

**RANCANG BANGUN SISTEM DETEKSI DAN PENGUSIR
HAMA BURUNG PADA TANAMAN PADI BERBASIS EDGE
AI MENGGUNAKAN RASPBERRY PI AI CAMERA DAN
YOLOV11N**



Disusun Oleh:

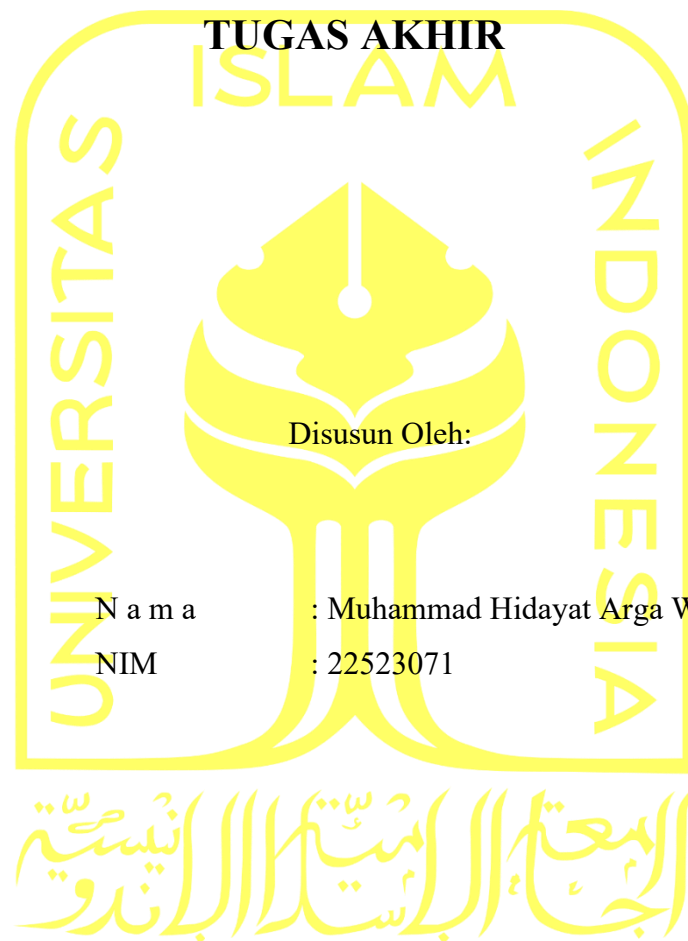
N a m a : Muhammad Hidayat Arga Wicaksana
NIM : 22523071

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2026

HALAMAN PENGESAHAN DOSEN PEMBIMBING

**RANCANG BANGUN SISTEM DETEKSI DAN PENGUSIR
HAMA BURUNG PADA TANAMAN PADI BERBASIS *EDGE*
AI MENGGUNAKAN RASPBERRY PI AI CAMERA DAN
YOLOV11N**



Yogyakarta, 27 April 2026

Pembimbing,

(Kurniawan Dwi Irianto, S.T., M.Sc.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**RANCANG BANGUN SISTEM DETEKSI DAN PENGUSIR
HAMA BURUNG PADA TANAMAN PADI BERBASIS *EDGE*
AI MENGGUNAKAN RASPBERRY PI AI CAMERA DAN
YOLOV11N**

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dari Program Studi Informatika – Program Sarjana di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 20 Mei 2026

Tim Penguji

Kurniawan Dwi Irianto, S.T., M.Sc.

Anggota 1

Moh. Idris, S.Kom., M.Kom.

Anggota 2

Taufiq Hidayat, S.T., M.C.S., Ph.D.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana

Fakultas Teknologi Industri

Universitas Islam Indonesia



(Dhomas Hatta Fudholi, S.T., M.Eng., Ph.D.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Muhammad Hidayat Arga Wicaksana

NIM : 22523071

Tugas akhir dengan judul:

RANCANG BANGUN SISTEM DETEKSI DAN PENGUSIR HAMA BURUNG PADA TANAMAN PADI BERBASIS *EDGE* AI MENGGUNAKAN RASPBERRY PI AI CAMERA DAN YOLOV11N

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila di kemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung risiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 27 April 2026



(Muhammad Hidayat Arga Wicaksana)

HALAMAN PERSEMBAHAN

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT atas rahmat, hidayah, dan kemudahan yang diberikan sehingga penulis dapat menyelesaikan skripsi ini. Karya ini penulis persembahkan kepada kedua orang tua tercinta, Ayah Arif Darmono dan Ibu Rini Rustiani yang selalu mendukung, mendoakan, memberi kasih sayang, serta pengorbanan yang tiada henti. Terima kasih juga penulis sampaikan kepada kakak, keluarga, sahabat, dosen pembimbing, dan semua pihak yang telah membantu dalam proses penyusunan skripsi ini.

HALAMAN MOTO

“Nikmati setiap prosesnya. Hasil hanyalah bonus, karena sejatinya kita sedang belajar dan bertumbuh dalam perjalanan itu.”

(Muhammad Hidayat Arga Wicaksana)

”Allah tidak memberikan sesuatu secara instan tetapi Allah tunjukkan jalannya”

(Muhammad Hidayat Arga Wicaksana)

”Boleh jadi kamu tidak menyenangi sesuatu, padahal itu baik bagimu, dan boleh jadi kamu menyukai sesuatu, padahal itu tidak baik bagimu. Allah mengetahui, sedang kamu tidak mengetahui.”

(Q.S. Al-Baqarah : 216)

“Bermimpilah setinggi langit. Jika engkau jatuh, engkau akan jatuh di antara bintang-bintang.”

(Ir. Soekarno)

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Alhamdulillahirabbil'alamin, puji syukur penulis panjatkan ke hadirat Allah SWT atas nikmat, karunia, dan ridha-Nya sehingga skripsi berjudul “Rancang Bangun Sistem Deteksi dan Pengusir Hama Burung pada Tanaman Padi Berbasis Edge AI menggunakan Raspberry Pi AI Camera dan YOLOv11n” dapat diselesaikan dengan baik dan tepat waktu. Skripsi ini disusun sebagai salah satu syarat memperoleh gelar Sarjana Informatika pada Program Studi Informatika, Fakultas Teknologi Industri, Universitas Islam Indonesia. Dalam proses penyusunan skripsi ini, penulis memperoleh bantuan, bimbingan, dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih yang sebesar-besarnya kepada:

1. Bapak Dr. Raden Teguh Dirgahayu, S.T, M.Sc., selaku Ketua Jurusan Informatika Universitas Islam Indonesia dan Bapak DThomas Hatta Fudholi, S.T., M.Eng., Ph.D. selaku Ketua Program Studi Informatika – Program Sarjana Universitas Islam Indonesia.
2. Bapak Kurniawan Dwi Irianto, S.T., M.Sc., selaku dosen pembimbing tugas akhir. Terima kasih atas bimbingan, masukan, serta waktu yang diluangkan kepada penulis selama melaksanakan penelitian hingga penyusunan skripsi ini.
3. Ibu Sheila Nurul Huda, S.Kom., M.Cs., selaku dosen pembimbing akademik.
4. Dosen dan karyawan di Fakultas Teknik Industri Universitas Islam Indonesia.
5. Kedua orang tua, Papa Arif Darmono dan Mama Rini Rustiani atas dukungan penuh kepada penulis baik dukungan moril maupun materil serta doa yang tiada henti.
6. Kakak Muhammad Hauzan Refin Mumtaz yang selalu memberikan dukungan dan semangat kepada penulis
7. Kepada Hana Aqilah Septini yang menemani saya dalam pengerjaan skripsi ini sebagai support system yang selalu memberikan saya support baik secara langsung maupun tidak langsung dalam membuat skripsi ini selesai serta selalu memberikan semangat dan motivasi tinggi dalam pengerjaannya.
8. Kepada teman seperjuangan NyeeSquad, Rakha, Aban, Agus, Wisnu, Radian, Yasir, Toni, Fikri, Hardy, Aufa, Abi, Fath, Bintang, Akmal, Imada, Narendra, dan Rangga yang membantu dalam pengerjaan skripsi dalam ham diskusi dan berjuang bersama.
9. Seluruh pihak terkait yang tidak dapat disebutkan satu-persatu yang telah membantu selama penyusunan tugas akhir ini.

Semoga Allah SWT senantiasa membalas segala kebbaikannya. Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, oleh karena itu penulis memohon maaf atas segala kekurangannya. Semoga skripsi ini dapat bermanfaat dan menjadi sumber kebaikan bagi siapa pun yang membacanya. Penulis juga mengharapkan adanya kritik dan saran yang konstruktif sebagai bahan evaluasi untuk pengembangan penelitian selanjutnya. Dengan demikian, diharapkan penelitian ini dapat terus disempurnakan dan memberikan kontribusi yang lebih optimal di masa mendatang.

Yogyakarta, 27 April 2026



(Muhammad Hidayat Arga Wicaksana)

SARI

Serangan hama burung dari genus *Lonchura* merupakan salah satu kendala utama yang dapat menyebabkan penurunan hasil panen padi hingga 50%, sementara metode pengendalian manual terbukti kurang efisien dan rentan memicu efek habituasi. Sebagai upaya mitigasi, penelitian ini mengembangkan prototipe sistem deteksi dan pengusir hama burung otomatis berbasis *edge AI*. Sistem ini mengintegrasikan Raspberry Pi 5, Raspberry Pi AI Camera (Sony IMX500) untuk pemrosesan inferensi *on-sensor*, mekanisme *pan-tilt*, serta deterren audio dinamis. Proses pengembangan mengadaptasi model Waterfall yang mencakup komparasi performa antara arsitektur YOLOv8n dan YOLOv11n. Hasil evaluasi performa menyimpulkan bahwa YOLOv11n lebih unggul karena mampu mencapai akurasi mAP50 sebesar 87,6% dengan efisiensi komputasi yang lebih baik. Melalui serangkaian pengujian lapangan, sistem terbukti mampu beroperasi secara *real-time* dengan kecepatan 24-28 *Frame Per Second* (FPS), mendeteksi objek burung hingga jarak 400 cm dalam simulasi gambar dan 200 cm di dalam simulasi sangkar, serta memicu reaksi terbang menjauh pada spesimen burung dalam waktu 1 detik. Evaluasi *User Acceptance Testing* (UAT) kepada petani menghasilkan skor rata-rata penerimaan sebesar 4,10 yang masuk ke dalam kategori baik, mengonfirmasi bahwa teknologi ini aplikatif, responsif, dan memberikan manfaat praktis untuk menggantikan metode penjagaan sawah secara konvensional.

Kata kunci: *Edge AI*, Hama Burung, Raspberry Pi AI Camera, *Smart Farming*, YOLOv11n.

GLOSARIUM

Anchor-free	Pendekatan pada deteksi objek yang tidak bergantung pada anchor box tetap untuk menentukan lokasi objek.
Augmentasi Data	Teknik memperbanyak variasi data latih dengan transformasi seperti rotasi, flip, atau perubahan skala.
Black Box Testing	Metode pengujian sistem yang menilai fungsi berdasarkan input dan output tanpa melihat kode program di dalamnya.
Bounding Box	Kotak pembatas yang menandai posisi objek yang terdeteksi pada citra. Bottleneck hambatan pada aliran data atau proses yang menyebabkan kinerja sistem menjadi lebih lambat.
Computer Vision	Bidang kecerdasan buatan yang mempelajari cara komputer memahami dan menganalisis citra atau video.
Confidence Score	Nilai kepercayaan model terhadap hasil deteksi objek yang dinyatakan dalam bentuk probabilitas.
<i>Deep Learning</i>	Metode pembelajaran mesin yang menggunakan jaringan saraf berlapis untuk mempelajari pola dari data.
Edge AI	Pendekatan kecerdasan buatan yang menjalankan proses inferensi langsung pada perangkat di dekat sumber data.
Field of View	Wilayah pandang kamera atau sensor yang dapat diamati dalam satu waktu.
FPS	Singkatan dari <i>Frame Per Second</i> , yaitu jumlah frame yang diproses atau ditampilkan per detik.
Habitulasi	Penurunan respons organisme terhadap stimulus yang diberikan secara berulang dalam waktu lama.
Inference	Proses menggunakan model yang sudah dilatih untuk menghasilkan prediksi pada data baru.
In-sensor Computing	Pendekatan komputasi yang memindahkan proses pemrosesan langsung ke dalam sensor.
IoT	Singkatan dari Internet of Things, yaitu konsep perangkat yang saling terhubung dan bertukar data melalui internet.
Latensi	Jeda waktu antara input diberikan dan sistem menghasilkan respons.
mAP	Singkatan dari mean Average Precision, yaitu metrik evaluasi untuk mengukur performa deteksi objek.

NPU	Singkatan dari Neural Processing Unit, yaitu prosesor khusus untuk mempercepat komputasi kecerdasan buatan.
Object Detection	Teknik pengenalan objek yang tidak hanya mengklasifikasikan objek, tetapi juga menentukan lokasinya.
Pan-tilt	Mekanisme gerak dua sumbu yang memungkinkan kamera bergerak ke arah horizontal dan vertikal.
Precision	Metrik yang menunjukkan proporsi hasil deteksi benar dibandingkan seluruh hasil yang dinyatakan positif.
Recall	Metrik yang menunjukkan kemampuan model menemukan seluruh objek target yang sebenarnya ada.
Thread	Alur eksekusi terpisah dalam program yang dapat berjalan bersamaan dengan alur lainnya.
UAT	Singkatan dari User Acceptance Testing, yaitu pengujian yang dilakukan untuk menilai penerimaan pengguna terhadap sistem.
Waterfall	Metode pengembangan perangkat lunak beralur linier, di mana setiap tahapannya dikerjakan secara berurutan.
YOLO	Singkatan dari You Only Look Once, yaitu algoritma deteksi objek satu tahap yang memproses citra secara cepat

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR.....	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO.....	vi
KATA PENGANTAR.....	vii
SARI.....	ix
GLOSARIUM	x
DAFTAR ISI	xii
DAFTAR TABEL	xiv
DAFTAR GAMBAR.....	xv
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	3
1.3. Batasan Masalah	4
1.4. Tujuan	4
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	5
1.7. Sistematika Penulisan	5
BAB II TINJAUAN PUSTAKA	7
2.1 Hama Burung Pada Tanaman Padi	7
2.2 Penelitian Terdahulu	8
2.3 <i>Computer vision</i> dan <i>Object detection</i>	11
2.4 Model YOLO	12
2.5 <i>Edge AI</i>	13
2.6 Komponen Alat	13
2.6.1 Raspberry Pi 5	13
2.6.2 Raspberry Pi AI Camera (Sony IMX 500).....	14
2.6.3 Servo MG996R	14
2.6.4 PCA 9685.....	15
2.6.5 Speaker dan <i>Deteren Audio</i>	16
2.7 Pengujian Sistem.....	17
2.7.1 <i>Black Box</i>	17
2.7.2 <i>User Acceptance Testing (UAT)</i>	17
BAB III METODOLOGI PENELITIAN	19
3.1 Tahapan Penelitian	19
3.2 Tahapan Penelitian	21
3.3 Analisis Kebutuhan Sistem	23
4.7.1 Analisis Studi Literatur	23
4.7.2 Pengumpulan Data Observasi	23
4.7.3 Analisis Data Observasi	24
4.7.4 Analisis Kebutuhan Sistem	25
3.4 Perancangan Sistem	27
3.4.1 Perancangan Perangkat Keras	28
3.4.2 Perancangan Perangkat Lunak	34
3.5 Pengumpulan Dataset.....	36
3.5.1 Sumber dan Proses Pengumpulan Data	36
3.5.2 Anotasi Dataset	37

3.5.3	Pembagian Dataset dan Augmentasi.....	39
3.6	Pembuatan Model	40
3.6.1	Arsitektur Model yang Digunakan.....	40
3.6.2	Konfigurasi dan Proses Pelatihan Model	41
3.7	Rancangan Pengujian	43
3.7.1	Pengujian Performa Model	43
3.7.2	Pengujian Fungsional (<i>Black Box Testing</i>).....	43
3.7.3	Pengujian Lapangan	45
3.7.4	Penngujian Penerimaan Pengguna (UAT)	45
BAB IV	HASIL DAN PEMBAHASAN.....	49
4.1	Hasil Pelatihan Model.....	49
4.1.1	Implementasi Skrip Pelatihan	49
4.1.2	Analisis Kurva Pelatihan YOLOv8n.....	51
4.1.3	Analisis Kurva Pelatihan YOLOv11n.....	55
4.1.4	Perbandingan Hasil YOLOv8n dan YOLOv11n	58
4.1.5	Konversi Model ke Format IMX500	60
4.1.6	Implementasi Program Deteksi Objek	61
4.2	Hasil Pengujian <i>Black Box Testing</i>	63
4.3	Hasil Pengujian Kecepatan Inferensi (FPS).....	65
4.4	Hasil Pengujian Jarak Deteksi.....	67
4.4.1	Penngujian Jarak dengan Gambar.....	67
4.4.2	Penngujian Jarak dengann Burung Asli	68
4.5	Hasil Pengujian Reaksi Burung Terhadap Suara	69
4.6	Hasil Pengujian Suhu dan Ketahanan Sistem	71
4.7	Hasil <i>User Acceptance Testing</i> (UAT)	72
4.7.1	Hasil Evaluasi Kuantitatif UAT	72
4.7.2	Hasil Evaluasi Kualitatif UAT	76
BAB V	KESIMPULAN DAN SARAN	78
5.1	Kesimpulan	78
5.2	Saran.....	79
	DAFTAR PUSTAKA	81
	LAMPIRAN	86

DAFTAR TABEL

Tabel 2.1 Perbandingan Penelitian Terdahulu.....	9
Tabel 3.1 Profil Responden dan Temuan Survei Lapangan.....	25
Tabel 3.2 Matriks Kebutuhan Sitem dan Keterkaitannya dengan Data Lapangan.....	26
Tabel 3.3 Spesifikasi Komponen Utama Sistem.....	28
Tabel 3.4 Ringkasan Statistik Dataset.....	40
Tabel 3.5 Konfigurasi Parameter Pelatihan Model YOLO.....	42
Tabel 3.6 Format Rancangan Kasus Uji <i>Black Box</i> Testing.....	44
Tabel 3.7 Rancangan Skenario Pengujian Lapangan.....	45
Tabel 3.8 Rancangan Kuesioner UAT (Skala Likert 1–5).....	46
Tabel 3.9 Kategori Interpretasi Nilai Mean UAT.....	47
Tabel 4.1 Perbandingan Parameter Arsitektur YOLOv8n dan YOLOv11n.....	51
Tabel 4.2 Komparasi Metrik Evaluasi Model YOLOv8n dan YOLOv11n.....	58
Tabel 4.3 Hasil Pengujian <i>Black Box Testing</i> Sistem Deteksi Hama Burung.....	63
Tabel 4.4 Hasil Pengujian FPS pada Sistem Raspberry Pi AI Camera (Sony IMX500).....	64
Tabel 4.5 Hasil Pengujian Jarak Deteksi Menggunakan Gambar.....	66
Tabel 4.6 Rekapitulasi Hasil Pengujian Jarak Deteksi dengan Latar Persawahan.....	67
Tabel 4.7 Hasil Pengujian Reaksi Spesimen Burung terhadap Audio.....	68
Tabel 4.8 Hasil Pengujian Suhu dan Ketahanan Sistem.....	69
Tabel 4.9 Data Hasil Kuisisioner UAT.....	71
Tabel 4.10 Rata-rata Penilaian UAT per Matriks.....	73

