Telur Retak



Gambar 3.2 Satu Telur Tidak Retak

Setiap kondisi pada pengambilan gambar dilakukan dengan pencahayaan yang stabil. Prosedur pengambilan gambar dataset dilakukan dalam dua kondisi untuk satu telur, yaitu telur retak dan telur tidak retak. Selain itu, gambar juga

diambil dalam empat kondisi pada dua telur, seperti Gambar 3.3 yang menampilkan kondisi dua telur retak di posisi kanan. Keempat kondisi tersebut adalah:

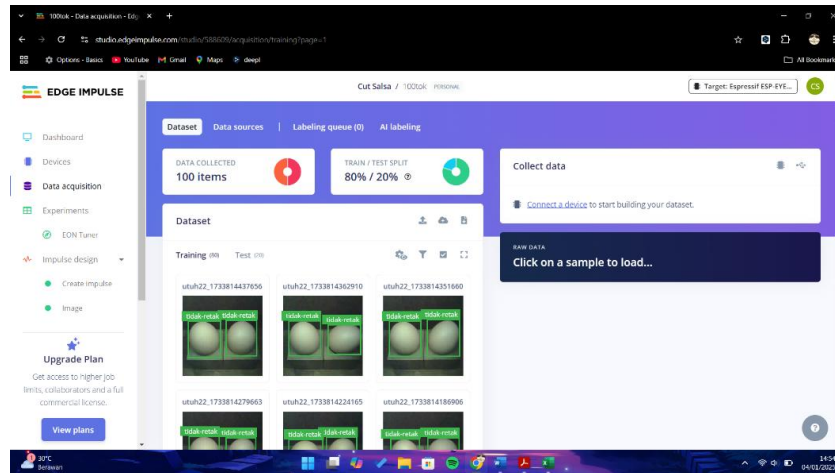
1. Telur 1 retak, telur 2 tidak retak
2. Telur 1 tidak retak, telur 2 retak
3. Telur 1 retak, telur 2 retak (Kedua telur retak)
4. Telur 1 tidak retak, telur 2 tidak retak (Kedua telur tidak retak)



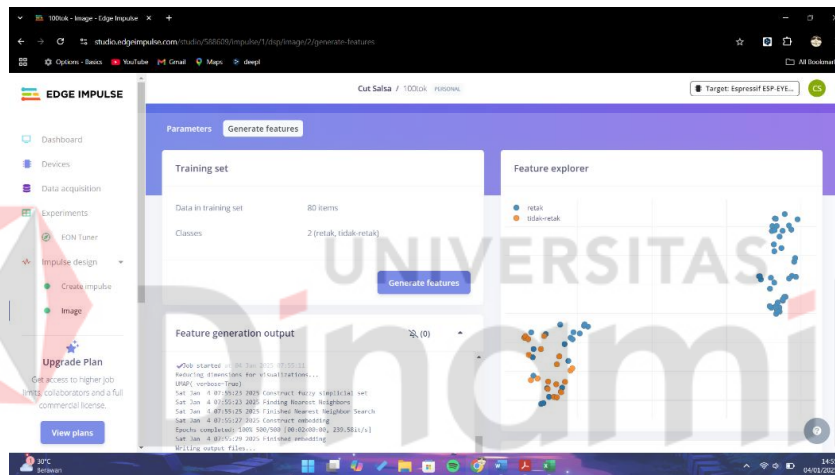
Gambar 3.3 Dua Telur Retak Kanan

3.4 *Pre-Processing* Dataset

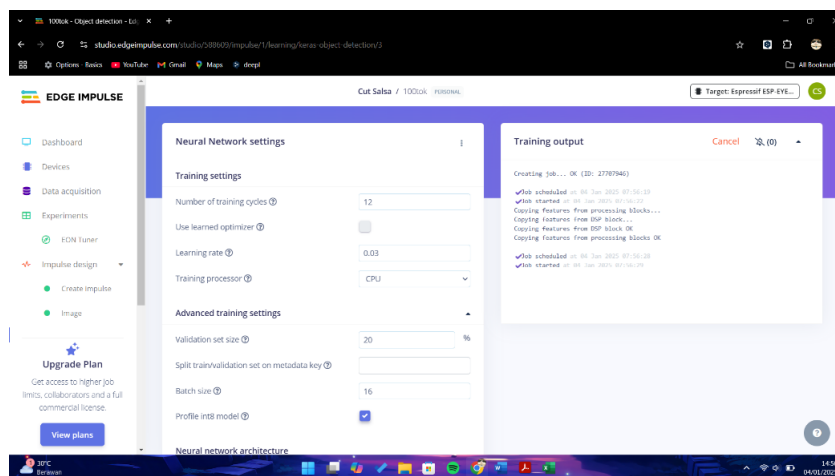
Pada *Pre-Processing* dataset ini dapat dilihat pada Gambar 3.4, yang dilakukan pada platform *Edge Impulse* dimulai dengan membuat sebuah proyek baru dan diberi nama proyek tersebut. Kemudian memasukan semua dataset gambar yang dilabeli dengan dua kelas yaitu retak dan tidak-retak. Setelah melabeli dataset gambar, berikutnya melakukan pemilihan *Impulse* atau proyek dan menginputkan ukuran piksel gambar yang digunakan, pada penelitian Tugas Akhir ini menggunakan *object detection* (deteksi objek). Setelah menyimpan *Impulse*, selanjutnya melakukan *Generate Features* untuk mengekstraksi gambar menjadi representasi histogram, seperti Gambar 3.5 hasil dari ekstraksi gambar.



Gambar 3.4 Labeling Dataset



Gambar 3.5 Ekstraksi Features



Gambar 3.6 Training Settings

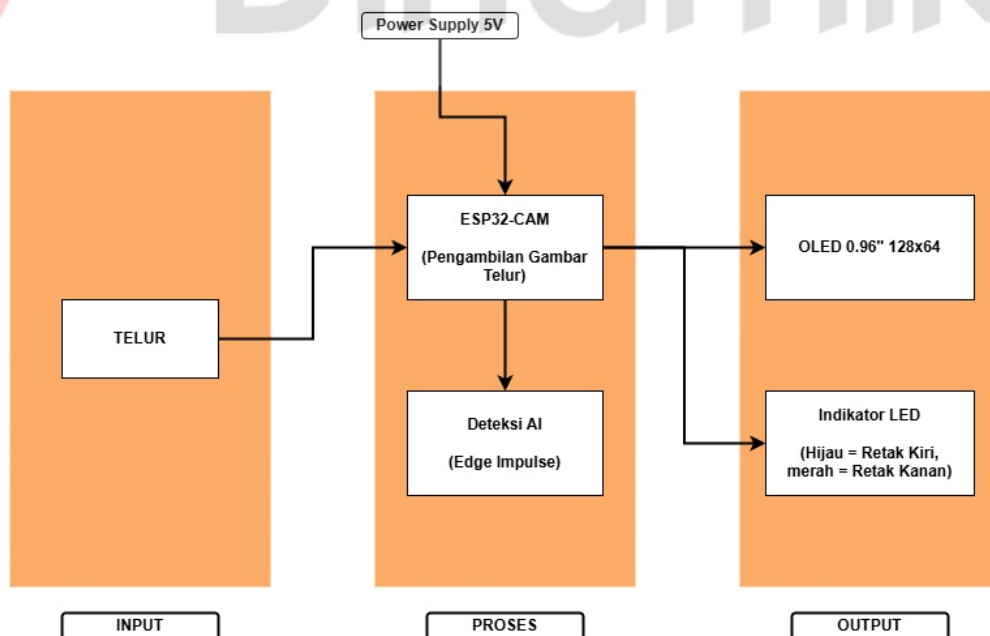
Dataset yang telah dilabeli dan dibuat *impulse*, secara otomatis menjadi data *training* dan data *validation*. Kemudian melakukan *training* dengan mengatur *training setting* dan akan menghasilkan validasi untuk mengetahui hasil *Precision* atau akurasi dari model yang telah dilatih. Tampilan dari pengaturan untuk *training* dapat dilihat pada Gambar 3.6 *Training Settings* pada *Edge Impulse*.