DAFTAR GAMBAR

Gambar 2.1 Burung Pemakan Padi dari Genus <i>Lonchura</i>	7
Gambar 3.1 Tahapan Metodologi Pengerjaan Penelitian.....	20
Gambar 3.2 Gambaran Umum Sistem.....	21
Gambar 3.3 Diagram Alur Perancangan Sistem.....	27
Gambar 3.4 Skematik Perancangan Perangkat Keras.....	29
Gambar 3.5 Konfigurasi Raspberry AI Camera.....	30
Gambar 3.6 Konfigurasi Raspberry Pi 5 ke PCA9685.....	31
Gambar 3.7 Konfigurasi PCA9685 ke Servo MG996R.....	32
Gambar 3.8 Konfigurasi Audio.....	32
Gambar 3.9 Rangkaian <i>Hardware</i> Alat.....	33
Gambar 3.10 Rancangan Box.....	34
Gambar 3.11 <i>Activity</i> Diagram Paralel Sistem Deteksi dan Pengusir Hama Burung.....	35
Gambar 3.12 Distribusi Dataset pada <i>Platform</i> Roboflow sebelum Augmentasi.....	37
Gambar 3.13 Contoh Anotasi <i>Bounding box</i> pada Kelas Hama_Burung.....	38
Gambar 3.14 Contoh Gambar Kelas <i>Null</i> (<i>Background</i> tanpa Burung Hama).....	38
Gambar 3.15 Distribusi Dataset Setelah Augmentasi.....	39
Gambar 4.1 Kode Program Fungsi <i>parse_arguments()</i> dan Potongan Fungsi <i>main()</i>	49
Gambar 4.2 <i>command-line arguments</i>	50
Gambar 4.3 Hasil Eksekusi Pelatihan YOLOv8n dan Parameternya.....	50
Gambar 4.4 Hasil Eksekusi Pelatihan YOLOv11n dan Parameternya.....	51
Gambar 4.5 Kurva Hasil pelatihan Model YOLOv8n.....	52
Gambar 4.6 Kurva <i>Precision-Recall</i> Model YOLOv8n	53
Gambar 4.7 <i>Confusion matrix</i> Model YOLOv8n (<i>Raw Count</i>).....	54
Gambar 4.8 <i>Confusion matrix</i> YOLOv8n (<i>Normalized</i>).....	54
Gambar 4.9 Kurva Hasil pelatihan Model YOLOv11n.....	55
Gambar 4.10 Kurva <i>Precision-Recall</i> Model YOLOv11n.....	56
Gambar 4.11 <i>Confusion matrix</i> Model YOLOv11n (<i>Raw Count</i>).....	57
Gambar 4.12 <i>Confusion matrix</i> YOLOv11n (<i>Normalized</i>).....	57
Gambar 4.13 Proses <i>Statistics Collection</i> MCT pada Konversi Model ke Format IMX500.....	60

Gambar 4.14 <i>Memory Report</i> Hasil Konversi Model YOLOv11n ke Format IMX500.....	60
Gambar 4.15 Konfigurasi Global dan Variabel <i>Shared</i> pada <code>final_code.py</code>	61
Gambar 4.16 Logika Koordinasi Servo dan Status Hama pada <code>final_code.py</code>	61
Gambar 4.17 Logika Deteksi dan Manajemen Audio pada fungsi <code>parse_detections()</code>	62
Gambar 4.18 Hasil Pengujian FPS pada Sistem.....	66
Gambar 4.19 Dokumentasi Visual Hasil Pengujian Jarak Deteksi Menggunakan Gambar (100 cm, 200 cm, 300 cm, 400 cm, 500 cm).....	67
Gambar 4.20 Dokumentasi Visual Hasil Pengujian Jarak Deteksi (50 cm, 100 cm, 150 cm, 200 cm, 250 cm).....	69
Gambar 4.21 Dokumentasi Pengujian Reaksi Burung di Sangkar Simulasi.....	67

BAB I PENDAHULUAN

1.1. Latar Belakang

Indonesia merupakan negara agraris di mana sektor pertanian memegang peranan vital dalam struktur perekonomian nasional. Padi (*Oryza sativa L.*) adalah komoditas pangan strategis utama karena beras merupakan makanan pokok bagi hampir 95% penduduk Indonesia (Wuli *et al.*, 2025). Stabilitas produksi padi sangat menentukan ketahanan pangan nasional, stabilitas ekonomi, dan kesejahteraan masyarakat. Oleh karena itu, upaya menjaga kuantitas dan kualitas hasil panen padi menjadi prioritas utama dalam pembangunan pertanian nasional (Afif *et al.*, 2023).

Meskipun produksi padi menjadi prioritas, para petani sering kali dihadapkan pada berbagai kendala biotik yang mengancam penurunan hasil panen. Salah satu Organisme Pengganggu Tanaman (OPT) yang meresahkan pada fase generatif tanaman padi adalah hama burung pemakan bulir padi, khususnya dari genus *Lonchura spp.* Jenis burung yang termasuk dalam kelompok ini antara lain burung bondol peking (*Lonchura punctulata*), bondol jawa (*Lonchura leucogastroides*), dan bondol haji (*Lonchura maja*) (Ardjansyah *et al.*, 2017). Burung-burung tersebut umumnya menyerang secara berkelompok, terutama ketika bulir padi mulai terisi hingga memasuki fase masak susu, karena bulir pada fase tersebut masih lunak dan mudah dikonsumsi. Seekor burung bondol diperkirakan dapat mengonsumsi rata-rata 5 gram padi per hari. Dalam kondisi serangan berat, keberadaan koloni burung pemakan padi dapat menyebabkan kehilangan hasil panen berkisar antara 30% hingga 50%, bahkan berpotensi memicu gagal panen apabila tidak dikendalikan secara serius (Tuti *et al.*, 2024). Oleh karena itu, dalam penelitian ini objek deteksi dibatasi pada hama burung pemakan padi dari genus *Lonchura spp.* agar ruang lingkup sistem selaras dengan karakteristik hama yang umum menyerang tanaman padi.

Selama ini, upaya pengendalian hama burung yang dilakukan oleh petani di Indonesia masih sangat bergantung pada metode konvensional. Berbagai cara yang umum diaplikasikan meliputi pemasangan orang-orangan sawah, jaring pengaman, hingga kaleng bekas yang digantung menggunakan tali (Ardjansyah *et al.*, 2017; Tuti *et al.*, 2024). Petani juga seringkali harus turun langsung ke area sawah untuk mengusir kawanan burung dengan cara berteriak atau menggoyangkan alat bantu fisik (Ardjansyah *et al.*, 2017). Akan tetapi, metode perlindungan manual ini dinilai kurang efektif karena menguras tenaga fisik serta menyita waktu petani (Afif *et al.*, 2023; Tuti *et al.*, 2024).

Seiring dengan kemajuan teknologi, pendekatan *smart farming* berbasis *Internet of Things* (IoT) mulai diterapkan untuk mengatasi keterbatasan metode manual karena sistem ini mampu bekerja secara otomatis (Irianto, 2022). Berbagai penelitian telah mengembangkan alat pengusir hama otomatis menggunakan mikrokontroler yang diintegrasikan dengan sensor gerak *Passive Infrared* (PIR) maupun sensor ultrasonik (Puryono & Pangestu, 2023). Namun penggunaan sensor PIR, memiliki kelemahan tingginya tingkat *false positive* (deteksi palsu) sering kali terjadi karena sensor terpicu oleh fluktuasi termal lingkungan, seperti pergerakan vegetasi atau daun yang tertiup angin (Hobbs & Brehme, 2017). Akibatnya, setiap aktivasi palsu ini tidak hanya akan menimbulkan polusi suara dan pemborosan energi perangkat (Arowolo *et al.*, 2020), tetapi juga berpotensi mempercepat habituasi burung karena terus-menerus terpapar stimulus pengusir tanpa adanya ancaman predator yang nyata (Storms *et al.*, 2022). Hal tersebut mendorong untuk adanya metode yang lebih akurat dalam deteksi hama.

Pendekatan berbasis kamera menawarkan perbaikan karena sistem dapat memvalidasi keberadaan burung secara visual sebelum respons diaktifkan. Penelitian Chen *et al.*, (2024) mengembangkan pengusir burung akustik yang dipicu oleh *computer vision* guna mengetahui pengaruh durasi operasi alat terhadap efektivitas penangkalan. Meskipun sistem ini terbukti andal, implementasi yang mengandalkan transmisi video dari *webcam* ke server eksternal menghadapi tantangan arsitektural. Besarnya volume data dari sensor dapat memicu hambatan komunikasi dalam pemrosesan *cloud*, sementara tuntutan operasi *real time* sering kali melampaui batas toleransi latensi sistem tersebut (Fabre *et al.*, 2024). Di sisi lain, pendekatan *near-sensor* yang memisahkan unit sensor citra dari prosesor pengolahnya memicu perpindahan data yang lambat, yang menjadi penyebab utama pemborosan energi dan hambatan *transfer* data (Fabre *et al.*, 2024). Kendala arsitektural ini mendorong perlunya transisi menuju teknologi *in-sensor computing*, di mana fungsionalitas pemrosesan diintegrasikan langsung ke dalam sensor untuk mencapai operasi mandiri yang lebih efisien di lahan pertanian.

Permasalahan pengendalian hama burung pada tanaman padi tidak hanya terletak pada keterbatasan alat, tetapi pada ketidaksesuaian pendekatan yang digunakan terhadap karakteristik perilaku burung. Metode konvensional terbukti masih bergantung pada stimulus statis yang mudah diadaptasi, sehingga efektivitasnya menurun dalam jangka panjang akibat fenomena habituasi (Storms *et al.*, 2022; Aboltins *et al.*, 2021). Sementara itu, pendekatan berbasis sensor sederhana seperti PIR masih memiliki kelemahan dalam membedakan objek target dan bukan target, yang menyebabkan tingginya tingkat deteksi palsu dan aktivasi yang

tidak relevan (Hobbs & Brehme, 2017; Arowolo *et al.*, 2020). Di sisi lain, pendekatan berbasis *computer vision* yang diaktifkan berdasarkan deteksi objek terbukti lebih efektif karena respon pengusiran tidak diberikan secara terus-menerus, sehingga mengurangi kemungkinan adaptasi burung terhadap stimulus yang diberikan (Chen *et al.*, 2024). Namun demikian, implementasi yang masih bergantung pada pemrosesan berbasis server eksternal menimbulkan kendala latensi dan perpindahan data dalam jumlah besar, yang berdampak pada respons sistem yang kurang optimal untuk aplikasi lapangan (Fabre *et al.*, 2024).

Tanpa sistem yang mampu memberikan deteksi akurat dan respons adaptif secara *real time*, kerugian akibat serangan hama burung berpotensi terus meningkat dan mengurangi hasil panen padi. Penelitian ini mengusulkan pengembangan sistem deteksi dan pengusir hama burung berbasis pendekatan *on-sensor edge* AI dengan memanfaatkan Raspberry Pi AI Camera (Sony IMX500) dan Raspberry Pi 5. Pendekatan ini memungkinkan proses deteksi dilakukan langsung pada sensor, sehingga dapat mengurangi ketergantungan terhadap server eksternal serta meningkatkan efisiensi dan kecepatan respons sistem (Fabre *et al.*, 2024). Sistem yang dikembangkan juga mengintegrasikan mekanisme pan-tilt untuk memperluas cakupan pengamatan secara dinamis, serta penerapan audio dinamis berbasis *event* deteksi guna menekan efek habituasi yang umum terjadi pada stimulus yang bersifat berulang (Storms *et al.*, 2022; Chen *et al.*, 2024). Selain itu, penelitian ini melakukan komparasi performa model YOLOv8n dan YOLOv11n untuk mengevaluasi efisiensi komputasi model yang digunakan. Dengan demikian, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem pengusir hama burung yang lebih adaptif, efisien, dan relevan dengan kondisi pertanian di Indonesia.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, rumusan masalah dalam penelitian ini adalah sebagai berikut:

1. Bagaimana rancangan sistem pendeteksi dan pengusir hama burung pada tanaman padi yang mampu bekerja secara otomatis dan *real time*?
2. Bagaimana kemampuan sistem dalam mengenali hama burung secara akurat dan memberikan respons pengusiran yang efektif?
3. Bagaimana model objek deteksi yang dapat mengenali hama burung pada lingkungan persawahan berdasarkan aspek akurasi dan kecepatan proses?

4. Bagaimana tingkat efektivitas dan penerimaan pengguna terhadap sistem yang dibangun?

1.3. Batasan Masalah

Penelitian ini memiliki beberapa batasan yang ditetapkan untuk memperjelas ruang lingkup penelitian:

1. Objek yang dideteksi dibatasi pada hama burung pemakan padi dari genus *Lonchura spp.*
2. Sistem difokuskan pada deteksi visual dan respons pengusiran berupa suara.
3. Sistem dibangun dengan diasumsikan lingkungan kondisi persawahan yang memiliki akses listrik PLN sebagai sumber catu daya eksternal.
4. Sistem dirancang sebagai prototipe berbasis pemrosesan lokal dan tidak membahas integrasi dengan sistem berbasis jaringan atau *cloud*.

1.4. Tujuan

Berdasarkan rumusan masalah yang telah diuraikan, tujuan penelitian ini adalah sebagai berikut:

1. Mengembangkan dan mengimplementasikan sistem pendeteksi dan pengusir hama burung pada tanaman padi yang mampu bekerja secara otomatis dan *real time*.
2. Menganalisis kemampuan sistem dalam mengenali hama burung secara akurat serta menghasilkan respons pengusiran yang efektif pada kondisi lapangan.
3. Mengevaluasi kinerja model deteksi objek dalam mengenali hama burung pada kondisi lapangan.
4. Menilai tingkat efektivitas sistem dalam mengurangi gangguan hama burung serta tingkat penerimaan pengguna terhadap sistem yang dikembangkan.

1.5. Manfaat Penelitian

Penelitian ini diharapkan memberikan manfaat bagi petani, akademisi, dan pengembang teknologi. Bagi petani, sistem yang dikembangkan dapat menjadi alternatif untuk membantu mengurangi kerugian hasil panen akibat serangan hama burung serta mengurangi ketergantungan pada penjagaan sawah secara manual. Bagi akademisi, penelitian ini dapat menambah referensi ilmiah mengenai penerapan kecerdasan buatan dan visi komputer dalam bidang pertanian, khususnya pada pengendalian hama. Sementara itu, bagi pengembang

teknologi, penelitian ini dapat menjadi acuan dalam pengembangan sistem deteksi objek berbasis perangkat komputasi berbiaya rendah yang dapat diterapkan di lingkungan luar ruang.

1.6. Metodologi Penelitian

Penelitian ini menggunakan pendekatan eksperimental terapan dengan model pengembangan sistem Waterfall yang diadaptasi. Model Waterfall pertama kali diperkenalkan oleh Winston W. Royce pada tahun 1970 sebagai pendekatan pengembangan perangkat lunak yang berlangsung secara bertahap dan berurutan. Dalam penelitian ini, model tersebut digunakan sebagai kerangka kerja umum untuk mengembangkan sistem deteksi dan pengusir hama burung secara sistematis. Penelitian ini dilaksanakan melalui beberapa tahap utama sebagai berikut:

1. Studi literatur dan analisis kebutuhan

Tahap ini dilakukan untuk memahami permasalahan pengusiran hama burung pada tanaman padi serta menentukan kebutuhan sistem yang akan dikembangkan.

2. Perancangan sistem

Pada tahap ini dilakukan penyusunan rancangan umum sistem, meliputi alur kerja deteksi, mekanisme pengusiran, dan integrasi antar komponen.

3. Pengumpulan dan pengelolaan data

Data citra dikumpulkan dari berbagai sumber, kemudian dipilih dan diolah sebagai dasar pelatihan model deteksi objek.

4. Pembuiatan dan implementasi model

Model deteksi objek dilatih, diuji, dan kemudian diterapkan ke dalam sistem yang dirancang untuk bekerja secara otomatis.

5. Pengujian Sistem

Sistem diuji untuk mengetahui kemampuan deteksi, respons pengusiran, dan kinerja keseluruhan pada kondisi yang menyerupai lingkungan persawahan.

1.7. Sistematika Penulisan

Struktur penulisan laporan tugas akhir ini disusun dalam beberapa bab, yaitu:

1. BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika

penulisan. Bagian ini memberikan gambaran umum mengenai cakupan dan arah penelitian.

2. BAB II TINJAUAN PUSTAKA

Memuat teori-teori pendukung mengenai karakteristik tanaman padi dan hama burung, konsep IoT dalam pertanian, dasar-dasar pengolahan citra, teknologi Raspberry Pi, dan kajian penelitian terdahulu yang relevan.

3. BAB III METODOLOGI PENELITIAN

Menjelaskan secara rinci metode yang digunakan, mulai dari alat dan bahan, blok diagram sistem, perancangan perangkat keras dan lunak, hingga skenario pengujian.

4. BAB IV HASIL DAN PEMBAHASAN

Menyajikan data hasil pengujian alat, analisis kinerja algoritma deteksi, pembahasan mengenai efektivitas sistem, serta perbandingan hasil dengan teori atau penelitian sebelumnya.

5. BAB V KESIMPULAN DAN SARAN

Bab ini menyimpulkan hasil penelitian dan memberikan saran untuk pengembangan lebih lanjut, baik dari sisi teknis, fitur sistem, maupun potensi integrasi dengan teknologi lain.

BAB II

TINJAUAN PUSTAKA

2.1 Hama Burung Pada Tanaman Padi

Hama burung dari genus *Lonchura* merupakan salah satu ancaman utama pada pertanaman padi, terutama pada fase generatif ketika bulir mulai terisi. Spesies seperti *L. punctulata*, *L. leucogastroides*, dan *L. maja* umumnya menyerang secara berkelompok dengan intensitas tinggi pada pagi dan sore hari (Ardjansyah *et al.*, 2017). Pola serangan ini menyebabkan gangguan yang berulang dalam satu hari dan sulit dikendalikan secara manual. Burung bondol cenderung menyerang bulir padi pada fase masak susu karena teksturnya masih lunak dan mudah dikonsumsi. Rata-rata konsumsi per individu dapat mencapai sekitar 5 gram per hari (Afif *et al.*, 2023). Meskipun nilai ini relatif kecil, serangan yang dilakukan oleh koloni dalam jumlah besar dapat menyebabkan kerusakan yang signifikan pada lahan persawahan.



Gambar 2.2 Burung Pemakan Padi dari Genus *Lonchura*

Sumber: MPLK Politeknik Negeri Kupang (2023)

Pola serangan berkelompok ini membuat gangguan terhadap tanaman padi tidak hanya terjadi pada satu titik, tetapi dapat menyebar pada beberapa bagian lahan. Burung bondol menyerang tanaman padi dengan memakan bulir padi secara langsung. Selain konsumsi langsung, aktivitas burung ketika hinggap pada malai juga dapat menyebabkan bulir padi rontok atau rusak. Dengan demikian, kerusakan yang ditimbulkan tidak hanya berasal dari jumlah padi yang dimakan, tetapi juga dari kerusakan fisik pada malai akibat aktivitas burung selama menyerang tanaman padi.

Waktu serangan burung bondol juga menunjukkan pola tertentu. Ardjansyah *et al.* (2017) melaporkan bahwa pagi hari menjadi waktu dengan frekuensi serangan tertinggi, yaitu 97 kali di Sawah Baru dan 37 kali di Ciherang. Pola ini menunjukkan bahwa serangan hama burung pada tanaman padi memiliki kecenderungan waktu tertentu, sehingga keberadaannya perlu diperhatikan terutama pada periode aktif burung mencari makan. Pada tingkat lahan, serangan

tersebut dapat menyebabkan kerusakan tanaman padi yang terukur. Hasil pengamatan menunjukkan kerusakan sebesar 4,24% di Sawah Baru dan 4,05% di Ciherang, dengan kerugian ekonomi yang diperkirakan mencapai Rp2.500.325,00 per hektar di Sawah Baru dan Rp1.064.040,00 per hektar di Ciherang.

2.2 Penelitian Terdahulu

Kajian terhadap penelitian terdahulu dilakukan sebagai langkah awal untuk memahami perkembangan penelitian yang telah ada serta mengidentifikasi celah penelitian yang menjadi dasar dilakukannya penelitian ini. Kajian literatur mencakup tiga domain utama, yaitu: (1) tinjauan terhadap metode pengusiran hama burung di lahan pertanian yang telah dilaporkan dalam penelitian sebelumnya; (2) tinjauan terhadap metode deteksi objek berbasis *computer vision*, khususnya arsitektur YOLO yang relevan; serta (3) tinjauan terhadap implementasi *edge AI* pada perangkat komputasi tertanam (*embedded computing*) skala kecil.

Metode pengendalian hama burung konvensional yang paling umum digunakan meliputi orang-orangan sawah, pengusir suara statis, dan kehadiran fisik manusia. Meskipun sederhana dan berbiaya rendah, metode ini memiliki tiga kelemahan utama. Pertama, burung mengalami habituasi dengan cepat terhadap stimulus yang tidak berubah, sehingga efektivitasnya menurun drastis dalam hitungan hari hingga minggu (Storms *et al.*, 2022; Aboltins *et al.*, 2021). Kedua, ketidakefisienan dari segi tenaga dan waktu karena memerlukan kehadiran dan intervensi manusia secara fisik (Afif *et al.*, 2023). Ketiga, sifat yang tidak adaptif: metode tidak dapat merespons perubahan perilaku atau pola serangan burung, dan tidak membedakan antara ancaman nyata dan stimulus palsu (Arowolo *et al.*, 2020). Ketiga kelemahan ini mendorong kebutuhan akan sistem yang otomatis, adaptif, dan berbasis deteksi visual untuk menggantikan pendekatan konvensional.

Perkembangan IoT dalam pertanian membuka peluang untuk mengotomatisasi sistem pengusir hama. Beberapa penelitian telah mengembangkan sistem berbasis sensor PIR, ultrasonik, dan mikrokontroler ESP32 yang mampu mendeteksi gerakan dan mengaktifkan aktuator secara otomatis tanpa intervensi manusia (Puryono & Pangestu, 2023; Afif *et al.*, 2023). Sistem ini menawarkan kemampuan pemantauan jarak jauh melalui aplikasi seperti Blynk atau Telegram. Namun, pendekatan ini tidak mampu melakukan klasifikasi visual objek sehingga rentan terhadap *false positive* akibat pergerakan vegetasi, angin, atau hewan bukan target. Selain itu, respons audio yang monoton dan berulang pada sistem IoT konvensional

justru memperburuk habituasi burung, mengurangi efektivitas pengusiran jangka panjang (Storms *et al.*, 2022; Arowolo *et al.*, 2020).

Pendekatan *computer vision* menawarkan kemampuan identifikasi objek yang jauh lebih akurat dibandingkan sensor gerak pasif. Penelitian Muhammad *et al.*, (2023) mengembangkan sistem *smart farming* menggunakan Raspberry Pi 4 dengan algoritma YOLOv5 dan webcam untuk mendeteksi burung secara *real time*, namun sistem ini masih bergantung pada pemrosesan di server eksternal dan memiliki keterbatasan jangkauan karena kamera bersifat statis. Penelitian Nugroho *et al.*, (2025) menunjukkan bahwa YOLOv8n pada Raspberry Pi 4 dengan Google Coral USB TPU Accelerator mampu melakukan deteksi burung secara *real time*, meski masih memerlukan akselerator eksternal tambahan. Sistem berbasis *computer vision* yang dipicu deteksi terbukti lebih efektif mengurangi kerusakan tanaman dibandingkan sistem akustik yang bekerja terus-menerus (Chen *et al.*, 2024). Kajian terhadap *edge AI* terkini memperlihatkan bahwa Raspberry Pi AI Camera dengan chip Sony IMX500 memungkinkan inferensi dilakukan langsung di tingkat sensor, sehingga beban pemrosesan pada CPU utama dapat dikurangi secara signifikan dan latensi sistem menjadi lebih rendah

Tabel 2. 2 Perbandingan Penelitian Terdahulu

No.	Penelitian	Platform	Metode Deteksi	Kelemahan	Relevansi terhadap Penelitian
1.	An experimental study of acoustic bird repellents for reducing bird encroachment in pear orchards (Chen <i>et al.</i> , 2024)	Server GPU + ESP8266	YOLOv5 + kamera IP + server GPU	Inferensi di server eksternal; memerlukan koneksi Wi-Fi aktif; tidak cocok untuk lahan terpencil	Motivasi penggunaan <i>edge AI</i> untuk eliminasi ketergantungan server eksternal
2.	Implementation of Sparrow Pest Detection Using YOLOv8 Method on Raspberry Pi and Google Coral USB Accelerator (Nugroho <i>et al.</i> , 2025)	Raspberry Pi 4 + Google Coral	YOLOv8n + Google Coral USB TPU	Resolusi input terbatas 480×480; memerlukan akselerator eksternal tambahan	<i>Baseline</i> model YOLOv8n di perangkat <i>embedded</i> ; motivasi penggunaan NPU terintegrasi IMX500

No.	Penelitian	Platform	Metode Deteksi	Kelemahan	Relevansi terhadap Penelitian
3.	Perancangan Sistem Kendali Irigasi Otomatis dan Pengusir Hama Burung Dengan Menggunakan Sensor PIR (Jalaludin & Laksmiati, 2023)	ESP32-Cam + buzzer	Sensor PIR + webcam (deteksi gerak)	Sensor PIR tidak membedakan jenis objek; <i>false positive</i> tinggi akibat angin	Dasar perbandingan sistem sensor pasif
4.	Application of Rice Pest Repellent Using ESP32 Microcontroller and Internet of Things Based Sound Wave (Puryono & Pangestu, 2023)	ESP32 + IoT (Blynk)	Sensor PIR + gelombang ultrasonik	Tidak melakukan deteksi visual; ketergantungan koneksi internet	Referensi arsitektur IoT berbasis ESP32; motivasi penggunaan deteksi visual
5.	Radar Wing Defender (RWD): Sistem untuk membantu para petani dalam mendeteksi dan mengusir hama burung (Fawwaz <i>et al.</i> , 2024)	ESP32 + IoT	Sensor ultrasonik HC-SR04	Tidak membedakan burung dari objek lain; cakupan area terbatas ± 3 meter	Motivasi sistem dengan kemampuan klasifikasi visual dan respons suara variatif
6.	Sistem Kontrol dan Monitoring Tinggi Air Serta Pengusir Hama Burung pada Tanaman Padi Berbasis Internet of Things Dan Android (Safinatun <i>et al.</i> , n.d.)	Raspberry Pi 3 + buzzer + tali	Webcam + deteksi gerak (tanpa <i>deep learning</i>)	Tidak menggunakan model AI; rentan <i>false positive</i> ; buzzer monoton (habitulasi cepat)	Dasar perbandingan sistem webcam sederhana; motivasi variasi suara pengusir

Dari hasil kajian terhadap penelitian terdahulu, terlihat bahwa pendekatan yang paling banyak digunakan masih bertumpu pada sensor sederhana, sistem pengusiran berbasis suara statis, atau kamera konvensional yang diproses pada perangkat eksternal. Sistem deteksi berbasis kamera pada perangkat eksternal atau server masih menghadapi kendala latensi, ketergantungan jaringan, dan kompleksitas implementasi di lahan pertanian yang minim infrastruktur. (Chen *et al.*, 2024) melaporkan bahwa sistem pengusir akustik yang dipicu oleh *computer vision* lebih efektif dibandingkan suara yang bekerja terus-menerus, karena respons berbasis deteksi membuat pengusiran lebih adaptif dan lebih sulit dihabituasi oleh burung.

Berdasarkan temuan tersebut, penelitian ini diarahkan pada integrasi Raspberry Pi AI Camera dengan chip Sony IMX500, model YOLO varian nano, dan mekanisme aktuasi pan-tilt serta audio dinamis sebagai pendekatan yang lebih efisien, adaptif, dan relevan untuk pengusiran hama burung di lahan pertanian. Model YOLOv8n digunakan sebagai *baseline* karena telah terbukti efektif di penelitian sebelumnya (Nugroho *et al.*, 2025), sementara YOLOv11n diuji sebagai kandidat model terbaru yang menawarkan akurasi lebih tinggi dengan parameter lebih sedikit (Ultralytics, 2024).

2.3 *Computer vision dan Object detection*

Computer vision adalah bidang kecerdasan buatan yang memungkinkan mesin untuk menganalisis dan menginterpretasikan informasi visual dari dunia nyata (Fabre *et al.*, 2024). Dalam konteks pertanian, *computer vision* berperan krusial untuk mengidentifikasi objek biologis seperti hama secara otomatis tanpa keterlibatan manusia secara langsung. Salah satu tugas utama dalam *computer vision* adalah deteksi objek (*object detection*), yaitu proses mengidentifikasi dan melokalisasi *instance* objek tertentu dalam sebuah citra, yang menghasilkan keluaran berupa *bounding box* beserta kelas dan skor kepercayaan (*confidence score*).

Evolusi metode deteksi objek dapat dibagi menjadi beberapa generasi. Generasi pertama adalah CNN klasik yang melakukan ekstraksi fitur hierarkis secara otomatis, namun pendekatan dua tahap seperti R-CNN dan Faster R-CNN memisahkan proses proposal region dan klasifikasi sehingga relatif lambat untuk keperluan *real time*. Generasi berikutnya menghadirkan *Single Shot Multibox Detector* (SSD) dan RetinaNet yang mempercepat proses dengan deteksi satu tahap, namun masih menghadapi *trade off* antara kecepatan dan akurasi pada objek kecil. YOLO (*You Only Look Once*) mengambil pendekatan yang berbeda dengan

memperlakukan deteksi objek sebagai masalah regresi tunggal yang menghasilkan *bounding box* dan probabilitas kelas secara langsung dari satu *forward pass* jaringan, menghasilkan kecepatan inferensi yang jauh lebih tinggi dibandingkan pendekatan dua tahap (Hussain & Khanam, 2024).

Keunggulan YOLO untuk penerapan *edge device* terletak pada arsitekturnya yang efisien secara komputasional: satu jaringan tunggal memproses seluruh citra sekaligus, menghasilkan latensi rendah yang esensial untuk aplikasi *real time* di lapangan. Dibandingkan SSD maupun *Faster R-CNN*, YOLO secara konsisten menawarkan *throughput* lebih tinggi pada *hardware* terbatas dengan akurasi yang kompetitif, menjadikannya pilihan dominan untuk sistem *embedded vision* (Fabre *et al.*, 2024).

2.4 Model YOLO

YOLO pertama kali diperkenalkan oleh Redmon *et al.*, dan sejak saat itu mengalami perkembangan yang signifikan. Ultralytics YOLOv8 menjadi standar *de facto* untuk deteksi objek *real time* dengan memperkenalkan arsitektur *anchor free*, *decoupled detection head*, dan *backbone* C2f yang meningkatkan akurasi pada objek kecil sekaligus mengurangi kompleksitas konfigurasi (Ultralytics, 2024). Varian YOLOv8n (nano) dirancang khusus untuk perangkat *edge* dengan parameter yang sangat efisien, dan penggunaannya pada Raspberry Pi telah divalidasi dalam penelitian Nugroho *et al.*, 2025, sehingga menjadikannya *baseline* yang tepat untuk penelitian ini.

Ultralytics YOLO11 membawa peningkatan arsitektur lebih lanjut dengan menggantikan blok C2f menjadi C3k2 yang lebih kompak dan memperkenalkan modul C2PSA (*Cross-Stage Partial Spatial Attention*) untuk meningkatkan kemampuan deteksi objek kecil. Secara empiris, YOLO11 mencapai mAP yang lebih tinggi dengan parameter lebih sedikit dibandingkan YOLOv8 pada dataset COCO: YOLO11n mencapai ~39.5% mAP dengan latensi CPU ~56.1 ms, lebih efisien dibandingkan YOLOv8n (Ultralytics, 2024). Kecepatan inferensi CPU yang lebih rendah ini menjadikan YOLO11n kandidat unggul untuk *deployment* pada perangkat *edge* dengan sumber daya terbatas.

Alasan utama pemilihan kedua model ini sebagai subjek komparasi adalah: (1) YOLOv8n telah divalidasi sebagai *baseline* efektif pada *platform embedded* untuk deteksi hama burung (Nugroho *et al.*, 2025); (2) YOLO11n menawarkan peningkatan akurasi dan efisiensi komputasional yang berpotensi meningkatkan performa pada kondisi lapangan; dan

(3) keduanya merupakan bagian dari ekosistem Ultralytics yang memiliki dukungan ekspor ke format yang kompatibel dengan platform *on-sensor inference* seperti Sony IMX500.

2.5 Edge AI

Edge AI mengacu pada pendekatan di mana inferensi model kecerdasan buatan dijalankan langsung pada perangkat yang berada di tepi jaringan (*edge device*), dekat dengan sumber data, tanpa bergantung pada server *cloud* eksternal (Fabre *et al.*, 2024). Konsep ini lahir sebagai respons terhadap keterbatasan fundamental pemrosesan berbasis *cloud* untuk aplikasi yang memerlukan respon *real time*: latensi transmisi data, ketergantungan koneksi internet, konsumsi *bandwidth* tinggi, dan biaya operasional yang tidak sesuai untuk lingkungan pertanian yang sering kali memiliki infrastruktur terbatas.

Dalam konteks pengusiran hama burung di lahan pertanian, *edge AI* menjadi kebutuhan yang tidak dapat dikompromikan. Ketika sistem mendeteksi burung, respons pengusiran harus terjadi dalam milidetik, latensi jaringan yang mengirim data ke server *cloud* dan menunggu respons dapat mencapai ratusan milidetik, yang membuat sistem tidak efektif. Selain itu, lahan persawahan sering kali berada di area terpencil dengan koneksi internet yang tidak stabil atau tidak tersedia sama sekali. (Fabre *et al.*, 2024) menjelaskan bahwa sistem *near sensor* konvensional masih memisahkan sensor dan prosesor sebagai entitas terpisah, menciptakan data *movement bottleneck* yang meningkatkan latensi dan konsumsi daya. Solusi *in sensor computing* di mana inferensi dilakukan langsung di dalam *chip* kamera mengeliminasi *bottleneck* ini secara fundamental.

2.6 Komponen Alat

Penelitian ini menggunakan beberapa komponen *hardware* utama yang saling terintegrasi untuk membentuk sistem deteksi dan pengusir hama burung. Berikut adalah landasan teori dari masing-masing komponen yang digunakan.

2.6.1 Raspberry Pi 5

Raspberry Pi 5 adalah komputer papan tunggal (*single-board computer*) generasi terbaru dari Raspberry Pi Foundation, dirilis pada Oktober 2023. Platform ini ditenagai oleh prosesor ARM Cortex-A76 quad-core berkecepatan 2,4 GHz yang memberikan peningkatan performa CPU hingga dua kali lipat dibandingkan generasi sebelumnya, dilengkapi dengan RAM LPDDR4X hingga 8 GB, antarmuka PCIe 2.0, dual-port USB 3.0, dan dua port CSI kamera (Raspberry Pi, 2024).

Dalam konteks sistem pengusir burung ini, Raspberry Pi 5 berperan sebagai orkestrator pusat yang menangani seluruh logika kendali: menerima dan menginterpretasikan metadata inferensi dari IMX500 melalui library Picamera2, mengevaluasi hasil deteksi, kemudian mengirimkan perintah aktuasi ke PCA9685 melalui bus I2C, serta mengelola *playback* audio. Kapasitas pemrosesan yang tinggi diperlukan agar seluruh subsistem ini dapat dijalankan secara bersamaan dalam satu alur Python yang *real time* tanpa *bottleneck*. Pemilihan Raspberry Pi 5 dibandingkan Raspberry Pi 4 atau versi lebih lama dilandasi oleh kebutuhan *throughput* komputasi yang lebih tinggi untuk mengorkestrasi kamera AI Camera generasi terbaru bersama aktuasi multi-komponen.

2.6.2 Raspberry Pi AI Camera (Sony IMX 500)

Raspberry Pi AI Camera adalah modul kamera kompak yang didasarkan pada Sony IMX500 Intelligent Vision Sensor, diluncurkan pada Oktober 2024. IMX500 merupakan sensor CMOS pertama di dunia yang mengintegrasikan akselerator AI langsung pada *chip* (*on-chip*), menggabungkan sensor gambar 12,3 megapiksel dengan unit pemrosesan *neural network* pada substrat chip yang sama (Raspberry Pi, 2024).