3.5 Desain Sistem

Pada desain sistem ini, langkah yang dilakukan adalah merancang sistem secara keseluruhan, berikut langkah-langkah yang dilakukan.

3.5.1 Diagram Blok Sistem

Pada Gambar 3.7 diagram blok sistem di bawah ini, terdapat 1 buah *power supply* dengan rincian 5 V ini digunakan untuk men-supply ESP32-CAM. Kemudian terdapat *input* telur untuk dideteksi keretakannya (berfungsi sebagai *input* utama). Selanjutnya pada tahap proses, terdapat ESP32-CAM yang digunakan untuk mengambil gambar telur dan mengirimkan gambar tersebut ke model AI untuk dianalisis. Gambar di proses oleh model Deteksi AI yang telah dilatih menggunakan *Edge Impulse*.

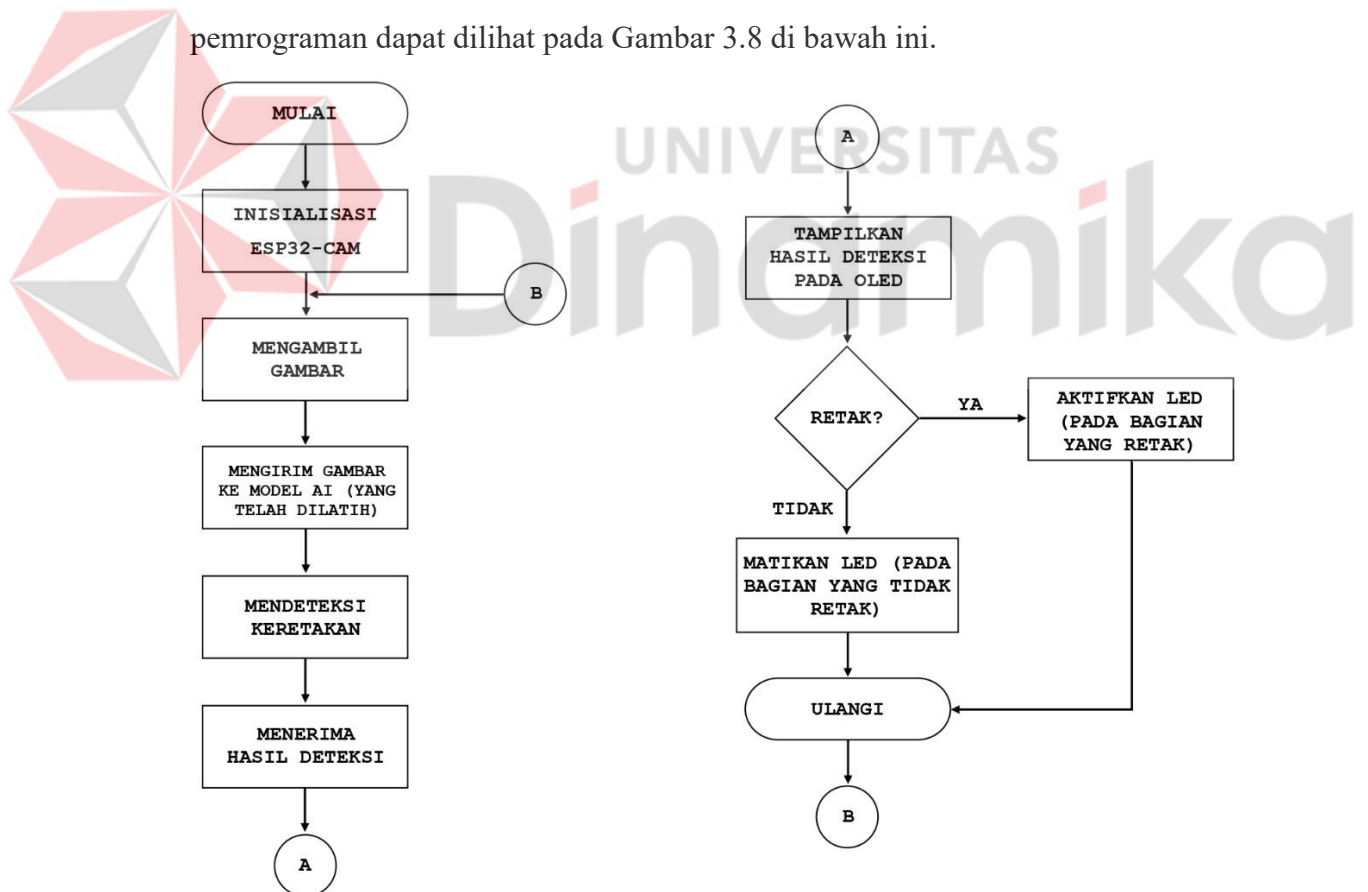


Gambar 3.7 Diagram Blok Sistem

Model Deteksi AI ini yang akan menentukan apakah telur tidak retak atau retak. Selanjutnya terdapat *output* yaitu LED sebagai indikator dari hasil deteksi model AI. LED hijau akan menyala jika telur retak di posisi kiri dan LED merah menyala jika ada telur yang retak di posisi kanan. Selain itu, hasil pendeteksian akan ditampilkan di layar OLED sehingga memberikan informasi visual yang lebih jelas mengenai kondisi telur.

3.5.2 Perancangan *Software*

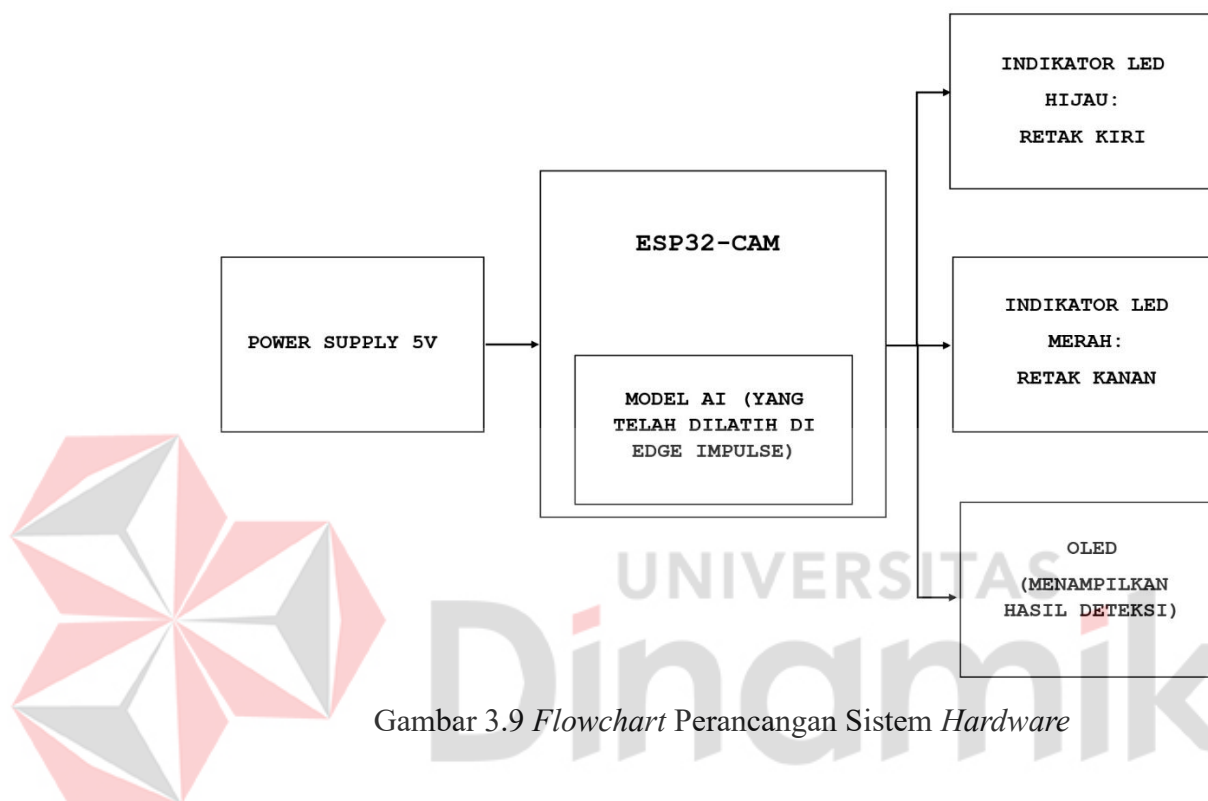
Perancangan *software* merupakan tahap perancangan yang penting dalam proses sistem. Perancangan ini meliputi analisis kebutuhan yang menunjukkan bahwa perangkat mampu mendeteksi keretakan pada telur dan memberikan penanda melalui LED. Desain arsitektur untuk menentukan alur dari pengambilan gambar, pengolahan data dan juga *output* (LED dan OLED). Perancangan pemrograman dapat dilihat pada Gambar 3.8 di bawah ini.



Gambar 3.8 *Flowchart* Perancangan Sistem *Software*

3.5.3 Perancangan *Hardware*

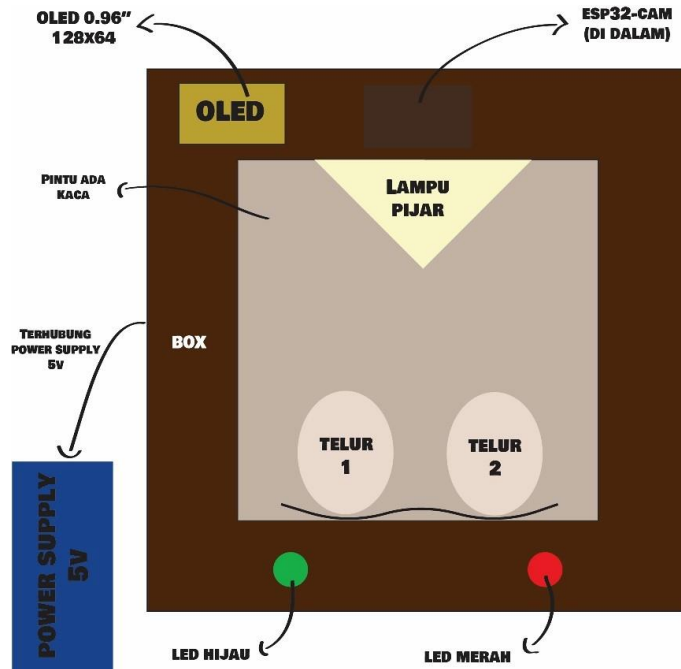
Perancangan *hardware* merupakan perancangan benda fisik yang penting bagi sistem. Perancangan perangkat keras sangat penting karena dapat mengurangi masalah dalam pembuatan desain *hardware*. Berikut perancangan *hardware* dapat dilihat dalam bentuk diagram blok rangkaian pada Gambar 3.9 berikut.



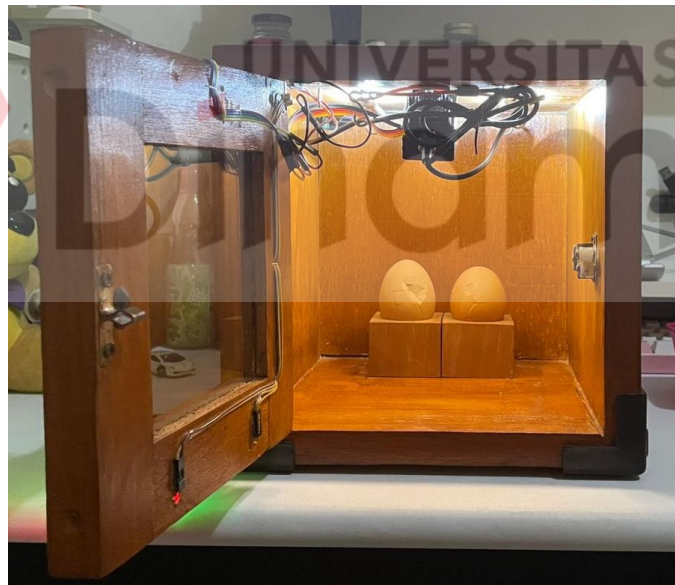
Gambar 3.9 *Flowchart* Perancangan Sistem *Hardware*

3.5.4 Perancangan Mekanik

Gambar 3.10 di bawah ini adalah gambar perancangan mekanik dari tempat untuk proses pendeteksian telur. Menggunakan kayu dengan panjang 25 cm, lebar 25 cm dan tinggi 25cm. Deteksi menggunakan mikrokontroler ESP32-CAM yang sudah tergabung dengan kamera OV2640. Ukuran dari tempat tersebut, dapat menampung 2 butir telur yang siap untuk dideteksi. Hasil deteksi akan ditampilkan pada layar OLED 0,96”, LED merah menyala jika ada telur retak pada posisi kanan dan LED hijau menyala jika ada telur retak pada posisi kiri. Serta lampu pijar yang digunakan untuk menerangi ruangan agar proses deteksi lebih optimal. Dapat dilihat pada Gambar 3.11 merupakan *Prototype* dari perancangan mekanik.