Arsitektur IMX500 memiliki dua aliran data: *image stream* berisi data gambar untuk *preview* atau perekaman, dan *inference stream* berisi tensor hasil komputasi *neural network* yang dieksekusi langsung pada sensor. Kedua aliran ini ditransmisikan secara simultan melalui antarmuka CSI-2 ke Raspberry Pi. Dengan demikian, Raspberry Pi tidak perlu menjalankan *neural network* pada CPU-nya; beban komputasi CPU dapat sepenuhnya dialokasikan untuk logika kendali sistem (Raspberry Pi, 2024; Ultralytics, 2025). Model custom yang telah dilatih dikonversi ke format IMX menggunakan Sony IMX500 Converter Tool yang terintegrasi dalam Ultralytics, yang melibatkan kuantisasi INT8 menggunakan Model *Compression Toolkit* (MCT) agar model dapat dimuat dalam memori *on-chip* sebesar 8 MB.

Alasan pemilihan Raspberry Pi AI Camera dalam sistem ini adalah kemampuannya melakukan inferensi *on-sensor* yang mengurangi beban komputasi CPU secara signifikan, kompatibilitasnya yang resmi dengan *pipeline* pelatihan YOLO11 via Ultralytics, serta integrasinya yang *native* dengan library Picamera2 pada Raspberry Pi OS. Kombinasi ini memungkinkan pengembangan sistem yang lebih hemat daya, lebih cepat merespons, dan lebih stabil dibandingkan arsitektur yang menempatkan seluruh beban inferensi pada CPU.

2.6.3 Servo MG996R

Servo MG996R merupakan aktuator berbasis servo yang digunakan secara luas dalam sistem robotika dan otomasi karena kemampuannya dalam menghasilkan gerakan terkontrol

berbasis sinyal *Pulse Width Modulation* (PWM). Berbeda dengan motor DC konvensional, servo motor bekerja menggunakan mekanisme kontrol tertutup yang memungkinkan penyesuaian posisi secara *real time* berdasarkan sinyal masukan. Dalam implementasi sistem elektromekanik, servo umumnya diklasifikasikan menjadi dua tipe utama, yaitu servo positional dengan rentang sudut terbatas dan servo *continuous* yang dapat berputar penuh 360° dengan kontrol kecepatan. Penggunaan servo dalam sistem pan-tilt telah banyak diterapkan pada aplikasi pelacakan objek dan sistem visi komputer karena kemampuannya dalam mengubah orientasi perangkat secara dinamis dan presisi terhadap target deteksi (Syarif Hidayat & Kusuma Raharja, 2022).

Dalam sistem pengusir burung ini, servo motor digunakan untuk mekanisme pan-tilt yang memungkinkan kamera AI Camera memindai area sawah yang lebih luas, serta untuk menciptakan gerakan fisik yang dapat berfungsi sebagai visual *deterrent* tambahan. Servo dengan torka tinggi (seperti MG996R dengan torka hingga 9,4 kg·cm pada tegangan 5V) dipilih karena harus mampu menggerakkan beban kamera beserta dudukannya, termasuk dalam kondisi lingkungan yang berdebu dan lembab. Penggunaan roda gigi logam memberikan ketahanan lebih baik dibandingkan servo plastik untuk operasi jangka panjang di lapangan terbuka (Shende & Anande, 2019).

2.6.4 PCA 9685

PCA9685 adalah IC *driver* PWM 16-kanal yang dikontrol melalui antarmuka I2C, diproduksi oleh NXP Semiconductors. *Chip* ini dirancang untuk situasi di mana mikrokontroler atau single-board computer tidak memiliki cukup pin GPIO yang mendukung output PWM untuk menggerakkan banyak servo motor secara bersamaan. PCA9685 memiliki *clock* internal yang menghasilkan sinyal PWM secara mandiri tanpa membebani prosesor utama (Adafruit, 2023). Spesifikasi utama mencakup 16 kanal *output* PWM dengan resolusi 12-bit (4.096 langkah per kanal), frekuensi operasi yang dapat diprogram dari 40 Hz hingga 1.000 Hz, tegangan input 5–10 V DC, serta antarmuka I2C yang memungkinkan kendali melalui hanya dua kabel (SDA dan SCL) (Fiolka *et al.*, 2023).

Pemilihan PCA9685 dalam sistem ini didasarkan pada tiga pertimbangan utama. Pertama, Raspberry Pi hanya memiliki satu kanal PWM *hardware* (GPIO18), sehingga untuk mengendalikan dua servo motor secara bersamaan dibutuhkan *driver* eksternal. Kedua, PCA9685 menghasilkan sinyal PWM berpresisi tinggi dengan resolusi 4.096 langkah, yang memungkinkan kontrol sudut servo yang halus dan dapat diandalkan. Ketiga, komunikasi I2C

hanya membutuhkan dua kabel, menjaga layout rangkaian tetap sederhana dan meminimalkan penggunaan pin GPIO pada Raspberry Pi (Adafruit, 2023).

2.6.5 Speaker dan *Deterren* Audio

Speaker merupakan komponen aktuator yang berfungsi untuk menghasilkan keluaran suara sebagai respons terhadap deteksi keberadaan hama burung. Dalam sistem pengendalian hama modern, penggunaan speaker tidak hanya berperan sebagai perangkat output, tetapi menjadi bagian dari mekanisme deterren berbasis audio yang dirancang untuk memanfaatkan respons perilaku alami burung terhadap ancaman.

Deterren audio merupakan metode pengendalian *non-lethal* yang bekerja dengan memanfaatkan sensitivitas burung terhadap sinyal akustik tertentu, khususnya *distress call* dan predator call. *Distress call* merupakan suara yang dihasilkan oleh burung dalam kondisi terancam, sedangkan predator call merepresentasikan keberadaan pemangsa alami. Kedua jenis suara ini memicu respons naluriah berupa penghindaran area, sehingga burung cenderung menjauhi sumber suara (Adekunle *et al.*; Lim *et al.*, 2025). Pendekatan ini dinilai lebih ramah lingkungan karena tidak melibatkan kontak fisik maupun eliminasi langsung terhadap hama.

Meskipun demikian, efektivitas deterren audio sangat dipengaruhi oleh pola penyajian stimulus. Penggunaan suara yang diputarkan secara berulang dan tidak bervariasi cenderung menimbulkan habituasi, yaitu penurunan respons akibat paparan stimulus yang sama secara terus-menerus. Dalam konteks ini, burung secara bertahap akan mengabaikan suara yang tidak lagi dianggap sebagai ancaman nyata. Penelitian menunjukkan bahwa variasi frekuensi, pola, dan waktu pemutaran memiliki peran penting dalam mempertahankan respons penghindaran burung terhadap stimulus akustik (Dong *et al.*, 2025; Lim *et al.*, 2025).

Implikasi dari temuan tersebut adalah bahwa sistem deterren audio tidak cukup hanya menghasilkan suara, tetapi harus mampu menghadirkan stimulus yang dinamis dan tidak mudah diprediksi. Pendekatan berbasis pemicu deteksi (*event-based triggering*) menjadi relevan, di mana suara hanya diaktifkan ketika keberadaan burung teridentifikasi. Strategi ini tidak hanya mengurangi kemungkinan habituasi, tetapi juga meningkatkan efisiensi penggunaan energi karena sistem tidak bekerja secara kontinu. Dengan demikian, integrasi speaker dalam sistem tidak hanya diposisikan sebagai komponen output, tetapi sebagai bagian dari mekanisme respons adaptif yang bergantung pada hasil deteksi visual. Pendekatan ini menunjukkan pergeseran dari sistem pengusiran statis menuju sistem yang bersifat kontekstual dan responsif terhadap kondisi lingkungan.

2.7 Pengujian Sistem

Pengujian pada sistem berbasis IoT tidak cukup dilakukan hanya pada level perangkat lunak karena melibatkan berbagai komponen yang saling terhubung, mulai dari sensor, perangkat *edge*, hingga aktuator dalam arsitektur yang berlapis dan kompleks. Kondisi ini menuntut pengujian yang tidak hanya memverifikasi fungsi program, tetapi juga memastikan integrasi antar komponen dapat berjalan dengan baik dalam kondisi nyata. Studi terkait pengujian IoT menunjukkan bahwa evaluasi sistem perlu dilakukan secara komprehensif dengan mempertimbangkan berbagai aspek seperti tujuan pengujian, lingkungan pengujian, objek yang diuji, serta pendekatan yang digunakan (Minani, 2025). Berdasarkan hal tersebut, pengujian dalam penelitian ini difokuskan pada validasi fungsi sistem sekaligus integrasi perangkat seperti kamera, aktuator servo, dan sistem audio dalam skenario kerja lapangan secara *real time*, dengan menggunakan metode *Black Box* untuk pengujian fungsional serta *User Acceptance Testing* (UAT) untuk mengevaluasi penerimaan pengguna.

2.7.1 *Black Box*

Black Box Testing merupakan metode pengujian yang berfokus pada penilaian fungsionalitas sistem berdasarkan kesesuaian dengan kebutuhan, tanpa perlu menganalisis struktur kode internal perangkat (Maspupah, 2024). Karakteristik metode ini dapat ditinjau berdasarkan tujuh kriteria utama, yang meliputi: fokus pengujian, basis uji, sudut pandang penguji, desain kasus uji, metode identifikasi kesalahan program, pengetahuan kode program, dan efektivitas pengujian (Maspupah, 2024). Dalam konteks sistem otomasi pengusir hama, pendekatan *Black Box* ini sangat presisi untuk memvalidasi apakah setiap pemicu masukan (trigger input) dari deteksi visual mampu menghasilkan luaran aktuasi yang benar pada mekanisme pan-tilt dan sistem audio.

2.7.2 *User Acceptance Testing* (UAT)

User Acceptance Testing (UAT) merupakan tahap pengujian akhir dalam siklus pengembangan perangkat lunak yang dilakukan secara langsung oleh pengguna akhir (*end-user*) untuk memverifikasi apakah seluruh fungsi yang dikembangkan telah berjalan sesuai dengan kebutuhan dan ekspektasi pengguna di lapangan (Aliyah *et al.*, 2024). UAT berbeda dari pengujian unit maupun pengujian integrasi karena titik fokusnya bukan pada komponen teknis secara terisolasi, melainkan pada keseluruhan pengalaman pengguna dalam menjalankan skenario nyata (Thabibi *et al.*, 2025). Menurut Hermansah *et al.*, (2025), UAT digunakan untuk mengukur tingkat penerimaan (*receptiveness*) pengguna terhadap sistem yang dikembangkan,

sehingga hasil pengujian dapat dijadikan sebagai bukti bahwa perangkat lunak telah dapat diterima dan siap digunakan secara operasional.

Terdapat beberapa jenis UAT yang umum digunakan dalam praktik pengembangan perangkat lunak, antara lain *Alpha Testing*, *Beta Testing*, *Contract Acceptance Testing*, *Regulation Acceptance Testing*, dan *Operational Acceptance Testing* (Kusuma & Yufon, 2024). Penelitian ini menggunakan pendekatan *beta testing*, di mana sistem didemonstrasikan secara langsung kepada pengguna akhir yang representatif, kemudian pengguna diminta mengisi kuesioner terstruktur untuk memberikan penilaian. Pelaksanaan UAT dilakukan setelah tahap pengujian fungsional (*black box testing*) selesai, karena UAT merupakan tahap validasi akhir sebelum sistem dinyatakan layak dioperasikan (Hasugian, 2023).

BAB III

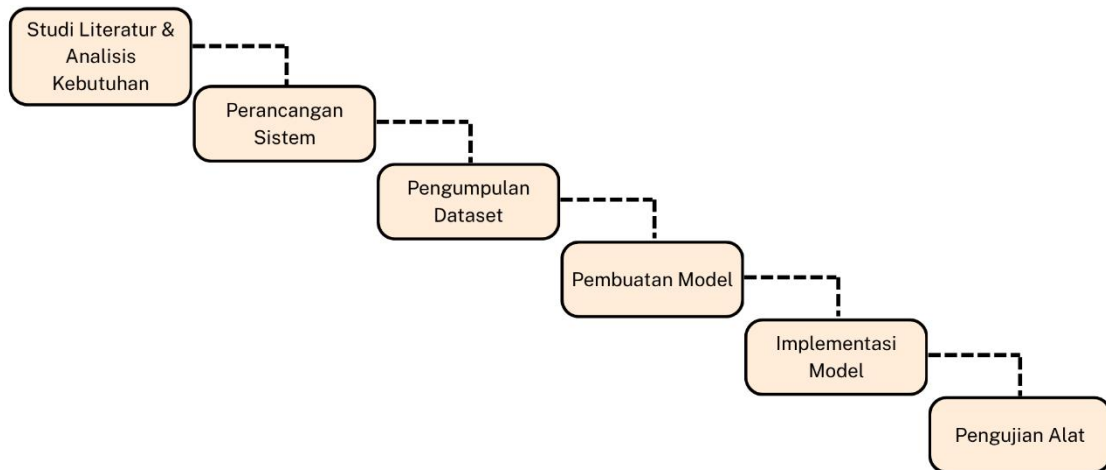
METODOLOGI PENELITIAN

Penelitian ini menggunakan pendekatan eksperimental-terapan dengan model pengembangan sistem yang mengacu pada metode Waterfall yang diadaptasi. Model Waterfall pertama kali diperkenalkan oleh Winston W. Royce pada tahun 1970 sebagai pendekatan pengembangan perangkat lunak yang bersifat sekuensial, di mana setiap tahap harus diselesaikan sebelum melanjutkan ke tahap berikutnya (Royce, 1970). Dalam implementasinya, pendekatan ini menekankan alur pengembangan yang terstruktur dan berurutan, sehingga setiap fase menghasilkan keluaran yang menjadi masukan bagi fase berikutnya. Namun demikian, berbagai penelitian menunjukkan bahwa pendekatan Waterfall klasik memiliki keterbatasan, terutama karena sifatnya yang kaku dan tidak fleksibel terhadap perubahan kebutuhan yang muncul selama proses pengembangan (Pargaonkar, 2023; (Wahyuni *et al.*, 2024).

Oleh karena itu, penelitian ini tidak menggunakan Waterfall klasik secara kaku, melainkan menggunakan Waterfall yang diadaptasi dengan menambahkan mekanisme validasi dan umpan balik terbatas antar tahapan. Adaptasi ini dilakukan dengan mempertimbangkan karakteristik penelitian yang melibatkan pengolahan dataset, pelatihan model *deep learning*, serta integrasi perangkat keras, yang secara inheren memiliki ketidakpastian selama proses pengembangan. Pendekatan ini sejalan dengan temuan literatur yang menyatakan bahwa model Waterfall modern sering dimodifikasi atau dikombinasikan dengan elemen lain untuk meningkatkan fleksibilitas tanpa menghilangkan struktur sistematisnya (Saravanos & Curinga, 2023).

3.1 Tahapan Penelitian

Penelitian ini dilaksanakan melalui enam tahap utama yang disusun secara berurutan dan logis. Setiap tahap memiliki masukan, proses, dan keluaran yang jelas, sehingga alur pengerjaan dapat diikuti secara sistematis. Keenam tahap tersebut adalah: (1) studi literatur dan analisis kebutuhan, (2) perancangan sistem, (3) pengumpulan dataset, (4) pembuatan model, (5) implementasi model, dan (6) rancangan pengujian. Diagram alur metodologi penelitian disajikan pada Gambar 3.1 berikut ini.



Gambar 3.16 Tahapan Metodologi Pengerjaan Penelitian

Tahap pertama, studi literatur dan analisis kebutuhan, bertujuan untuk membangun landasan teoretis yang kuat dan memahami kebutuhan nyata di lapangan sebelum proses perancangan dimulai. Pada tahap ini dilakukan kajian terhadap penelitian terdahulu, teori deteksi objek berbasis *deep learning*, dan tinjauan terhadap berbagai pendekatan pengusiran hama burung yang telah ada. Selain kajian pustaka, dilakukan pula wawancara dan observasi langsung kepada petani guna mengidentifikasi kebutuhan sistem secara empiris.

Tahap kedua adalah perancangan sistem, yang mencakup perancangan arsitektur perangkat keras (*hardware*) dan perangkat lunak (*software*). Perancangan *hardware* meliputi penyusunan skematik rangkaian elektronik yang menghubungkan Raspberry Pi 5, Raspberry Pi AI Camera, modul PCA9685, servo motor, speaker, dan catu daya eksternal. Perancangan *software* meliputi penyusunan alur program, pembuatan *flowchart*, dan penentuan *library* yang akan digunakan.

Tahap ketiga adalah pengumpulan dataset, yang bertujuan menyediakan data latih yang representatif dan berkualitas untuk proses pelatihan model. Dataset dikumpulkan dari berbagai sumber, termasuk *platform* anotasi *online* Roboflow, unduhan gambar dari internet, serta pengambilan sampel video rekaman di area persawahan. Seluruh gambar kemudian dianotasi secara manual menggunakan format *bounding box* standar YOLO sebelum dibagi menjadi set pelatihan, validasi, dan pengujian.

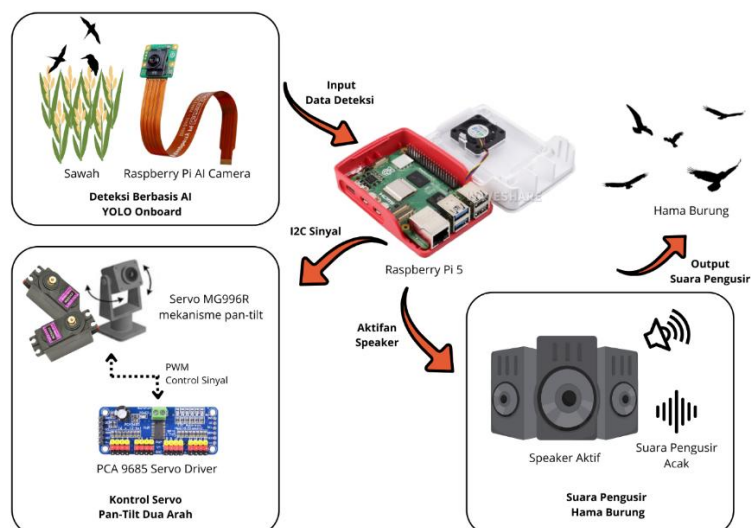
Tahap keempat adalah pembuatan model, yang meliputi pelatihan model deteksi objek menggunakan dua arsitektur YOLO, yaitu YOLOv8n dan YOLOv11n. Kedua model dilatih menggunakan dataset yang telah disiapkan pada tahap sebelumnya, dengan parameter training yang dikonfigurasi secara terstruktur. Hasil pelatihan berupa file bobot model terbaik yang akan dievaluasi berdasarkan metrik *precision*, *recall*, mAP50, dan mAP50-95.

Tahap kelima adalah implementasi model, yaitu proses mengubah model hasil pelatihan menjadi format yang dapat dijalankan secara *onboard* pada Raspberry Pi AI Camera. Model diekspor ke format ONNX, kemudian dikonversi lebih lanjut menjadi format yang kompatibel dengan IMX500 *Neural Processing Unit* (NPU) yang terdapat pada kamera AI. Setelah konversi, model diintegrasikan ke dalam sistem melalui skrip Python yang menghubungkan deteksi, kontrol servo, dan pemutaran suara.