Gambar 3.10 Perancangan Mekanik

Gambar 3.11 *Prototype* Penetasan Telur

3.6 Metode Deteksi FOMO

Metode deteksi FOMO (*Fast Object Detection for Mobile Oriented*) pertama yang dilakukan adalah menyiapkan dataset gambar termasuk dalam menormalisasi dan penyesuaian ukuran gambar yang sesuai dengan kebutuhan model. Kemudian gambar akan dipecah menjadi *grid* kecil-kecil untuk membagi area deteksi, setiap

grid ini memiliki tugas untuk memprediksi keadaan objek dengan menghasilkan *bounding box* dan kelas objek. Proses ini dibantu oleh ekstraksi fitur melalui *layer convolution* untuk menangkap pola dalam gambar. *Output* yang dihasilkan adalah *bounding box* dengan nilai akurasi yang digunakan untuk menentukan keberadaan dan posisi objek dalam gambar.

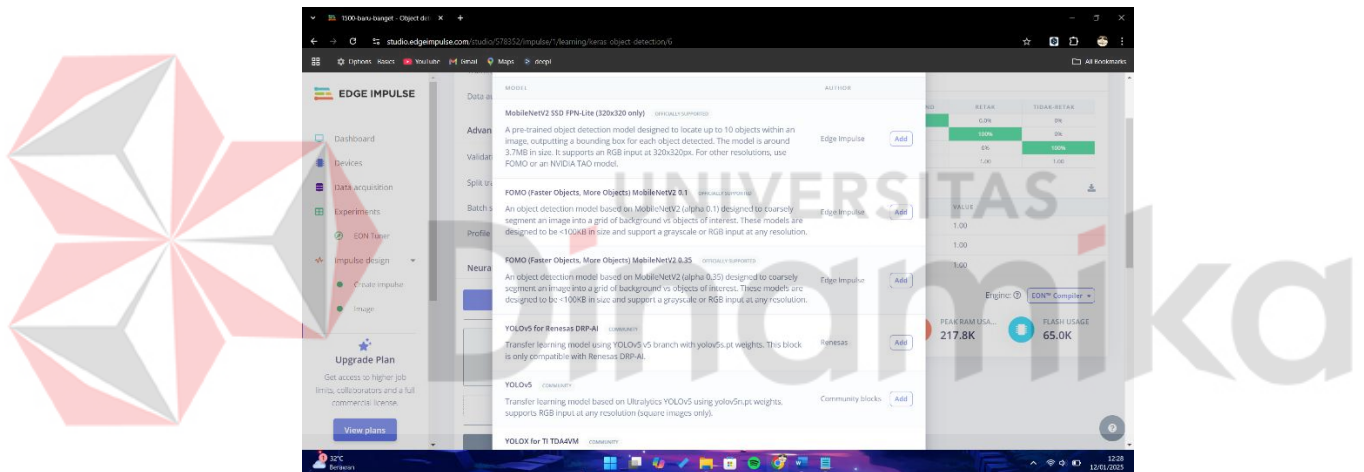


UNIVERSITAS
Dinamika

BAB IV HASIL DAN PEMBAHASAN

4.1 Validasi Proses *Training Data*

Validasi proses *training data* dalam penelitian ini digunakan untuk mengetahui performa dari model-model AI yang akan dilatih. Pada penelitian ini terdapat beberapa model *machine learning* yang akan di cari performa terbaiknya yaitu YOLO (*You Only Look Once*), SSD (*Single Shot Detector*) dan FOMO (*Fast Object Detection for Mobile Oriented*). Alat dan Bahan yang digunakan untuk proses validasi ini yaitu Laptop, platform *Edge Impulse* dan model-model AI yang ada di *Edge Impulse*. Gambar 4.1 merupakan model-model yang ada pada *Edge Impulse*.



Gambar 4.1 Model-model pada *Edge Impulse*

4.1.1 Langkah Validasi Proses *Training Data*

Langkah validasi proses *training data* ini dilakukan pada platform *Edge Impulse*. Tujuannya adalah untuk memastikan bahwa model yang dilatih memiliki performa yang baik bahkan pada data yang belum pernah ada sebelumnya. Berikut Langkah-langkahnya:

1. *Data Splitting* (Pemisahan Data)

Dataset yang diunggah dan sudah dilabeli akan di pisah menjadi *Training Set* dan *Validation Set*.

2. *Training Model*

Pada proses ini melibatkan optimasi parameter model dalam data *training set*.

3. Evaluasi Pada *Validation Set*

Setelah *training* selesai, model akan diuji dengan *validation set* dan akan memberikan hasil metrik (*Precision*, *Recall*, *mAP* dan *F1-Score*).

4.1.2 Hasil Validasi Proses *Training Data*

Proses dari *training data* menghasilkan tiga model *machine learning* dengan *Epoch* dan *Learning Rate* yang berbeda-beda. Berikut adalah hasil untuk masing-masing model.

A. YOLO

Metrik dari model YOLO ini terdapat *Precision*, *mAP* (*mean Average Precision*) dan *Recall*. *Precision* dan *Recall* menunjukkan informasi detail tentang keberhasilan proses deteksi. Serta *mAP* adalah gabungan dari informasi *Precision* dan *Recall*.

Tabel 4.1 Hasil Validasi Proses *Training Model YOLO*

Epoch	Learning Rate	Precision	Recall	mAP
12	0.1	44.1%	0.12	0.06
	0.02	54.0%	0.20	0.12
	0.003	50.2%	0.11	0.03
9	0.1	54.2%	0.21	0.15
	0.02	52.0%	0.12	0.04
	0.003	53.4%	0.12	0.09
6	0.1	57.0%	0.42	0.27
	0.02	40.7%	0.24	0.09
	0.003	50.3%	0.18	0.09
3	0.1	25.8%	0.11	0.04
	0.02	51.1%	0.13	0.06
	0.003	42.9%	0.12	0.09

Pada Tabel 4.1 di atas menampilkan hasil dari proses validasi *training dataset* dengan menggunakan model YOLO. Hasil menunjukkan bahwa *Learning Rate* berpengaruh dalam kinerja model. *Learning Rate* 0.1 memberikan hasil *mAP* terbaik pada *Epoch* 6 dengan nilai 0.27, sedangkan pada *Epoch* lainnya lebih rendah. Pada *Epoch* 6 memberikan hasil kinerja terbaik secara menyeluruh, dengan nilai *Precision* mencapai 57.0%, *Recall* 0.42 dan *mAP* 0.27. Model YOLO dalam proses ini belum optimal karena nilai *mAP* masih tergolong rendah.

B. SSD

Metrik untuk model SSD ini adalah *Precision*, *Recall* dan *mAP*. *Precision* dan *Recall* memberikan informasi rinci tentang keberhasilan model dalam mendeteksi, serta *mAP* merupakan gabungan informasi *Precision* dan *Recall*.