Tahap keenam adalah rancangan pengujian, yang mencakup pengujian fungsional *Black Box*, pengujian performa model, pengujian lapangan dalam kondisi menyerupai sawah, serta pengujian penerimaan pengguna (UAT) yang melibatkan petani sebagai responden. Rancangan pengujian disusun secara sistematis agar hasil yang diperoleh dapat dianalisis secara kuantitatif maupun kualitatif.

3.2 Tahapan Penelitian

Sistem yang dikembangkan dalam penelitian ini merupakan prototipe pengusir hama burung berbasis *computer vision* yang dirancang untuk bekerja secara otomatis pada lingkungan persawahan. Sistem memanfaatkan model *object detection* YOLO untuk melakukan identifikasi hama burung secara visual, kemudian menghasilkan respons berupa pergerakan aktuator pan-tilt dan aktivasi output audio. Berbeda dengan pendekatan konvensional yang bersifat statis dan tidak selektif, sistem ini bekerja berdasarkan hasil *inference* sehingga hanya merespons ketika objek target benar-benar terdeteksi.



Gambar 3.17 Gambaran Umum Sistem

Gambar 3.2 memperlihatkan alur integrasi antar subsistem. Proses *inference* dijalankan langsung pada modul kamera berbasis Sony IMX500 menggunakan pendekatan *edge* AI.

Pendekatan ini mengurangi ketergantungan pada CPU Raspberry Pi 5 sebagai *controller*, sehingga beban komputasi utama tidak terpusat. Secara teknis, distribusi ini menurunkan *latency* deteksi dan memungkinkan sistem mempertahankan performa *real time*, sebagaimana tercermin pada implementasi *pipeline* deteksi yang berjalan kontinu pada kamera dan hanya mengirim metadata hasil deteksi ke *controller*.

Secara arsitektur, sistem terdiri atas empat subsistem utama. Pertama, subsistem deteksi yang menjalankan model YOLO dan menghasilkan output berupa *bounding box*, label, dan *confidence score*. Kedua, subsistem aktuator berupa mekanisme pan-tilt berbasis dua servo motor yang dikendalikan melalui *driver* PCA9685 menggunakan komunikasi I2C. Ketiga, subsistem output berupa speaker aktif yang menghasilkan stimulus audio sebagai respon pengusiran. Keempat, subsistem kontrol berbasis Raspberry Pi 5 yang berfungsi sebagai *controller* untuk mengelola logika sistem, termasuk sinkronisasi antara deteksi, aktuasi servo, dan audio.

Dari sisi operasional, sistem bekerja dalam dua mode utama, yaitu mode *scanning* dan mode *response*. Pada mode *scanning*, kamera secara terus-menerus melakukan akuisisi citra dan menjalankan proses *detection*. Di saat yang sama juga mekanisme pan-tilt bergerak menggunakan pola *time based scanning*. Pola ini dirancang untuk memperluas cakupan pengamatan dan mengurangi *blind spot* akibat keterbatasan *field of view* kamera statis.

Ketika objek dengan label target terdeteksi dan nilai *confidence* melebihi *threshold* yang ditentukan, sistem berpindah ke mode *response*. Pada kondisi ini, mekanisme servo dihentikan sementara menjaga stabilitas *frame* dan menghindari gangguan *motion blur* terhadap proses *detection* lanjutan. Secara bersamaan, sistem mengaktifkan speaker audio sebagai respons langsung. Implementasi ini bersifat *event driven* untuk memastikan sinkronisasi antara subsistem deteksi dan aktuator.

Untuk mengatasi keterbatasan sistem konvensional yang mudah mengalami habituasi, audio tidak diputar secara statis melainkan menggunakan pendekatan dinamis. Selain itu, diterapkan mekanisme *cooldown* berbasis waktu, di mana audio tetap aktif dalam durasi tertentu meskipun objek tidak lagi terdeteksi. Pendekatan ini menghindari kondisi *false negative* sesaat akibat *occlusion* atau kehilangan *frame*, sekaligus mencegah sistem menjadi terlalu reaktif terhadap fluktuasi deteksi.

Setelah tidak terdapat deteksi dalam periode tertentu, sistem kembali ke mode *scanning* dan melanjutkan siklus pengamatan. Pola ini membentuk pola sistem berulang yang menghubungkan proses *detection* dan *response* secara langsung tanpa intervensi manual.

Sebagai pendukung evaluasi, sistem juga dilengkapi dengan mekanisme *logging* yang mencatat jumlah deteksi dan kondisi perangkat secara periodik. Implementasi ini memungkinkan analisis performa sistem dalam jangka waktu tertentu, termasuk stabilitas operasi dan frekuensi kemunculan objek target di lapangan.

3.3 Analisis Kebutuhan Sistem

3.3.1 Analisis Studi Literatur

Kajian literatur menunjukkan bahwa pengembangan sistem pengusir hama burung telah mengalami pergeseran dari pendekatan berbasis sensor sederhana menuju pendekatan berbasis *computer vision* dan *edge AI*. Sistem berbasis sensor seperti PIR memiliki keterbatasan utama pada selektivitas karena tidak mampu membedakan objek target dan *non-target*, sehingga memicu banyak deteksi palsu. Di sisi lain, pendekatan berbasis *computer vision* menawarkan kemampuan identifikasi objek secara spesifik, namun masih menghadapi kendala pada latensi dan beban komputasi ketika inferensi dilakukan pada CPU atau server eksternal.

Penelitian terdahulu juga menunjukkan bahwa efektivitas sistem pengusiran tidak hanya ditentukan oleh kemampuan deteksi, tetapi juga oleh pola respons yang diberikan. Sistem audio yang berjalan secara terus-menerus terbukti kurang efektif karena memicu habituasi, sedangkan sistem yang diaktifkan secara selektif berdasarkan deteksi objek memiliki tingkat efektivitas yang lebih tinggi dalam mengurangi kerusakan tanaman. Dengan demikian, kebutuhan sistem tidak hanya terletak pada akurasi deteksi, tetapi juga pada respons yang adaptif, kontekstual, dan tidak monoton.

3.3.2 Pengumpulan Data Observasi

Analisis kebutuhan sistem dilakukan melalui dua metode pengumpulan data primer yang saling melengkapi: wawancara semi-terstruktur dan survei kuesioner tertulis. Wawancara dilakukan pada bulan April, 2025 di wilayah Kecamatan Kadudampit, Kabupaten Sukabumi, Jawa Barat, mencakup empat narasumber, Lukmanul Hakim (usia 39 tahun, lahan ± 1 ha), Jamaludin (usia 40 tahun, lahan $\pm 0,5$ ha), Dahlan (usia 66 tahun), dan Yusuf (usia 39 tahun) serta Iing (usia 36 tahun) yang dipilih secara purposif berdasarkan pengalaman aktif bertani padi minimal 5 tahun dan riwayat serangan burung hama. Seluruh wawancara dilakukan secara tatap langsung di lokasi lahan dengan panduan pertanyaan semi-terstruktur yang mencakup aspek pola serangan burung, metode konvensional yang digunakan, persepsi efektivitas, dan ekspektasi terhadap sistem otomatis.

Survei kuesioner dilakukan terhadap 14 responden petani padi di wilayah Kadudampit menggunakan instrumen tertulis yang mencakup 30 pertanyaan tertutup dan terbuka. Responden memiliki rentang usia 25–66 tahun (rata-rata 40,9 tahun) dan pengalaman bertani 5–40 tahun (rata-rata 19,9 tahun), sehingga representatif sebagai petani yang sudah memiliki pengetahuan lapangan yang matang. Seluruh responden (100%) menyatakan pernah mengalami gangguan hama burung, yang mengonfirmasi relevansi topik penelitian terhadap populasi sasaran.

3.3.3 Analisis Data Observasi

Seluruh 14 responden (100%) menggunakan kombinasi metode konvensional, dengan orang-orangan sawah, kaleng atau benda pemantul suara, jaring, dan penjagaan manual sebagai pilihan paling umum. Lima responden menggunakan kombinasi keempatnya secara bersamaan. Tingkat efektivitas metode yang ada dinilai sedang oleh 8 responden (skala 3 dari 5), sementara 3 responden menilai cukup efektif (skala 4) dan 3 responden menilai kurang efektif (skala 2). Tidak ada satupun responden yang menilai efektivitas sangat tinggi atau sangat rendah.

Temuan yang paling signifikan adalah pada aspek habituasi: 11 dari 14 responden (78,6%) menilai bahwa burung sangat mudah terbiasa dengan metode yang ada (skor 4–5 dari 5). Lukmanul Hakim menyatakan efektivitas jaring hanya sekitar 50% karena burung perlahan terbiasa dan tetap kembali meski suara kaleng dibunyikan. Dahlan mengonfirmasi bahwa metode suara manual sudah tidak efektif sejak lama karena burung akan kembali jika lahan tidak dijaga terus-menerus. Ini selaras dengan temuan Chen *et al.*, (2024) yang secara eksperimental membuktikan bahwa audio terus-menerus menyebabkan habituasi lebih cepat dibandingkan sistem yang dipicu secara selektif.

Penjagaan manual juga menghadapi kendala tenaga kerja yang semakin serius. Petani perlu hadir di sawah selama berjam-jam pada jam-jam aktif serangan. Yusuf dan Iing sama-sama menyebut bahwa kendati jaring adalah metode paling efektif, efektivitasnya tetap maksimal hanya ketika dipadukan dengan penjagaan manusia untuk menghalau burung yang masih masuk dari celah. Hal ini mengonfirmasi kebutuhan akan sistem yang dapat bekerja mandiri tanpa ketergantungan pada kehadiran manusia

Tabel 3. 10 Profil Responden dan Temuan Survei Lapangan

Aspek yang Diukur	Temuan Utama	Implikasi Desain
Profil responden	14 petani; usia rata-rata 40,9 th; pengalaman rata-rata 19,9 th; lahan 0,5–2 ha	Sistem harus sesuai untuk lahan skala kecil-menengah dengan kemampuan teknis terbatas
Fase & frekuensi serangan	100% responden: serangan setiap hari selama fase masak susu; 100% pada siang hari; 43% juga di pagi & sore	Sistem harus aktif penuh saat fase pengisian gabah; prioritas deteksi siang hari
Estimasi kerugian panen	64,3% estimasi ~50%; 35,7% estimasi >75% dari total panen	Kerugian sangat signifikan; justifikasi kuat untuk investasi sistem otomatis
Efektivitas metode konvensional	57,1% menilai sedang (skor 3/5); tidak ada yang menilai sangat efektif	Metode konvensional tidak cukup; kebutuhan solusi baru tervalidasi
Tingkat habituasi burung	78,6% menilai burung sangat mudah terbiasa (skor 4–5/5)	Sistem harus menerapkan variasi deterren; tidak boleh bersifat monoton
Ketersediaan akses listrik	35,7% ada akses; 35,7% tidak pasti/jauh; 28,6% tidak ada akses listrik	Sistem harus hemat daya; desain modular untuk sumber daya alternatif di masa depan
Fitur prioritas sistem otomatis	100% memilih suara bervariasi; 78,6% memilih deteksi otomatis + gerak kamera + suara bervariasi	Tiga komponen wajib: deteksi visual, aktuasi mekanik (pan-tilt), audio dinamis
Pentingnya operasi otonom	85,7% menilai sangat penting (skor 5/5); 14,3% penting (skor 4/5)	Sistem harus mampu beroperasi penuh tanpa pengawasan manusia
Penempatan alat	100% responden memilih tengah sawah; Yusuf: 'bisa berputar menjangkau seluruh area'	Pemasangan di tiang tengah lahan; mekanisme rotasi pan-tilt diperlukan untuk jangkauan luas
Akses listrik (wawancara)	Lukmanul Hakim & Jamaludin: tidak ada akses listrik dari PLN di tengah lahan; Dahlan: hanya lahan dekat pemukiman	Konfirmasi pentingnya efisiensi daya; akses listrik tidak dapat diasumsikan tersedia

3.3.4 Analisis Kebutuhan Sistem

Berdasarkan sintesis data survei, wawancara, dan kajian literatur, sistem yang dikembangkan harus memenuhi spesifikasi kebutuhan fungsional dan *non-fungsional* berikut.

Dari sisi kebutuhan fungsional, sistem wajib mampu mendeteksi keberadaan burung hama secara visual dan selektif menggunakan model *deep learning* yang telah dilatih pada dataset *Lonchura spp.* Sistem harus merespons deteksi secara otomatis dalam waktu yang sangat singkat dengan memutar audio deterren yang dipilih secara acak dari pustaka suara yang beragam. Sistem juga harus menggerakkan kamera secara mekanis melalui mekanisme pan-tilt untuk memperluas cakupan pemindaian. Selain itu, sistem harus mampu beroperasi mandiri tanpa intervensi pengguna selama periode aktif serangan burung, serta menghentikan atau menjeda *deterren* secara otomatis ketika tidak ada deteksi dalam periode tertentu untuk mencegah habituasi.

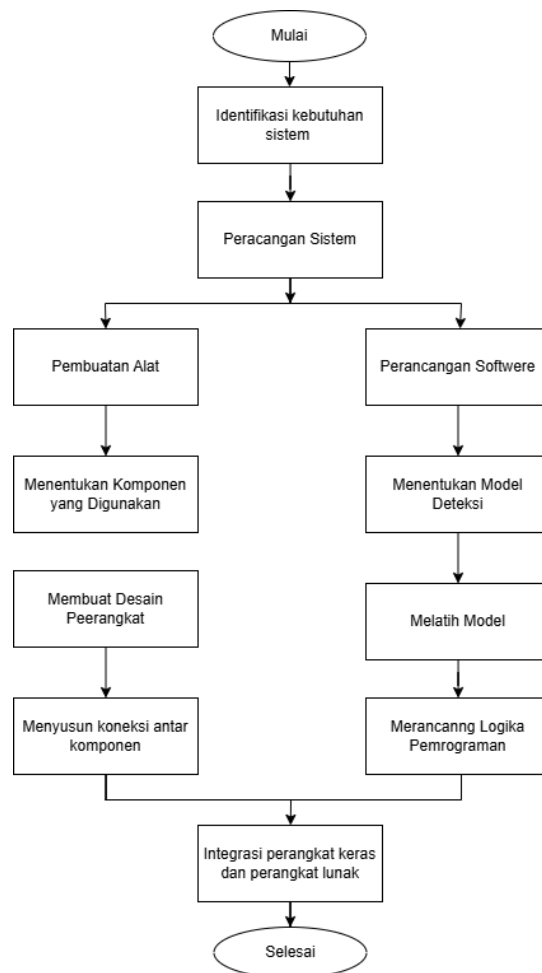
Dari sisi kebutuhan *non-fungsional*, sistem harus memiliki latensi deteksi yang rendah agar respons pengusiran terjadi sebelum kawanan burung mendarat dan mulai memakan padi. Konsumsi daya harus efisien mungkin mengingat 64,3% responden tidak memiliki akses listrik yang pasti di tengah lahan ini menjadi pertimbangan penting untuk keberlanjutan penggunaan. Sistem harus memiliki ketahanan yang memadai terhadap kondisi lingkungan luar ruangan, termasuk variasi pencahayaan, suhu, kelembapan, dan debu. Antarmuka sistem harus cukup sederhana sehingga dapat dipahami dan dioperasikan oleh petani dengan latar belakang pendidikan yang beragam (dari SD hingga SMA, sesuai profil responden).

Tabel 3. 11 Matriks Kebutuhan Sistem dan Keterkaitannya dengan Data Lapangan

Kebutuhan Sistem	Kategori	Dasar: Data Lapangan & Literatur
Deteksi visual otomatis berbasis <i>deep learning</i>	Fungsional	100% responden menyatakan metode konvensional tidak cukup; (Nugroho <i>et al.</i> , 2025) membuktikan YOLOv8n mencapai mAP50 0,92
Audio deterren yang bervariasi dan dipicu selektif	Fungsional	100% responden menyatakan suara bervariasi lebih baik (rata-rata skor 4,7/5); (Chen <i>et al.</i> , 2024): audio dipicu CV mengurangi kerusakan 53,86% vs tanpa deterren
Mekanisme pan-tilt servo untuk cakupan luas	Fungsional	100% responden memilih tengah sawah; 92,9% membutuhkan alat yang bisa bergerak; Yusuf: 'bisa berputar menjangkau seluruh area'
Operasi otonom tanpa pengawasan terus-menerus	Fungsional	100% responden menilai penting/sangat penting (skor $\geq 4/5$); seluruh narasumber wawancara mengonfirmasi keterbatasan tenaga kerja manual
Latensi deteksi rendah (<i>real time</i>)	Non-Fungsional	Nugroho <i>et al.</i> : CPU murni hanya 0,39 FPS tidak memadai; <i>on-sensor inference</i> IMX500 mengatasi <i>bottleneck</i> ini

Kebutuhan Sistem	Kategori	Dasar: Data Lapangan & Literatur
Konsumsi daya efisien	Non-Fungsional	64,3% responden tidak memiliki akses listrik yang pasti; Lukmanul Hakim & Dahlan: tidak ada PLN di tengah lahan
Ketahanan terhadap kondisi outdoor	Non-Fungsional	Lahan sawah terpapar sinar matahari langsung, hujan, kelembapan tinggi, dan debu; persyaratan lingkungan operasional lapangan terbuka
Operasi tanpa koneksi internet	Non-Fungsional	Lokasi lahan di area terpencil (Kadudampit, Sukabumi) dengan akses internet tidak konsisten; <i>edge AI</i> mengeliminasi ketergantungan <i>cloud</i>

3.4 Perancangan Sistem



Gambar 3.18 Diagram Alur Perancangan Sistem

Perancangan sistem pada penelitian ini disusun dengan membagi pengembangan ke dalam dua aspek utama, yaitu perangkat keras dan perangkat lunak. Pembagian ini dilakukan agar setiap komponen dapat dirancang sesuai perannya dalam sistem, sekaligus memudahkan proses integrasi antar bagian. Pada sisi perangkat keras, perancangan mencakup pemilihan

komponen utama seperti Raspberry Pi 5, AI Camera, modul pengendali servo, serta sistem audio, beserta konfigurasi koneksi antar komponen melalui antarmuka seperti I2C, PWM, dan CSI. Pada sisi perangkat lunak, perancangan difokuskan pada alur kerja sistem yang mengatur proses deteksi objek menggunakan model YOLO, pemrosesan hasil deteksi, serta pengendalian aktuator dan keluaran audio sebagai respons terhadap keberadaan hama. Selain itu, sistem juga dirancang untuk mencatat data operasional yang digunakan sebagai bahan evaluasi kinerja selama pengujian. Integrasi antara perangkat keras dan perangkat lunak ini membentuk mekanisme kerja sistem yang berjalan secara otomatis berdasarkan hasil deteksi visual di lapangan. Gambar 3.3 menunjukkan alur perancangan sistem yang menggambarkan tahapan utama mulai dari identifikasi kebutuhan hingga integrasi seluruh komponen yang membentuk sistem secara keseluruhan.