Tabel 4.2 Hasil Validasi Proses *Training* Model SSD

Epoch	Learning Rate	Precision	Recall	mAP
12	0.1	60.5%	0.64	0.50
	0.02	66.5%	0.64	0.55
	0.003	50.3%	0.54	0.36
9	0.1	52.8%	0.58	0.40
	0.02	55.5%	0.57	0.45
	0.003	48.1%	0.45	0.33
6	0.1	39.8%	0.47	0.27
	0.02	54.1%	0.62	0.41
	0.003	45.8%	0.51	0.36
3	0.1	36.1%	0.37	0.25
	0.02	49.0%	0.60	0.39
	0.003	4.6%	0.05	0.04

Berdasarkan Tabel 4.2 di atas, hasil validasi proses *training* model SSD memiliki nilai *mAP* tinggi yang dicapai pada *Epoch* 12 dengan *Learning Rate* 0.02 yaitu 0.55, *Precision* 66.5% dan *Recall* 0.64. Model SSD ini lebih baik dari model YOLO jika dibandingkan nilai *mAP* (nilai tertinggi SSD 0.55 sedangkan YOLO 0.27). *Learning Rate* 0.02 pada *Epoch* 12 adalah nilai terbaik untuk model SSD ini.

C. FOMO

Metrik pada model FOMO ini terdapat *Precision*, *Recall* dan *F1-Score*. *Precision* dalam FOMO menunjukkan bahwa ini mendeteksi objek tertentu. *Recall* untuk mengukur kemampuan model dalam hal mendeteksi dan berapa banyak yang berhasil dideteksi oleh model. *F1-Score* adalah rata-rata dari *Precision* dan *Recall* digunakan untuk keakuratan deteksi dengan menyeimbangkan *Precision* dan *Recall*.

Tabel 4.3 Hasil Validasi Proses *Training* Model FOMO

Epoch	Learning Rate	Precision	Recall	F1-Score
12	0.1	0.69	0.63	0.66
	0.02	0.71	0.69	0.70
	0.003	0.69	0.69	0.69
9	0.1	0.52	0.47	0.49
	0.02	0.59	0.81	0.68
	0.003	0.79	0.84	0.82

Epoch	Learning Rate	Precision	Recall	F1-Score
6	0.1	0.35	0.47	0.40
	0.02	0.59	0.81	0.68
	0.003	0.75	0.75	0.75
3	0.1	0.27	0.50	0.35
	0.02	0.50	0.50	0.50
	0.003	0.47	0.44	0.45

Dari Tabel 4.3 hasil dari validasi *training* model FOMO, menunjukkan bahwa model ini memiliki performa terbaik pada *Epoch* ke 9 dengan *Learning Rate* 0.003. *Precision* mencapai 0.79, *Recall* 0.84 dan *F1-Score* mencapai 0.82. Kombinasi *Precision* dan *Recall* yang tinggi menunjukkan bahwa model dapat mendeteksi objek dengan akurat.

Setelah validasi proses *training* tiga model yaitu YOLO, SSD dan FOMO, Diperoleh performa dari tabel-tabel di atas. YOLO memiliki performa yang lebih rendah karena mAP (0.27) ini membuat model kurang akurat dan tidak cocok untuk pengujian. SSD memiliki mAP yang lebih tinggi (0.55), tetapi FOMO lebih baik dari sisi *F1-Score* (0.82) sehingga dapat mendeteksi lebih maksimal. FOMO memiliki keseimbangan antara *Precision* dan *Recall*, serta model ini lebih ringan dibanding YOLO atau SSD.

4.2 Pengujian Proses Deteksi Objek

Pada pengujian ini dilakukan evaluasi tingkat akurasi dan presisi dari model dalam mendeteksi objek telur ayam. Setelah proses *training* selesai, proses evaluasi ini bertujuan untuk memastikan bahwa model dapat mengenali dan mengklasifikasikan objek dengan benar dan sesuai dengan data yang telah dilatihkan. Alat yang digunakan untuk proses pengujian deteksi objek yaitu *Prototype* Tempat Penetasan Telur, Telur Ayam dan Mikrokontroler ESP32-CAM. Pada Gambar 4.2 di bawah ini merupakan proses pengujian deteksi objek dengan *Prototype*.



Gambar 4.2 Proses Pengujian Deteksi Objek dengan *Prototype*

4.2.1 Langkah Pengujian Proses Deteksi Objek

Proses pengujian deteksi objek ini dilakukan pada *Prototype* Penetasan Telur. Tujuannya adalah untuk mengevaluasi kinerja model dalam mendeteksi telur dengan dua kategori yaitu mendeteksi satu butir telur dan dua butir telur dalam keadaan sebenarnya. Untuk satu butir telur mendeteksi telur yang tidak retak dan telur yang retak. Kemudian untuk mendeteksi dua butir telur mendeteksi berbagai kombinasi telur retak dan telur tidak retak.

Berikut langkah-langkahnya:

1. Menghidupkan *Prototype* dan menghubungkan mikrokontroler ESP32-CAM dengan adaptor.
2. Meletakkan telur ayam pada *Prototype* yang sesuai dengan area deteksi kamera ESP32-CAM.
3. Mencatat hasil nilai akurasi dari deteksi yang dilakukan oleh mikrokontroler ESP32-CAM.

4.2.2 Hasil Pengujian Proses Deteksi Objek

Hasil Pada pengujian deteksi objek ini diambil dari keadaan yang sebenarnya, yang dimana mikrokontroler ESP32-CAM telah diberi model AI dan akan mendeteksi telur secara *real-time*. Hasil dari deteksi akan menampilkan posisi telur mana yang retak dan nilai akurasi yang didapatkan. Model AI yang digunakan untuk pengujian ini adalah *default* dari platform *Edge Impulse* yaitu FOMO.

A. Hasil Pengujian Deteksi Satu Butir Telur

Pengujian ini dilakukan untuk mendeteksi satu butir telur dengan menggunakan model deteksi objek yang telah dilatih yaitu FOMO. Hasilnya menunjukkan bahwa sistem mampu menentukan kondisi telur, yaitu retak atau tidak retak dengan cukup akurat. Hal ini ditunjukkan pada Tabel 4.4 dan Tabel 4.5 berikut ini.

Tabel 4.4 Hasil Pengujian Deteksi Satu Butir Telur Retak

Iterasi	Kondisi Telur	Akurasi	Deteksi yang diperoleh
1	Retak	0,996094	Retak
2	Retak	0,996094	Retak
3	Retak	0,996094	Retak
4	Retak	0,996094	Retak
5	Retak	0,996094	Retak
6	Retak	0,996094	Retak
7	Retak	0,996094	Retak
8	Retak	0,996094	Retak
9	Retak	0,996094	Retak
10	Retak	0,996094	Retak

Dari informasi yang terdapat pada Tabel 4.4 menunjukkan 10 data dari 100 data pada hasil pengujian deteksi satu butir telur retak. Dengan rata-rata seluruh akurasi yaitu 0,990547 atau 99%. Data selengkapnya dapat dilihat pada Lampiran 2.

Tabel 4.5 Hasil Pengujian Deteksi Satu Butir Telur Tidak Retak

Iterasi	Kondisi Telur	Akurasi	Deteksi yang diperoleh
1	Tidak Retak	0,562500	Tidak Retak
2	Tidak Retak	0,976562	Tidak Retak
3	Tidak Retak	0,996094	Tidak Retak
4	Tidak Retak	0,988281	Tidak Retak
5	Tidak Retak	0,984375	Tidak Retak
6	Tidak Retak	0,996094	Tidak Retak
7	Tidak Retak	0,996094	Tidak Retak
8	Tidak Retak	0,996094	Tidak Retak
9	Tidak Retak	0,988281	Tidak Retak
10	Tidak Retak	0,996094	Tidak Retak

Dari data pada Tabel 4.5 menunjukkan 10 data dari 100 data pada hasil pengujian deteksi satu butir telur tidak retak. Dengan rata-rata seluruh akurasi yaitu 0,957730 atau 96%. Data selengkapnya dapat dilihat pada Lampiran 3.

B. Hasil Pengujian Deteksi Dua Butir Telur

Pengujian deteksi objek ini dilakukan untuk mendeteksi dua butir telur sekaligus. Sistem diuji untuk dapat membedakan telur yang retak dan tidak retak pada kedua telur tersebut. Hasilnya meliputi keakuratan model dan keberhasilan model dalam mendeteksi kedua objek secara bersamaan. Hal ini dapat dilihat pada Tabel 4.6, Tabel 4.7, Tabel 4.8 dan Tabel 4.9 di bawah ini.

Tabel 4.6 Hasil Deteksi Kondisi Telur 1 Retak dan Telur 2 Tidak Retak

Iterasi	Kondisi Telur	Akurasi		Deteksi yang diperoleh
		Telur 1	Telur 2	
1	Retak	0,996094	0,996094	Retak Kiri
2	Retak	0,996094	0,988281	Retak Kiri
3	Retak	0,996094	0,996094	Retak Kiri
4	Retak	0,996094	0,996094	Retak Kiri
5	Retak	0,996094	0,996094	Retak Kiri
6	Retak	0,996094	0,984375	Retak Kiri
7	Retak	0,996094	0,996094	Retak Kiri
8	Retak	0,996094	0,996094	Retak Kiri
9	Retak	0,996094	0,996094	Retak Kiri
10	Retak	0,996094	0,996094	Retak Kiri

Tabel 4.6 menunjukkan 10 data dari 100 data pada hasil pengujian deteksi dua butir telur dengan kondisi telur 1 retak dan telur 2 tidak retak. Rata-rata seluruh akurasi yang diperoleh dalam deteksi dua telur ini yaitu 0,995938 atau 99%. Data selengkapnya dapat dilihat pada Lampiran 4.

Tabel 4.7 Hasil Deteksi Kondisi Telur 1 Tidak Retak dan Telur 2 Retak

Iterasi	Kondisi Telur	Akurasi		Deteksi yang diperoleh
		Telur 1	Telur 2	
1	Retak	0,925781	0,992188	Retak Kanan
2	Retak	0,734375	0,980469	Retak Kanan
3	Retak	0,996094	0,992188	Retak Kanan
4	Retak	0,988281	0,996094	Retak Kanan
5	Retak	0,972656	0,996094	Retak Kanan
6	Retak	0,855469	0,953125	Retak Kanan
7	Retak	0,972656	0,996094	Retak Kanan
8	Retak	0,972656	0,996094	Retak Kanan
9	Retak	0,984375	0,996094	Retak Kanan
10	Retak	0,562500	0,996094	Retak Kanan

Tabel 4.7 menampilkan 10 data dari 100 data pada hasil pengujian deteksi dua butir telur dengan kondisi telur 1 tidak retak dan telur 2 retak. Rata-rata seluruh akurasi yang diperoleh dalam deteksi dua telur ini yaitu 0,979180 atau 98%. Data selengkapnya dapat dilihat pada Lampiran 5.

Tabel 4.8 Hasil Deteksi Kondisi Telur 1 Retak dan Telur 2 Retak

Iterasi	Kondisi Telur	Akurasi		Deteksi yang diperoleh
		Telur 1	Telur 2	
1	Retak	0,996094	0,996094	Retak Semua
2	Retak	0,996094	0,996094	Retak Semua
3	Retak	0,996094	0,996094	Retak Semua
4	Retak	0,996094	0,996094	Retak Semua
5	Retak	0,996094	0,996094	Retak Semua
6	Retak	0,996094	0,996094	Retak Semua
7	Retak	0,996094	0,996094	Retak Semua
8	Retak	0,996094	0,996094	Retak Semua
9	Retak	0,996094	0,996094	Retak Semua
10	Retak	0,996094	0,996094	Retak Semua

Pada Tabel 4.8 menunjukkan 10 data dari 100 data pada hasil pengujian deteksi dua butir telur dengan kondisi telur 1 retak dan telur 2 retak. Rata-rata seluruh akurasi yang diperoleh dalam deteksi dua telur ini yaitu 0,981504 atau 98%. Data selengkapnya dapat dilihat pada Lampiran 6.

Tabel 4.9 Hasil Deteksi Kondisi Telur 1 Tidak Retak dan Telur 2 Tidak Retak

Iterasi	Kondisi Telur	Akurasi		Deteksi yang diperoleh
		Telur 1	Telur 2	
1	Tidak Retak	0,992188	0,996094	Tidak Retak Semua
2	Tidak Retak	0,996094	0,988281	Tidak Retak Semua
3	Tidak Retak	0,953125	0,996094	Tidak Retak Semua
4	Tidak Retak	0,996094	0,996094	Tidak Retak Semua
5	Tidak Retak	0,996094	0,996094	Tidak Retak Semua
6	Tidak Retak	0,996094	0,988281	Tidak Retak Semua
7	Tidak Retak	0,984375	0,984375	Tidak Retak Semua
8	Tidak Retak	0,996094	0,996094	Tidak Retak Semua
9	Tidak Retak	0,984375	0,996094	Tidak Retak Semua
10	Tidak Retak	0,925781	0,996094	Tidak Retak Semua

Sebagaimana terlihat pada Tabel 4.9 menunjukkan 10 data dari 100 data pada hasil pengujian deteksi dua butir telur dengan kondisi telur 1 tidak retak dan telur 2 tidak retak. Rata-rata seluruh akurasi yang diperoleh dalam deteksi dua telur ini yaitu 0,932441 atau 93%. Data selengkapnya dapat dilihat pada Lampiran 7.

4.3 Kesimpulan Pengujian Deteksi Objek

Setelah melakukan pengujian deteksi objek dan memperoleh akurasi dari setiap deteksi, Selanjutnya kesimpulan hasil deteksi yang ditampilkan pada layar OLED dengan hasil deteksi sebenarnya. Dapat dilihat pada Tabel 4.10, Tabel 4.11, Tabel 4.12, Tabel 4.13, dan Tabel 4.14 berikut ini.