3.4.1 Perancangan Perangkat Keras

Perancangan perangkat keras sistem pengusir hama burung ini didasarkan pada prinsip modularitas. Setiap komponen dipilih berdasarkan kompatibilitas teknis satu sama lain, ketersediaan di pasaran Indonesia, serta kemampuan untuk memenuhi kebutuhan fungsional sistem yang telah diidentifikasi pada tahap analisis kebutuhan. Komponen utama sistem terdiri atas Raspberry Pi 5 sebagai unit komputasi pusat, Raspberry Pi AI Camera sebagai subsistem persepsi visual, modul PCA9685 sebagai pengendali servo, dua buah servo motor sebagai aktuator pan-tilt, speaker aktif sebagai keluaran suara, konverter USB DAC sebagai antarmuka audio, serta catu daya eksternal 5V 6A sebagai sumber energi seluruh sistem.

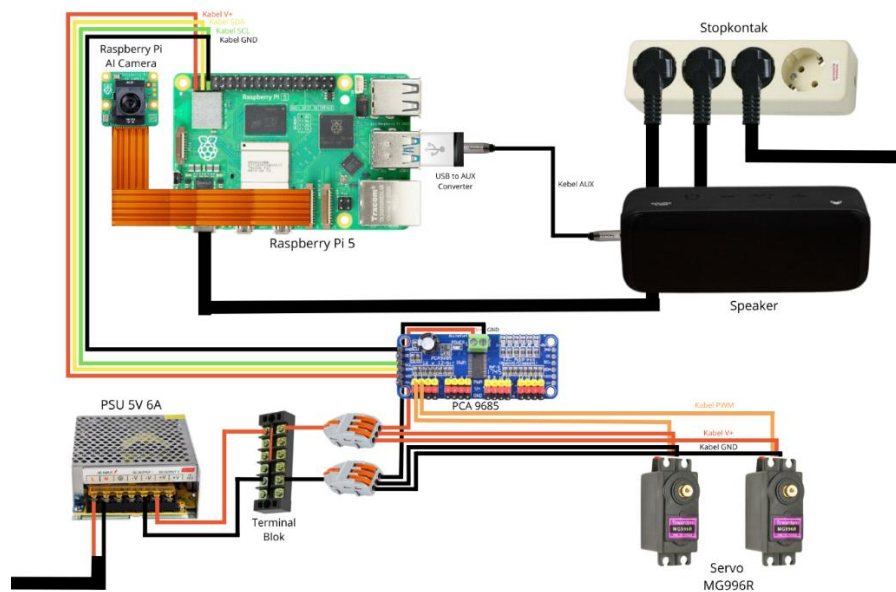
Spesifikasi komponen utama yang digunakan dalam penelitian ini disajikan pada Tabel 3.3 berikut ini. Pemilihan setiap komponen didasarkan pada pertimbangan fungsional, ketersediaan di pasaran, kompatibilitas antarkomponen, serta kesesuaian dengan batasan daya dan ukuran prototipe.

Tabel 3. 12 Spesifikasi Komponen Utama Sistem

Komponen	Spesifikasi Utama	Fungsi dalam Sistem
Raspberry Pi 5	ARM Cortex-A76, 4-core, 4GB RAM, GPIO 40-pin	Unit pemrosesan utama, menjalankan skrip Python, mengintegrasikan seluruh subsistem
Raspberry Pi AI Camera	Sony IMX500, 12MP, NPU <i>onboard</i> , antarmuka CSI-2	Menangkap citra dan menjalankan inferensi model YOLO secara <i>onboard</i>
Modul PCA9685	16-channel PWM, I2C, frekuensi hingga 1.6 kHz	Menghasilkan sinyal PWM untuk mengendalikan dua servo motor melalui I2C
Servo MG996R (Standar)	Torsi 9.4 kg.cm, sudut 0–180°, PWM 50Hz	Menggerakkan sumbu vertikal (tilt) mekanisme kamera

Komponen	Spesifikasi Utama	Fungsi dalam Sistem
Servo Kontinu	360° rotasi, kecepatan dikendalikan <i>throttle</i> , PWM	Menggerakkan sumbu horizontal (pan) mekanisme kamera
Speaker Aktif	Stereo, daya 3–5W, input AUX 3.5mm	Mengeluarkan suara pengusir hama burung
Konverter USB DAC	USB 2.0, output AUX 3.5mm, stereo	Mengubah sinyal digital dari Raspberry Pi menjadi sinyal audio analog
Power supply 5V 6A	Input 220V AC, output 5V DC, arus maksimal 6A	Menyediakan daya untuk servo dan sistem melalui terminal blok
Terminal Blok	5-way, konektor kawat 2.5mm ²	Mendistribusikan jalur daya positif dan <i>ground</i> ke komponen aktuator

Seluruh komponen dihubungkan melalui terminal blok sebagai titik distribusi daya. PSU 5V 6A menyediakan jalur positif dan *ground* yang didistribusikan ke Raspberry Pi 5, PCA9685, dan servo motor secara terpisah. Pemisahan jalur daya ini penting untuk mencegah interferensi elektrik antara subsistem digital (Raspberry Pi 5 dan PCA9685) dan subsistem aktuator (servo motor) yang memiliki karakteristik arus yang berbeda. Skematik lengkap perancangan *hardware* disajikan pada Gambar 3.4.



Gambar 3.19 Skematik Perancangan Perangkat Keras

Konfigurasi Raspberry AI Camera

Raspberry Pi AI Camera berfungsi sebagai subsistem persepsi visual yang tidak hanya merekam citra, tetapi juga melakukan inferensi model secara langsung di tingkat sensor melalui chip Sony IMX500 yang dilengkapi NPU *onboard*. Pendekatan ini dikenal sebagai *in-sensor*

computing atau *edge inference*, yang memungkinkan pemrosesan data visual dilakukan sangat dekat dengan sumber data tanpa harus mengirimkan seluruh *frame* video ke unit CPU utama. Dengan demikian, beban komputasi pada Raspberry Pi 5 dapat dikurangi secara signifikan, dan latensi antara deteksi dan respons aktuasi dapat diperkecil. Kamera terhubung ke Raspberry Pi 5 melalui antarmuka CSI-2 dengan menggunakan kabel pita fleksibel khusus.

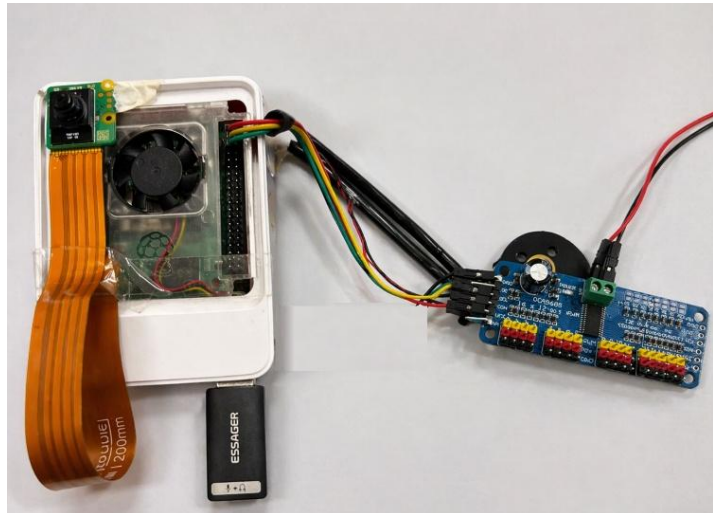


Gambar 3. 20 Konfigurasi Raspberry AI Camera

Konfigurasi Komunikasi I2C: Raspberry Pi 5 ke PCA9685

Modul PCA9685 merupakan pengendali PWM berbasis I2C yang mampu menghasilkan sinyal PWM pada 16 kanal secara independen dengan resolusi 12-bit. Modul ini dipilih karena Raspberry Pi 5 tidak memiliki cukup pin PWM *hardware* untuk mengendalikan dua servo motor secara bersamaan dengan frekuensi yang stabil. Dengan menggunakan PCA9685, Raspberry Pi 5 hanya perlu mengirimkan data melalui dua kabel I2C (SDA dan SCL), sementara modul PCA9685 bertanggung jawab menghasilkan sinyal PWM yang tepat untuk masing-masing servo. Frekuensi PWM yang digunakan adalah 50 Hz, sesuai dengan spesifikasi standar servo.

Secara konfigurasi, jalur I2C yang digunakan terdiri atas VCC, SDA, SCL, dan GND. VCC dihubungkan ke pin 2 Raspberry Pi 5 sebagai suplai logik untuk chip PCA9685, SDA dihubungkan ke pin 3, SCL dihubungkan ke pin 5, dan GND dihubungkan ke pin 6. Alamat I2C default modul PCA9685 adalah 0x40. Frekuensi PWM yang digunakan adalah 50 Hz, sesuai dengan karakteristik servo hobby.

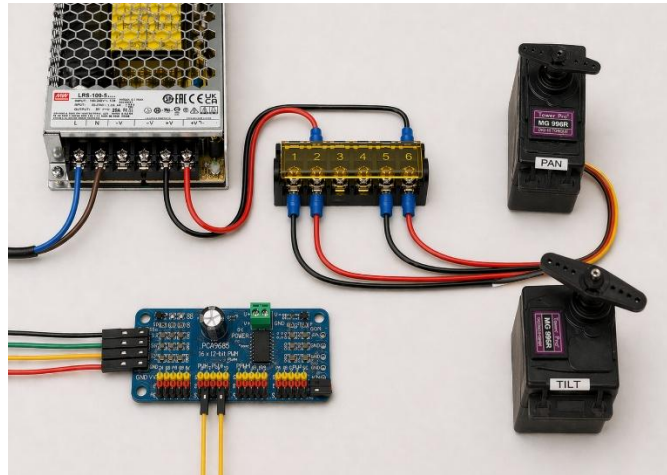


Gambar 3.21 Konfigurasi Raspberry Pi 5 ke PCA9685

Konfigurasi PCA9685 ke Servo MG996R

Mekanisme pan-tilt dibangun menggunakan dua servo motor dengan karakteristik yang berbeda sesuai fungsi mekaniknya. Servo *continuous* digunakan untuk sumbu pan (horizontal) karena mampu berputar secara kontinu hingga 360 derajat, sehingga cocok untuk melakukan pemindaian area secara menyapu tanpa batas sudut. Sementara itu, servo standar MG996R (180 derajat) digunakan untuk sumbu tilt (vertikal) karena membutuhkan kontrol sudut yang presisi dan terbatas pada rentang posisi tertentu, misalnya 0° , 45° , 90° , 135° , dan 180° , untuk mengatur elevasi kamera secara stabil.

Kedua servo memperoleh suplai daya dari jalur 5V eksternal yang berasal dari catu daya PSU, bukan dari pin 5V Raspberry Pi 5. Pemisahan jalur daya ini penting untuk mencegah lonjakan arus saat servo bergerak atau mengalami beban tinggi, yang berpotensi mengganggu stabilitas sistem atau merusak regulator pada Raspberry Pi. Setiap pin output PCA9685 menyediakan tiga jalur utama, yaitu V+ sebagai suplai daya servo, GND sebagai *ground*, dan PWM sebagai sinyal kendali. PCA9685 bekerja pada frekuensi 50 Hz, dengan periode sekitar 20 ms. Pada servo standar, lebar pulsa menentukan sudut gerak, sedangkan pada servo *continuous*, sinyal PWM digunakan untuk mengatur arah dan kecepatan putaran melalui nilai *throttle*.



Gambar 3. 22 Konfigurasi PCA9685 ke Servo MG996R

Konfigurasi Audio

Subsistem audio pada sistem ini menggunakan speaker aktif yang terhubung ke Raspberry Pi 5 melalui konverter USB DAC. Penggunaan USB DAC diperlukan karena Raspberry Pi 5 tidak menyediakan jack audio 3.5 mm yang dapat digunakan secara langsung. USB DAC berfungsi mengubah sinyal audio digital dari sistem operasi menjadi sinyal analog yang dapat diterima oleh speaker aktif.



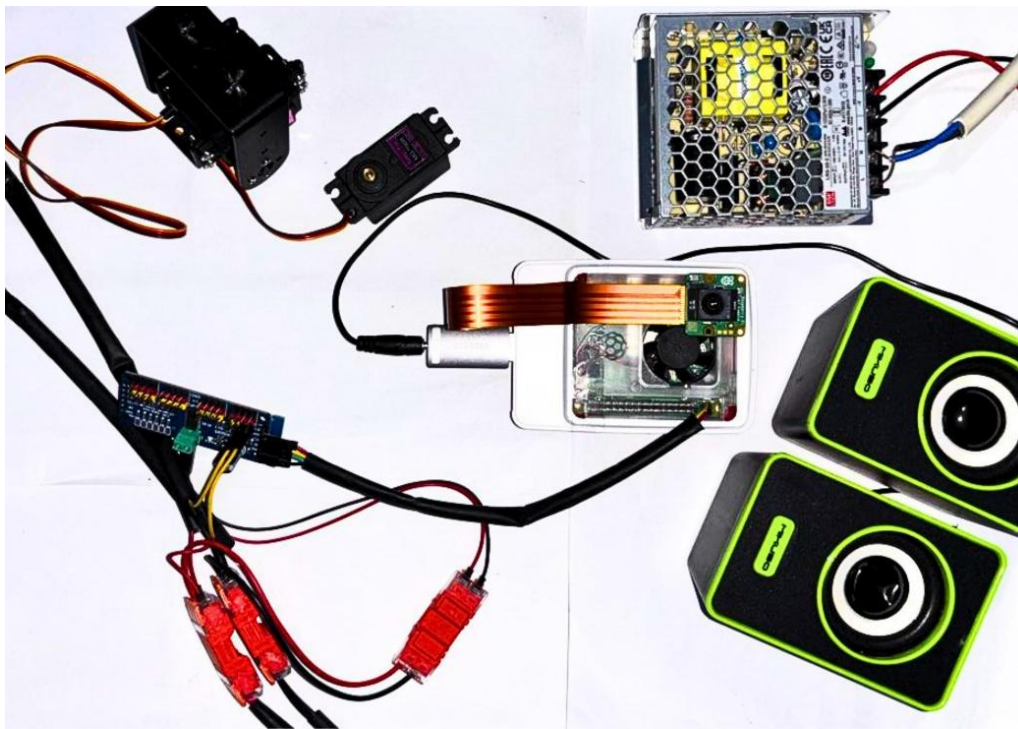
Gambar 3.23 Konfigurasi Audio

Konfigurasi Catu Daya

PSU 5V 6A digunakan untuk menyuplai dua servo MG996R. Setiap servo dapat menarik arus sekitar 1 hingga 3 A saat bergerak, terutama ketika mendapat beban. Jika dua servo aktif secara bersamaan, kebutuhan arus maksimum dapat mendekati 6 A, yang masih berada pada batas kapasitas PSU. Namun, dalam kondisi nyata, servo tidak bekerja terus-menerus pada arus puncak. Pergerakan servo hanya terjadi dalam waktu singkat, sehingga arus rata-rata yang digunakan lebih rendah. Selain itu, sistem dirancang agar kedua servo tidak aktif bersamaan

dalam durasi lama untuk menghindari lonjakan arus. Dengan kondisi tersebut, PSU 5V 6A masih mencukupi untuk kebutuhan sistem dan dapat bekerja stabil saat digunakan di lapangan.

Terminal blok berfungsi sebagai titik distribusi yang memisahkan jalur daya secara fisik. Dua rel pada terminal blok rel V+ dan rel GND menerima output PSU dan mendistribusikannya ke komponen tujuan yang berbeda. Yang paling penting adalah pemastian *common ground*: seluruh *ground* dari Raspberry Pi 5, PCA9685, dan servo harus terhubung ke titik yang sama di terminal blok agar referensi tegangan bus I2C dan sinyal PWM konsisten. Jika *ground* tidak sama maka sinyal logik 3.3V dari RPi 5 bisa terbaca sebagai tegangan berbeda oleh PCA9685 dan menyebabkan komunikasi I2C tidak stabil.

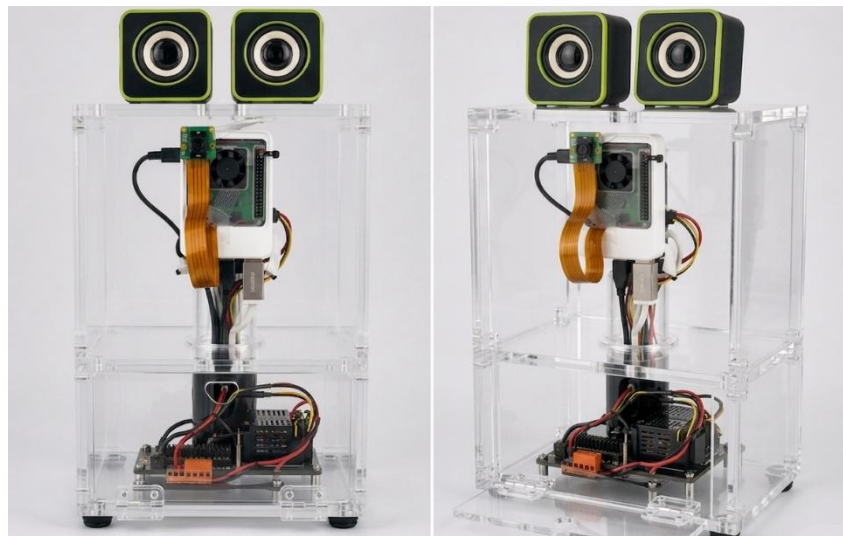


Gambar 3. 24 Rangkaian *Hardware* Alat

Perancangan Box

Perancangan box pada sistem ini dilakukan sebagai struktur utama untuk menempatkan komponen pendukung seperti sistem catu daya, distribusi listrik, modul kontrol, dan perangkat audio. Pada tahap pengembangan akhir, box dirancang menggunakan material akrilik transparan dengan struktur bertingkat. Pemilihan akrilik dilakukan karena material ini lebih kaku, rapi, dan mudah dibentuk dibandingkan rancangan prototipe awal, sekaligus memungkinkan posisi komponen internal tetap terlihat selama proses instalasi dan pengujian. Bagian bawah box digunakan untuk menempatkan komponen elektronika pendukung seperti rangkaian distribusi daya, terminal blok, dan modul pengendali, sedangkan bagian atas

dimanfaatkan sebagai area penempatan modul kamera, Raspberry Pi, dan speaker. Desain ini membuat susunan perangkat lebih terorg



Gambar 3.25 Rancangan Box

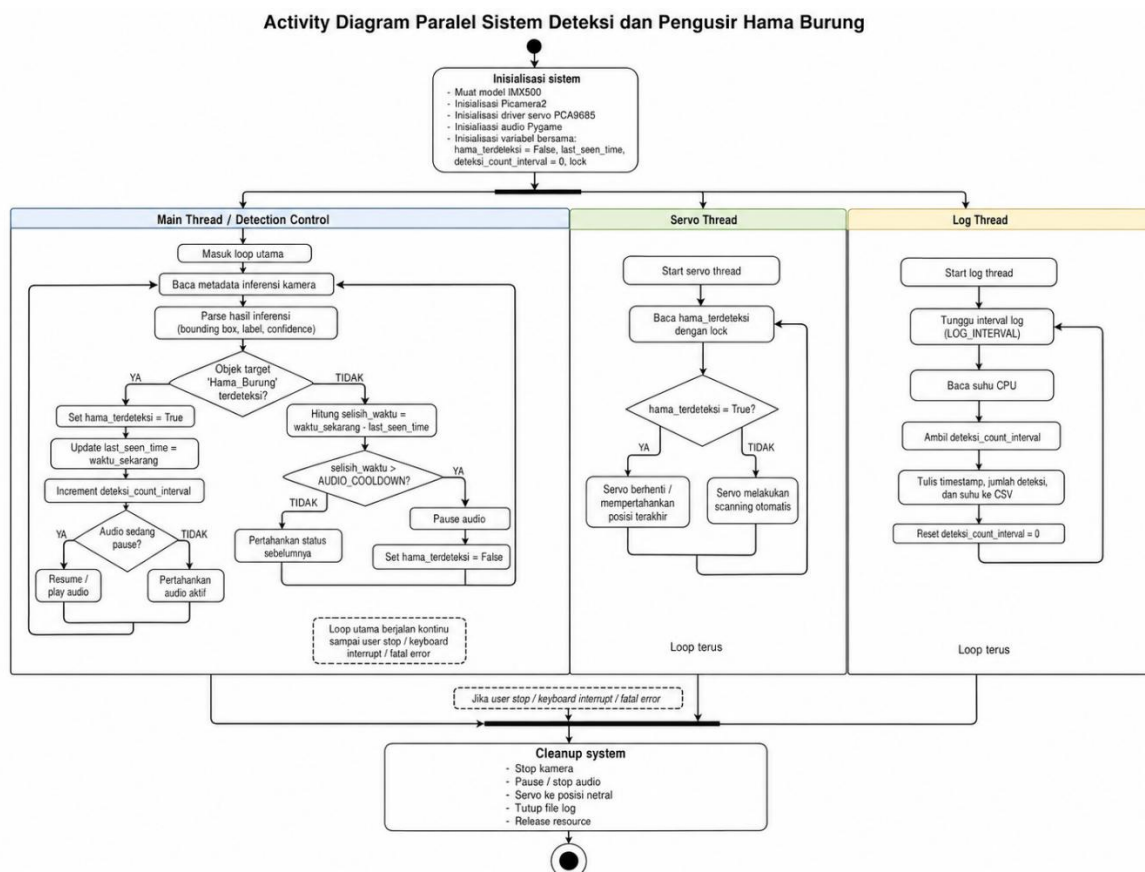
Pada bagian tengah box, ditambahkan strukturudukan untuk menempatkan mekanisme pan-tilt berbasis dual servo. Servo kontinu digunakan sebagai penggerak horizontal, sedangkan servo standar digunakan untuk mengatur pergerakan vertikal kamera. Modul Raspberry Pi AI Camera dan perangkat pemrosesan ditempatkan pada dudukan akrilik sehingga arah kamera dapat menyesuaikan area pemantauan. Sementara itu, speaker ditempatkan pada bagian atas box agar keluaran suara tidak terhalang oleh struktur enclosure. Penggunaan akrilik juga memberikan perlindungan fisik yang lebih baik terhadap komponen dibandingkan rancangan terbuka sebelumnya. Namun, karena sistem ini masih berupa prototipe, perlindungan terhadap faktor lingkungan seperti hujan, debu, dan kelembapan belum dapat diklaim optimal tanpa penambahan seal, ventilasi, dan pelindung luar yang lebih sesuai untuk penggunaan lapangan jangka panjang.

3.4.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak sistem ini didasarkan pada arsitektur berbasis skrip Python, di mana setiap subsistem dijalankan dalam *thread* terpisah namun tetap terintegrasi melalui mekanisme *shared state*. Sistem terdiri atas tiga komponen utama, yaitu modul deteksi berbasis kamera, modul kendali aktuator servo, dan modul pencatatan log. Modul deteksi berjalan pada *thread* utama dan bertanggung jawab membaca metadata inferensi dari sensor IMX500, sedangkan modul servo dan log dijalankan sebagai *background thread* yang terus memantau perubahan status sistem. Komunikasi antar komponen dilakukan melalui variabel global yang

dilindungi menggunakan mekanisme *lock* untuk menjaga konsistensi data saat diakses secara bersamaan.

Alur kerja sistem dimulai dari proses inisialisasi yang mencakup pemuatan model jaringan ke perangkat IMX500, konfigurasi kamera menggunakan *library* Picamera2, inisialisasi *driver* servo PCA9685 melalui antarmuka I2C, serta inisialisasi sistem audio menggunakan *library* Pygame. Setelah seluruh subsistem aktif, sistem masuk ke *loop* utama yang berjalan secara kontinu. Pada tahap ini, kamera secara berkala membaca metadata hasil inferensi yang berisi informasi *bounding box*, label objek, dan nilai *confidence*. Data tersebut kemudian diproses untuk menentukan apakah terdapat objek target dengan label “Hama_Burung” yang memenuhi ambang batas deteksi. Jika objek target terdeteksi, sistem memperbarui status hama menjadi aktif, mengaktifkan atau melanjutkan pemutaran audio, serta menambah jumlah deteksi pada interval berjalan. Jika objek target tidak lagi terdeteksi, sistem tidak langsung mengubah kondisi respons, tetapi menghitung selisih waktu sejak deteksi terakhir untuk memastikan bahwa perubahan status benar benar stabil.



Gambar 3.26 Activity Diagram Paralel Sistem Deteksi dan Pengusir Hama Burung

Gambar 3.11 menyajikan activity diagram paralel sistem deteksi dan pengusir hama burung. Diagram ini digunakan untuk menggambarkan alur kerja perangkat lunak yang

berjalan secara bersamaan melalui tiga bagian utama, yaitu *Main Thread / Detection Control*, *Servo Thread*, dan *Log Thread*. Pada *Main Thread / Detection Control*, sistem membaca metadata inferensi kamera, memproses hasil deteksi, dan menentukan respons berdasarkan keberadaan objek target. Sementara itu, *Servo Thread* membaca status deteksi untuk menentukan apakah servo mempertahankan posisi atau melakukan scanning otomatis, sedangkan *Log Thread* mencatat jumlah deteksi dan suhu CPU secara periodik. Dengan pemisahan ini, hubungan antara proses deteksi, aktuasi, dan pencatatan log dapat divisualisasikan secara lebih jelas serta selaras dengan rancangan perangkat lunak berbasis thread yang digunakan pada sistem.

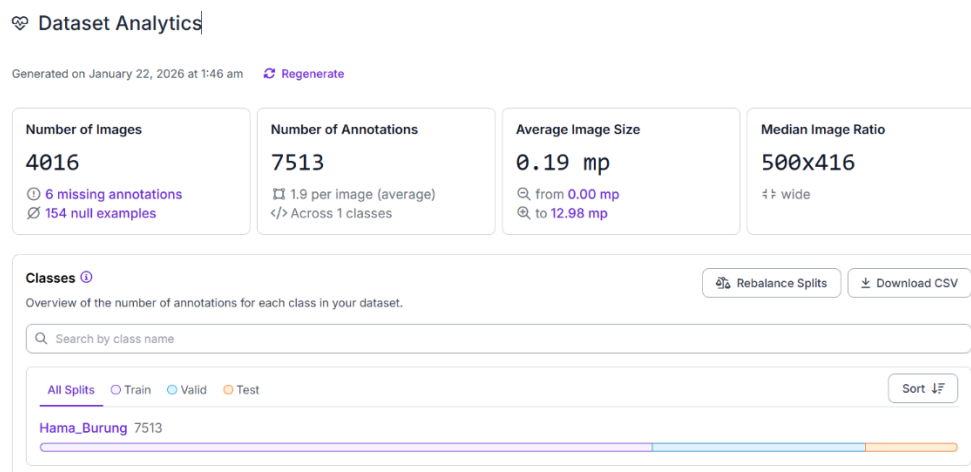
3.5 Pengumpulan Dataset

3.5.1 Sumber dan Proses Pengumpulan Data

Dataset yang digunakan untuk melatih model deteksi objek dalam penelitian ini dikumpulkan dari beberapa sumber untuk memastikan representasi yang beragam dan memadai. Sumber pertama adalah platform anotasi dan manajemen dataset Roboflow, yang digunakan sebagai repositori utama untuk mengorganisir, menganotasi, dan memproses dataset. Gambar dikumpulkan secara mandiri melalui pencarian gambar di internet dengan kata kunci yang relevan seperti "burung pipit sawah", "burung bondol padi", "*sparrow rice field*", dan variasi lainnya dalam Bahasa Indonesia maupun Bahasa Inggris. Sumber kedua adalah pengambilan cuplikan layar dari video rekaman yang diperoleh dari YouTube dan sumber video lainnya, di mana video tersebut menampilkan burung di lingkungan persawahan. Kombinasi kedua sumber ini bertujuan untuk menghasilkan dataset yang mencakup variasi kondisi pencahayaan, jarak, dan latar belakang yang beragam.

Dataset yang dikumpulkan terdiri dari dua kelas utama. Kelas pertama adalah "Hama_Burung", yang mencakup semua jenis burung yang berpotensi menjadi hama di lahan padi, termasuk burung pipit, bondol, gereja, dan jenis burung kecil lainnya yang lazim ditemukan di area persawahan. Kelas kedua adalah kelas "*background*" atau "*null*", yang mencakup gambar yang menampilkan kondisi sawah tanpa kehadiran burung hama, termasuk orang-orangan sawah, petani yang sedang bekerja di lahan, burung predator seperti burung hantu atau elang yang tidak termasuk hama, serta gambar sawah kosong pada berbagai kondisi. Keberadaan kelas "*background*" berfungsi sebagai data negatif yang membantu model belajar membedakan objek target dari lingkungan sekitarnya, sehingga tingkat positif palsu dapat ditekan secara signifikan.

Proses seleksi dan kurasi gambar dilakukan secara manual untuk memastikan kualitas data yang digunakan dalam pelatihan. Gambar yang tidak jelas, buram, terlalu gelap, atau tidak relevan dengan konteks persawahan dibuang dari dataset. Selain itu, duplikasi gambar juga diperiksa dan dihilangkan untuk mencegah data *leakage* antara set pelatihan dan set pengujian. Setelah kurasi, distribusi kelas diperiksa untuk memastikan tidak terjadi ketidakseimbangan yang ekstrem antara kelas "Hama_Burung" dan kelas "background". Distribusi dataset sebelum augmentasi disajikan pada Gambar 3.12.



Gambar 3.27 Distribusi Dataset pada Platform Roboflow sebelum Augmentasi

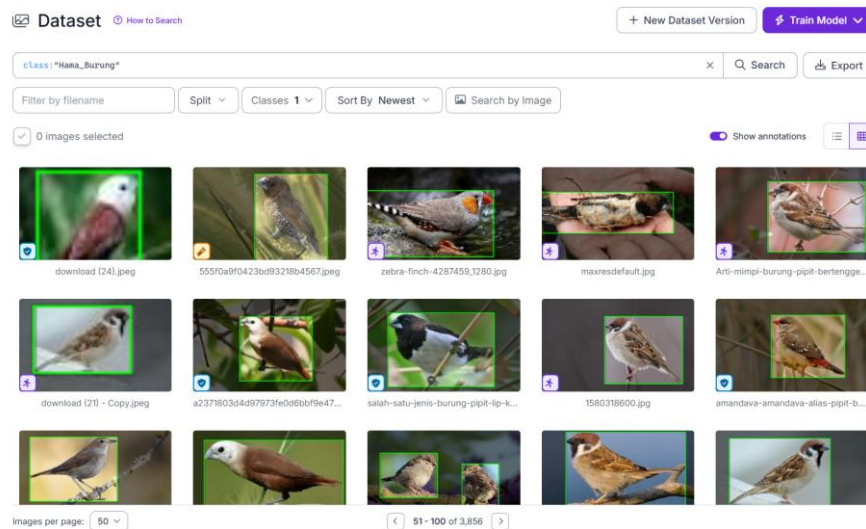
3.5.2 Anotasi Dataset

Proses anotasi dilakukan secara manual menggunakan alat anotasi berbasis web yang tersedia pada platform Roboflow. Setiap gambar dianotasi dengan membuat kotak pembatas (*bounding box*) yang mengelilingi setiap burung hama yang tampak dalam gambar. Format anotasi yang digunakan adalah format YOLO, di mana setiap *bounding box* direpresentasikan oleh lima nilai, yaitu indeks kelas (*class index*), koordinat titik tengah kotak pada sumbu-x (cx), koordinat titik tengah kotak pada sumbu-y (cy), lebar kotak (w), dan tinggi kotak (h), seluruhnya dalam bentuk nilai ternormalisasi terhadap dimensi gambar (rentang 0,0 hingga 1,0). Format ini secara langsung kompatibel dengan *framework* pelatihan Ultralytics YOLO yang digunakan dalam penelitian ini.

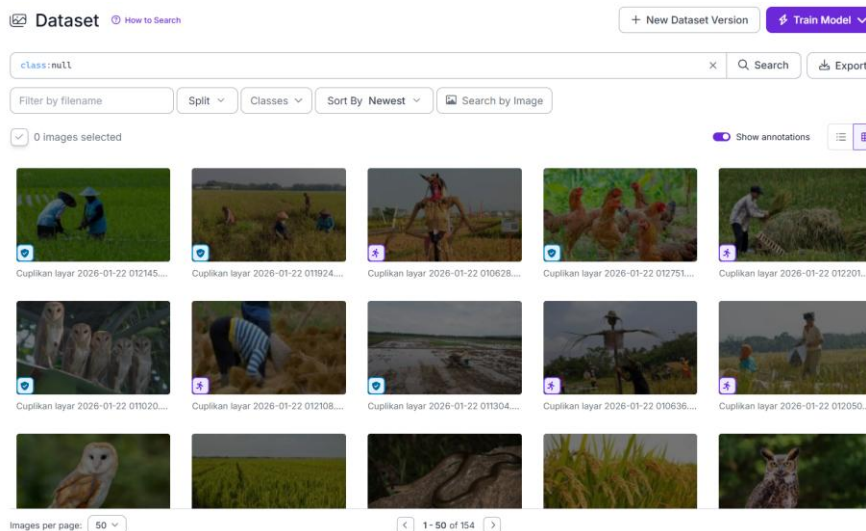
Pada gambar yang termasuk kelas "null" atau "background", tidak dilakukan penambahan anotasi *bounding box*. Gambar-gambar tersebut disertakan dalam dataset sebagai contoh negatif (*null examples*) yang memberikan informasi kepada model tentang latar belakang yang tidak mengandung objek target. Pendekatan ini merupakan praktik umum dalam pelatihan model YOLO untuk meningkatkan kemampuan model dalam menekan angka *false*

positive. Dalam statistik dataset Roboflow, gambar-gambar ini tercatat sebagai *null examples* yang dipisahkan dari gambar dengan anotasi aktif.

Contoh hasil anotasi pada kelas "Hama_Burung" disajikan pada Gambar 3.13 dan Gambar 3.14, yang menampilkan representasi gambar dengan *bounding box* pada kondisi berlatar sawah. Konsistensi anotasi dijaga dengan memastikan *bounding box* mencakup seluruh tubuh burung tanpa margin berlebihan, mengikuti panduan anotasi standar yang berlaku dalam komunitas deteksi objek berbasis YOLO.



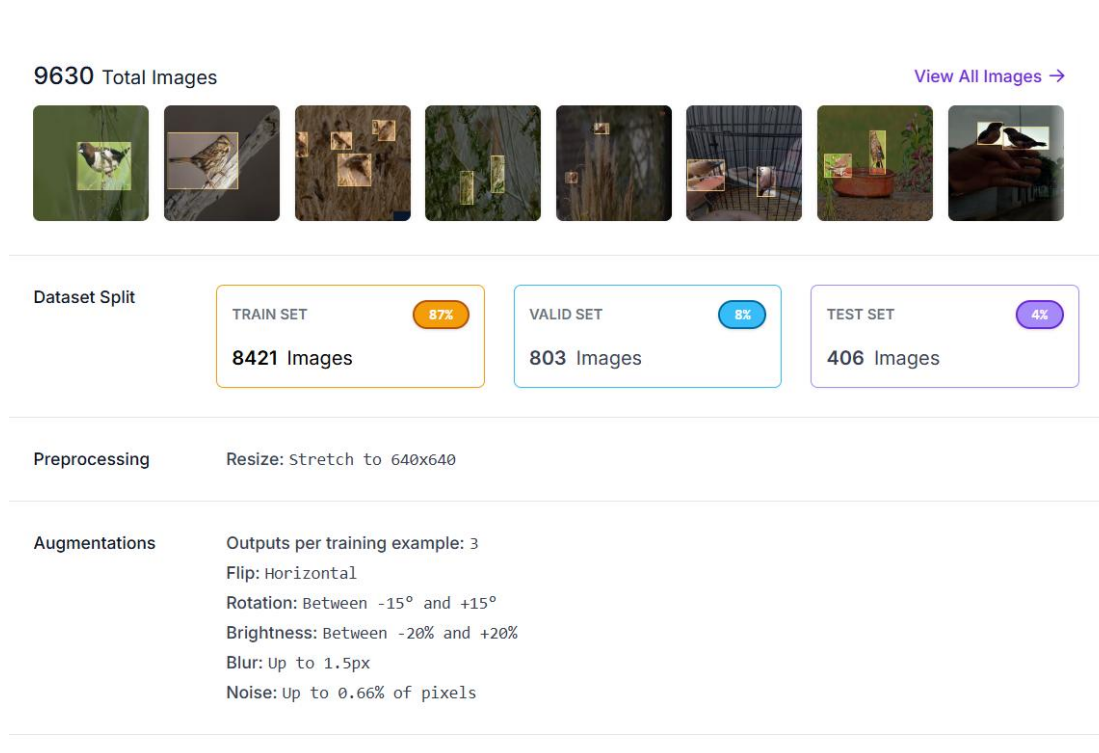
Gambar 3.28 Contoh Anotasi *Bounding box* pada Kelas Hama_Burung



Gambar 3.29 Contoh Gambar Kelas *Null* (*Background* tanpa Burung Hama)

3.5.3 Pembagian Dataset dan Augmentasi

Setelah proses anotasi selesai, dataset dibagi menjadi tiga subset mengikuti praktik standar dalam pembelajaran mesin, yaitu: set pelatihan (*training set*), set validasi (*validation set*), dan set pengujian (*test set*). Pembagian yang digunakan adalah 70% untuk pelatihan, 20% untuk validasi, dan 10% untuk pengujian. Set pelatihan digunakan untuk memperbarui bobot jaringan saraf selama proses *backpropagation*. Set validasi digunakan untuk memantau performa model pada data yang tidak dilihat selama pelatihan, sehingga dapat digunakan sebagai dasar pengambilan keputusan *early stopping* dan pemilihan model terbaik. Set pengujian digunakan secara eksklusif pada tahap evaluasi akhir untuk menghasilkan estimasi performa model yang tidak bias.