A. Kondisi Telur 1 Retak dan Telur 2 Tidak Retak

Tabel 4.10 Kesimpulan Kondisi Telur 1 Retak dan Telur 2 Tidak Retak

Iterasi	Kondisi Telur		Hasil Deteksi			
	Telur 1	Telur 2	Retak Kanan	Retak Kiri	Retak Semua	Tidak Retak Semua
1	Retak	Tidak Retak		✓		
2	Retak	Tidak Retak			✓	
3	Retak	Tidak Retak		✓		
4	Retak	Tidak Retak		✓		
5	Retak	Tidak Retak		✓		
6	Retak	Tidak Retak		✓		
7	Retak	Tidak Retak			✓	
8	Retak	Tidak Retak		✓		
9	Retak	Tidak Retak		✓		
10	Retak	Tidak Retak		✓		
11	Retak	Tidak Retak		✓		
12	Retak	Tidak Retak		✓		
13	Retak	Tidak Retak		✓		
14	Retak	Tidak Retak		✓		
15	Retak	Tidak Retak		✓		

Pada Tabel 4.10 menunjukkan 15 data dari 100 data pada kesimpulan deteksi telur sebenarnya dengan kondisi telur 1 retak dan telur 2 tidak retak. Akurasi yang diperoleh dari kondisi ini yaitu 94 terdeteksi benar dan 6 terdeteksi salah. Data selengkapnya dapat dilihat pada Lampiran 8.

B. Kondisi Telur 1 Tidak Retak dan Telur 2 Retak

Tabel 4.11 Kesimpulan Kondisi Telur 1 Tidak Retak dan Telur 2 Retak

Iterasi	Kondisi Telur		Hasil Deteksi			
	Telur 1	Telur 2	Retak Kanan	Retak Kiri	Retak Semua	Tidak Retak Semua
1	Tidak Retak	Retak	✓			
2	Tidak Retak	Retak	✓			
3	Tidak Retak	Retak	✓			
4	Tidak Retak	Retak	✓			
5	Tidak Retak	Retak		✓		
6	Tidak Retak	Retak	✓			
7	Tidak Retak	Retak	✓			
8	Tidak Retak	Retak	✓			
9	Tidak Retak	Retak	✓			
10	Tidak Retak	Retak	✓			
11	Tidak Retak	Retak	✓			

Iterasi	Kondisi Telur		Hasil Deteksi			
	Telur 1	Telur 2	Retak Kanan	Retak Kiri	Retak Semua	Tidak Retak Semua
12	Tidak Retak	Retak	✓			
13	Tidak Retak	Retak	✓			
14	Tidak Retak	Retak	✓			
15	Tidak Retak	Retak	✓			

Merujuk pada Tabel 4.11 di atas menunjukkan 15 data dari 100 data pada kesimpulan deteksi telur sebenarnya dengan kondisi telur 1 tidak retak dan telur 2 retak. Akurasi yang diperoleh dari kondisi ini yaitu 94 terdeteksi benar dan 6 terdeteksi salah. Data selengkapnya dapat dilihat pada Lampiran 9.

C. Kondisi Telur 1 Retak dan Telur 2 Retak

Tabel 4.12 Kesimpulan Kondisi Telur 1 Retak dan Telur 2 Retak

Iterasi	Kondisi Telur		Hasil Deteksi			
	Telur 1	Telur 2	Retak Kanan	Retak Kiri	Retak Semua	Tidak Retak Semua
1	Retak	Retak			✓	
2	Retak	Retak			✓	
3	Retak	Retak			✓	
4	Retak	Retak			✓	
5	Retak	Retak			✓	
6	Retak	Retak			✓	
7	Retak	Retak			✓	
8	Retak	Retak			✓	
9	Retak	Retak			✓	
10	Retak	Retak			✓	
11	Retak	Retak			✓	
12	Retak	Retak			✓	
13	Retak	Retak			✓	
14	Retak	Retak			✓	
15	Retak	Retak		✓		

Tabel 4.12 menunjukkan 15 data dari 100 data pada kesimpulan deteksi telur sebenarnya dengan kondisi telur 1 retak dan telur 2 retak. Akurasi yang diperoleh dari kondisi ini yaitu 95 terdeteksi benar dan 5 terdeteksi salah. Data selengkapnya dapat dilihat pada Lampiran 10.

D. Kondisi Telur 1 Tidak Retak dan Telur 2 Tidak Retak

Tabel 4.13 Kesimpulan Kondisi Telur 1 Tidak Retak dan Telur 2 Tidak Retak

Iterasi	Kondisi Telur		Hasil Deteksi			
	Telur 1	Telur 2	Retak Kanan	Retak Kiri	Retak Semua	Tidak Retak Semua
1	Tidak Retak	Tidak Retak				✓
2	Tidak Retak	Tidak Retak				✓
3	Tidak Retak	Tidak Retak				✓
4	Tidak Retak	Tidak Retak	✓			
5	Tidak Retak	Tidak Retak				✓
6	Tidak Retak	Tidak Retak				✓
7	Tidak Retak	Tidak Retak				✓
8	Tidak Retak	Tidak Retak				✓
9	Tidak Retak	Tidak Retak				✓
10	Tidak Retak	Tidak Retak				✓
11	Tidak Retak	Tidak Retak				✓
12	Tidak Retak	Tidak Retak				✓
13	Tidak Retak	Tidak Retak				✓
14	Tidak Retak	Tidak Retak				✓
15	Tidak Retak	Tidak Retak				✓

Dari informasi yang ada pada Tabel 4.13 menunjukkan 15 data dari 100 data pada kesimpulan deteksi telur sebenarnya dengan kondisi telur 1 tidak retak dan telur 2 tidak retak. Akurasi yang diperoleh dari kondisi ini yaitu 89 terdeteksi benar dan 11 terdeteksi salah. Data selengkapnya dapat dilihat pada Lampiran 1.

Tabel 4.14 Kesimpulan Pengujian Deteksi Dua Butir Telur

Kategori Deteksi		Jumlah Uji Sampel	Jumlah Terdeteksi Benar	Jumlah Terdeteksi Salah	Akurasi (%)
Telur 1	Telur 2				
Retak	Tidak Retak	100	94	6	94%
Tidak Retak	Retak	100	94	6	94%
Retak	Retak	100	95	5	95%
Tidak Retak	Tidak Retak	100	89	11	89%

Dari data pada Tabel 4.14, hasil pengujian sampel deteksi terhadap dua butir telur menunjukkan bahwa akurasi deteksi beragam tergantung pada kategori yang diujikan. Akurasi tertinggi terdapat pada kategori telur 1 retak dan telur 2 retak dengan nilai sebesar 95%. Selain itu, akurasi terendah didapatkan pada kategori telur 1 tidak retak dan telur 2 tidak retak dengan nilai sebesar 89%. Indeks 4.1

menunjukkan cara untuk menghitung akurasi deteksi, presentase sampel yang diukur adalah sampel benar dari seluruh sampel.

$$Akurasi = \left(\frac{\text{Jumlah deteksi benar}}{\text{Jumlah sampel uji}} \right) \times 100\% \quad (4.1)$$



UNIVERSITAS
Dinamika

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan pengujian dan analisis yang telah dilakukan dalam pengerjaan Tugas Akhir ini, berikut kesimpulan yang didapat:

1. Pada proses validasi *training* dataset, model FOMO menunjukkan bahwa kinerjanya terbaik dibandingkan dengan model YOLO dan SSD dalam tugas deteksi objek pada dataset yang digunakan. Model FOMO dengan *Epoch* 9 dan *Learning Rate* 0.003 menghasilkan *Precision* sebesar 0.79, nilai *Recall* 0.84 dan *F1-Score* sebesar 0.82. Hal ini menunjukkan keseimbangan optimal dari akurasi dan kemampuan deteksi. Model FOMO lebih efisien dalam penggunaan sumber daya, menjadikan model ini lebih cocok untuk implementasi pada perangkat IoT seperti ESP32-CAM.
2. Proses Pengujian Deteksi Objek terdapat pengujian dengan dua kategori yaitu untuk satu butir telur dan dua butir telur. Nilai akurasi yang didapat yaitu berkisar 0.77 – 0.99, dengan pengujian satu telur dengan kondisi retak akurasi paling rendah adalah 0.77 (77%) dan paling besar adalah 0.99 (99%), sedangkan untuk kondisi tidak retak akurasi paling rendah adalah 0.56 (56%) dan paling besar adalah 0.99 (99%) ini menunjukkan bahwa model cukup akurat dalam mendeteksi objek satu butir telur.
3. Pada pengujian dua butir telur, model dapat mendeteksi keretakan telur dan menghasilkan prediksi posisi dimana telur yang retak. Terlihat nilai akurasi yang didapat berkisar 0.56 – 0.99, dengan akurasi terendah yaitu 0.56 (56%) pada kategori telur 1 tidak retak dan telur 2 retak, telur 1 retak dan telur 2 retak, dan juga telur 1 tidak retak dan telur 2 tidak retak. Di setiap kategori nilai akurasi tertinggi yaitu 0,99 (99%) Hal ini menunjukkan model dapat mendeteksi dengan akurat.
4. Model menunjukkan proses deteksi yang cukup tinggi untuk setiap kategori pengujian dengan nilai akurasi antara 89% sampai dengan 95%. Pada kategori telur 1 retak dan telur 2 retak memiliki akurasi yang tertinggi yaitu 95%, ini menunjukkan model lebih baik dalam mendeteksi kedua telur yang retak.

Sedangkan untuk kategori telur 1 tidak retak dan telur 2 tidak retak mendapatkan akurasi 89%, ini menunjukkan pada proses deteksi sering terjadi kesalahan pada kedua telur yang tidak retak.

5. Secara keseluruhan, model FOMO menunjukkan keunggulan dibandingkan model SSD dan model YOLO dalam hal kinerja dan efisiensi. Pada *Epoch* 12 dan *Learning Rate* 0.02, SSD mendapatkan hasil terbaiknya yaitu *Precision* 66.5%, *Recall* 0.64 dan *mAP* 0.55. Sedangkan untuk model YOLO lebih rendah dari model SSD yaitu pada *Epoch* 6 dan *Learning Rate* 0.1 dengan nilai *Precision* 57.0%, *Recall* 0.42 dan *mAP* 0.27. Dari hasil tersebut, model FOMO lebih cocok digunakan pada perangkat IoT yang memiliki sumber daya terbatas. Di sisi lain, SSD lebih unggul dibandingkan dengan model YOLO dalam hal akurasi pada dataset yang sama.

5.2 Saran

Dari pengujian dan pengembangan sistem deteksi keretakan telur ayam ini, terdapat beberapa saran yang dapat dipertimbangkan untuk lebih lanjut yaitu sebagai berikut:

1. Menggunakan model FOMO untuk diintegrasikan dengan perangkat IoT seperti ESP32-CAM karena model ini memiliki keseimbangan kinerja dan efisiensi dalam penggunaan sumber daya.
2. Mengembangkan Dataset yang lebih banyak dan bervariasi akan membuat hasil deteksi jauh lebih baik.
3. Menguji sistem pada keadaan yang nyata dan lebih kompleks untuk memastikan stabilitas dan keandalan model dalam berbagai keadaan.
4. Mengimplementasikan sistem dengan aktuator tambahan untuk memperluas fungsi sistem.

DAFTAR PUSTAKA

- Adi, P. D. P., & Wahyu, Y. (2022). Performance evaluation of ESP32 Camera Face Recognition for various projects. *Internet of Things and Artificial Intelligence Journal*, 2(1), 10–21. <https://doi.org/10.31763/iota.v2i1.512>
- Neethirajan, S. (2020). The role of sensors, big data and machine learning in modern animal farming. *Sensing and Bio-Sensing Research*, 29, 100367. <https://doi.org/10.1016/j.sbsr.2020.100367>
- Nusyirwan, D., Fahrudin, M., & Putra Perdana, P. P. (2019). Perancangan Purwarupa Pengatur Suhu Otomatis pada Inkubator Penetasan Telur Ayam Menggunakan Arduino Uno dan Sensor Suhu IC LM 35. *JAST: Jurnal Aplikasi Sains Dan Teknologi*, 3(1), 60. <https://doi.org/10.33366/jast.v3i1.1315>
- Parenreng, J. M., Zulhajji, & Fradila, A. I. (2019). Pengembangan Sistem Cerdas Monitoring Inkubator Penetas Telur Jarak Jauh Berbasis Android. *Journal Zetroem*, 1(2), 8–16. <http://eprints.unm.ac.id/20155/>
- Qassim A. Al-Jarwany. (2024). System Design and Implementation for Machine Learning and Internet of Things Based Anomaly Detection in Patient Movement. *Journal of Electrical Systems*, 20(5s), 2216–2232. <https://doi.org/10.52783/jes.2582>
- Raju, N., Navya, A., Koteswaramma, N., Mounika, B., & Rajeshwari, T. (2022). IoT based Door Access Control System using ESP32cam. *International Journal of Engineering Inventions*, 11(12), 9–15. www.ijejournal.com
- Setyawan, A., Yulianto, & Budi, E. S. (2024). Penetas Telur ayam Otomatis dengan Metode Fuzzy Logic Control Dalam Upaya Meningkatkan Penetasan. *Jurnal Elektronika Dan Otomasi Industri*, 11(1), 54–64. <https://doi.org/10.33795/elkolind.v11i1.3488>
- Taleb, H. M., Mahrose, K., Abdel-Halim, A. A., Kasem, H., Ramadan, G. S., Fouad, A. M., Khafaga, A. F., Khalifa, N. E., Kamal, M., Salem, H. M., Alqhtani, A. H., Swelum, A. A., Arczewska-Włosek, A., Świątkiewicz, S., & Abd El-Hack, M. E. (2024). Using artificial intelligence to improve poultry productivity – a review. *Annals of Animal Science*. <https://doi.org/10.2478/aoas-2024-0039>