Gambar 3.15 Distribusi Dataset Setelah Augmentasi

Augmentasi data diterapkan pada set pelatihan menggunakan fitur augmentasi yang tersedia dalam platform Roboflow, sehingga set validasi dan set pengujian tetap bersih dan merepresentasikan kondisi data asli tanpa modifikasi artifisial. Augmentasi yang diterapkan mencakup beberapa transformasi geometris dan fotometrik, antara lain: *flip* horizontal, rotasi acak antara -15 hingga +15 derajat, penyesuaian kecerahan antara -20% hingga +20%, penambahan *noise* acak hingga 0,66% piksel, serta pengaburan hingga 1,5 piksel. Selain itu, *preprocessing* berupa *resize* dan *stretch* ke dimensi 640×640 piksel diterapkan secara konsisten pada seluruh subset untuk memastikan kompatibilitas dengan ukuran input yang diharapkan.

oleh model YOLO. Setiap sampel *training* diaugmentasi sebanyak tiga kali, sehingga jumlah efektif gambar pada set pelatihan meningkat tiga kali lipat dari data aslinya.

Penggunaan augmentasi data digunakan untuk meningkatkan variasi data yang dilihat oleh model selama pelatihan tanpa harus mengumpulkan dan menganotasi gambar baru secara manual. Augmentasi geometris membantu model menjadi lebih bervariasi terhadap perubahan orientasi dan posisi objek, sedangkan augmentasi fotometrik membantu model menjadi lebih robust terhadap variasi kondisi pencahayaan yang lazim ditemui di lingkungan pertanian luar ruangan. Penerapan augmentasi juga berfungsi sebagai teknik regularisasi yang dapat mengurangi risiko *overfitting*, terutama pada dataset dengan ukuran terbatas. Statistik dataset setelah proses augmentasi, termasuk jumlah gambar pada masing-masing subset, disajikan pada Gambar 3.15.

Tabel 3.13 Ringkasan Statistik Dataset

Keterangan	Sebelum Augmentasi	Setelah Augmentasi
Total Gambar	4.016	9.630
Kelas hama_burung (anotasi)	7.513 <i>bounding box</i>	± 22.539 <i>bounding box</i>
Kelas <i>Null/Background</i>	154 gambar	± 462 gambar (train)
Set Pelatihan (Train)	± 2.811 gambar	8.421 gambar
Set Validasi (Valid)	± 803 gambar	803 gambar
Set Pengujian (Test)	± 402 gambar	406 gambar
Resolusi Input Model	Berbagai ukuran	640×640 piksel (resize)

3.6 Pembuatan Model

3.6.1 Arsitektur Model yang Digunakan

Penelitian ini menggunakan dua arsitektur model dari keluarga YOLO sebagai objek perbandingan, yaitu YOLOv8n dan YOLO11n. Keduanya merupakan varian nano dari masing-masing generasi, yang dirancang untuk beroperasi pada perangkat dengan sumber daya komputasi terbatas. Model YOLOv8n dipilih sebagai *baseline* karena merupakan arsitektur yang telah banyak digunakan dan divalidasi dalam berbagai penelitian deteksi objek pada perangkat tertanam. Model YOLO11n dipilih sebagai model pembanding karena merupakan arsitektur yang lebih baru dengan klaim performa lebih baik dalam hal akurasi dan efisiensi komputasi dibandingkan generasi sebelumnya, sebagaimana dilaporkan dalam dokumentasi resmi Ultralytics.

YOLOv8n memiliki arsitektur yang terdiri dari 130 lapisan dengan total 3.011.043 parameter, menggunakan *backbone* berbasis C2f (*Cross Stage Partial Bottleneck*) dengan

Spatial Pyramid Pooling Fast (SPPF) dan *head* deteksi yang terpisah dari *backbone*. Model ini beroperasi pada ukuran input 640×640 piksel dengan kedalaman fitur yang lebih dalam dibandingkan varian sebelumnya seperti YOLOv5. YOLO11n memiliki arsitektur yang terdiri dari 182 lapisan dengan 2.590.035 parameter, menggunakan blok C3k2 (*Cross Stage Partial with 2 bottlenecks*) dan modul C2PSA (*Parallel Spatial Attention*) yang memberikan kemampuan fokus pada fitur-fitur spasial yang relevan. Meskipun memiliki lebih banyak lapisan, YOLO11n justru memiliki jumlah parameter yang lebih sedikit dari YOLOv8n, yang menunjukkan efisiensi arsitektur yang lebih baik dalam hal kompresi model.

Kedua model diinisialisasi menggunakan bobot *pretrained* yang dilatih pada dataset COCO (*Common Objects in Context*), sebuah dataset deteksi objek skala besar yang mencakup 80 kelas objek umum. Penggunaan bobot *pretrained* memungkinkan proses *transfer learning*, di mana representasi fitur visual tingkat rendah yang telah dipelajari dari dataset besar dapat dimanfaatkan ulang untuk tugas deteksi burung hama yang lebih spesifik, sehingga mengurangi kebutuhan data *training* dan mempercepat konvergensi model. Konfigurasi jumlah kelas (nc) dalam file data.yaml disesuaikan menjadi 1 (satu kelas hama_burung), menggantikan nilai default 80 kelas dari COCO.

3.6.2 Konfigurasi dan Proses Pelatihan Model

Proses pelatihan model dilakukan menggunakan *framework* Ultralytics YOLO versi 8.4.14 pada lingkungan Python 3.11.14 dengan PyTorch 2.5.1 yang dikonfigurasi untuk memanfaatkan akselerasi GPU NVIDIA melalui CUDA. Perangkat yang digunakan untuk pelatihan adalah komputer dengan GPU NVIDIA GeForce RTX 3050 6GB, yang menyediakan kapasitas memori grafis yang cukup untuk menjalankan pelatihan dengan *batch size* yang optimal. Pelatihan tidak dilakukan langsung di Raspberry Pi 5 karena keterbatasan komputasi perangkat tersebut untuk proses *training*, melainkan dilakukan di lingkungan komputasi PC terlebih dahulu.

Parameter pelatihan dikonfigurasi melalui argumen command-line yang dimasukkan ke dalam skrip yolo_train.py. Parameter utama yang dikonfigurasi meliputi: jumlah *epoch* maksimum sebesar 500 *epoch*, ukuran input gambar 640×640 piksel, inisialisasi model dengan bobot *pretrained* (yolov8n.pt atau yolo11n.pt), nama eksperimen untuk pengelompokan hasil, serta pemilihan *device* GPU (*device=0*). *Batch size* tidak ditentukan secara eksplisit, melainkan diserahkan kepada mekanisme *AutoBatch* yang secara otomatis menentukan *batch size* optimal berdasarkan memori GPU yang tersedia, dengan target utilisasi GPU sebesar 84–90%. Parameter ini dipilih untuk memaksimalkan penggunaan sumber daya komputasi yang tersedia

sambil menjaga stabilitas pelatihan. Seluruh konfigurasi parameter pelatihan dirangkum dalam Tabel 3.5.

Tabel 3.14 Konfigurasi Parameter Pelatihan Model YOLO

Parameter	YOLOv8n	YOLO11n	Keterangan
Framework	Ultralytics 8.4.14	Ultralytics 8.4.14	Versi <i>framework</i> yang digunakan
Bobot awal (pretrained)	yolov8n.pt	yolo11n.pt	<i>Transfer learning</i> dari COCO
Jumlah kelas (nc)	1	1	hama_burung
Epoch maksimum	500	500	Dengan <i>early stopping</i> <i>patience</i> =100
Ukuran input	640×640 px	640×640 px	Diresize saat preprocessing
Batch size	AutoBatch (~15)	AutoBatch (~13)	Disesuaikan dengan GPU 6GB
Optimizer	MuSGD (auto)	MuSGD (auto)	Ditentukan otomatis oleh Ultralytics
Learning rate (lr0)	0,01	0,01	Awal training
Augmentasi lanjutan	Mosaic, flipud, fliplr, hsv	Mosaic, flipud, fliplr, hsv	Augmentasi <i>online</i> saat training
Early stopping patience	100 epoch	100 epoch	Hentikan jika tidak ada peningkatan
Format export	ONNX + IMX	ONNX + IMX	Untuk <i>deployment</i> di AI Camera

Mekanisme *early stopping* merupakan salah satu fitur penting dalam konfigurasi pelatihan ini. *Early stoppng* bekerja dengan memantau metrik validasi, khususnya mAP50-95, setiap *epoch*. Jika dalam 100 *epoch* berturut-turut tidak terjadi peningkatan pada metrik validasi terbaik, pelatihan dihentikan secara otomatis dan model terbaik yang pernah dicapai disimpan. Mekanisme ini mencegah model mengalami *overfitting* akibat pelatihan yang terlalu lama, sekaligus menghemat waktu komputasi dengan tidak melanjutkan pelatihan ketika model sudah tidak mengalami kemajuan yang berarti.

Proses pelatihan menghasilkan beberapa komponen utama, yaitu: (1) file bobot model terbaik (best.pt) dan bobot terakhir (last.pt); (2) grafik kurva loss pelatihan dan validasi yang mencakup *box loss*, *classification loss*, dan *DFL loss*; (3) grafik metrik performa yang mencakup *precision*, *recall*, mAP50, dan mAP50-95 per *epoch*; (4) *confusion matrix* yang menunjukkan distribusi prediksi benar dan salah; serta (5) kurva *Precision-Recall* (PR curve) dan kurva *F1-Confidence*. Seluruh komponen ini disimpan secara otomatis oleh Ultralytics ke dalam direktori runs/detect/ dengan nama eksperimen yang telah dikonfigurasi. Setelah pelatihan selesai, model diekspor ke format ONNX menggunakan perintah ekspor bawaan Ultralytics. File ONNX ini merupakan format representasi model yang bersifat *framework-agnostic*, sehingga dapat dikonversi lebih lanjut ke format yang kompatibel dengan *hardware*

akselerator khusus. Proses ekspor menggunakan opset ONNX versi 11 dengan aplikasi kompresi menggunakan *library onnxslim* untuk mengurangi ukuran file model. Hasil ekspor menghasilkan file *best.onnx* yang siap untuk tahap konversi selanjutnya.

3.7 Rancangan Pengujian

Rancangan pengujian dalam penelitian ini disusun untuk mengevaluasi sistem dari berbagai sudut pandang, mulai dari fungsionalitas perangkat lunak, performa model kecerdasan buatan, efektivitas sistem di lapangan, hingga penerimaan pengguna akhir. Secara keseluruhan, pengujian dibagi menjadi empat kategori utama, yaitu: (1) pengujian fungsional *Black Box*, (2) pengujian performa model, (3) pengujian lapangan, dan (4) pengujian penerimaan pengguna (UAT). Setiap kategori memiliki tujuan, skenario, dan indikator keberhasilan yang berbeda.

3.7.1 Pengujian Performa Model

Pengujian performa model dilakukan untuk mengevaluasi kualitas model deteksi objek yang telah dilatih menggunakan metrik standar yang lazim digunakan dalam komunitas deteksi objek. Metrik utama yang digunakan adalah: (1) *Precision* (P), yaitu proporsi deteksi positif yang benar-benar positif dari seluruh deteksi positif yang dihasilkan model; (2) *Recall* (R), yaitu proporsi objek positif yang berhasil dideteksi dari seluruh objek positif yang sebenarnya ada; (3) mAP50, yaitu *mean Average Precision* pada ambang batas IoU (*Intersection over Union*) sebesar 0,50; serta (4) mAP50-95, yaitu rata-rata mAP pada berbagai ambang batas IoU dari 0,50 hingga 0,95 dengan interval 0,05.

Perbandingan performa antara YOLOv8n dan YOLO11n akan disajikan pada bab hasil dan pembahasan. Namun, aspek metodologi pengujian yang perlu ditegaskan di sini adalah bahwa evaluasi dilakukan pada kondisi yang sepenuhnya identik untuk kedua model, meliputi dataset pengujian yang sama, nilai *threshold confidence* yang sama (0,50), dan *platform hardware* yang sama. Hal ini penting untuk memastikan bahwa perbandingan yang dihasilkan bersifat adil dan dapat dijadikan dasar pengambilan keputusan dalam memilih model terbaik untuk diimplementasikan pada tahap *deployment*.

3.7.2 Pengujian Fungsional (*Black Box Testing*)

Pengujian *Black Box* dilakukan untuk memverifikasi bahwa setiap fungsi dan fitur sistem menghasilkan keluaran yang sesuai dengan spesifikasi yang telah ditetapkan, tanpa memperhatikan implementasi internal dari fungsi tersebut. Pendekatan ini berfokus pada antarmuka input-output dari setiap komponen sistem. Skenario pengujian *Black Box* mencakup serangkaian kasus uji yang dirancang untuk menguji setiap fungsi utama sistem, termasuk

inisialisasi kamera, pemuatan model ke IMX500, deteksi objek dari gambar diam, deteksi objek dari video, aktivasi servo saat objek terdeteksi, pemutaran audio saat objek terdeteksi, penghentian audio saat objek menghilang, dan penanganan kondisi kesalahan seperti kegagalan inisialisasi.

Setiap kasus uji *Black Box* didefinisikan dengan format yang terdiri dari: nomor kasus uji, deskripsi fungsi yang diuji, kondisi awal, masukan yang diberikan, keluaran yang diharapkan, keluaran yang sebenarnya diperoleh, dan status hasil pengujian (berhasil/gagal). Format pengujian *Black Box* disajikan pada Tabel 3.6. Pengujian *Black Box* dilakukan sebagai bagian dari pengujian unit sistem sebelum pengujian lapangan dilaksanakan.

Tabel 3.15 Format Rancangan Kasus Uji *Black Box* Testing

No.	Fungsi yang Diuji	Kondisi Awal	Input	Expected Output	Status
BB-01	Inisialisasi kamera & model	Sistem dinyalakan	Program dijalankan	Kamera aktif, model termuat, tidak ada error	-
BB-02	Deteksi burung dari gambar	Sistem aktif	Gambar berisi burung	<i>Bounding box</i> muncul, label hama_burung	-
BB-03	Aktivasi audio saat deteksi	Sistem aktif, audio ready	Burung terdeteksi	Speaker membunyikan suara pengusir	-
BB-04	Penghentian audio saat tidak ada deteksi	Audio aktif	Tidak ada objek $\geq$ <i>cooldown</i>	Audio dijeda (pause)	-
BB-05	Gerakan servo saat deteksi	Servo terhubung	Sinyal deteksi aktif	Servo bergerak sesuai skenario	-
BB-06	Servo diam saat deteksi aktif	Servo sedang mindai	Burung terdeteksi, hama_terdeteksi= <i>True</i>	Servo berhenti bergerak, tetap di posisi terakhir	-
BB-07	Penanganan <i>confidence</i> rendah	Sistem aktif	Gambar burung dengan <i>confidence</i> < 0,50	Tidak ada <i>bounding box</i> , tidak ada respons audio/servo	-
BB-08	Pencatatan log suhu & deteksi	<i>Thread</i> log berjalan	Waktu berjalan melewati interval 15 menit	Baris baru tercatat di log_deteksi.csv dengan timestamp, suhu, jumlah deteksi	-
BB-09	Pengujian false positive (objek non-target)	Sistem aktif, kamera streaming	Gambar non-burung (misal manusia/objek lain) diarahkan ke kamera	Tidak ada bounding box “Hama_Burung”, tidak ada respons audio/servo	-

3.7.3 Pengujian Lapangan

Pengujian lapangan dilakukan dalam kondisi yang menyerupai lingkungan sawah sesungguhnya untuk mengevaluasi kinerja sistem dalam skenario operasional nyata. Pengujian ini mencakup beberapa subpengujian, yaitu: (1) pengujian deteksi jarak, untuk menentukan jarak efektif deteksi sistem terhadap objek burung; serta (2) pengujian waktu respon sistem, untuk mengukur waktu yang dibutuhkan sejak burung terdeteksi hingga suara pengusir berbunyi. Pengujian deteksi jarak dilakukan dengan menempatkan objek uji berupa burung asli di dalam sangkar pada berbagai jarak dari kamera, yaitu 50 cm, 100 cm, 150 cm, 200 cm, dan 250 cm. Setiap jarak diuji sebanyak 10 kali percobaan, dan sistem dikatakan berhasil mendeteksi jika objek terdeteksi dengan *confidence* di atas *threshold* pada minimal 7 dari 10 percobaan ($\geq 70\%$ *success rate*). Latar belakang yang digunakan adalah *backdrop* yang menampilkan gambar sawah atau area hijau untuk mensimulasikan kondisi di lapangan. Rancangan skenario pengujian lapangan disajikan pada Tabel 3.7.

Tabel 3.16 Rancangan Skenario Pengujian Lapangan

No.	Skenario Uji	Kondisi	Parameter Ukur	Indikator Keberhasilan
FL-01	Deteksi jarak 50 cm	Pencahayaan normal, latar sawah	<i>Success rate</i> , <i>confidence avg</i>	$\geq 70\%$ terdeteksi dari 10 percobaan
FL-02	Deteksi jarak 100 cm	Pencahayaan normal, latar sawah	<i>Success rate</i> , <i>confidence avg</i>	$\geq 70\%$ terdeteksi dari 10 percobaan
FL-03	Deteksi jarak 150 cm	Pencahayaan normal, latar sawah	<i>Success rate</i> , <i>confidence avg</i>	$\geq 70\%$ terdeteksi dari 10 percobaan
FL-04	Deteksi jarak 200 cm	Pencahayaan normal, latar sawah	<i>Success rate</i> , <i>confidence avg</i>	$\geq 50\%$ terdeteksi dari 10 percobaan
FL-05	Deteksi jarak 250 cm	Pencahayaan normal, latar sawah	<i>Success rate</i> , <i>confidence avg</i>	$\geq 50\%$ terdeteksi dari 10 percobaan
FL-06	FPS deteksi kamera	Semua jarak, kondisi normal	FPS rata-rata	FPS ≥ 10 fps
FL-07	Latensi respons audio	Objek muncul tiba-tiba	Waktu deteksi hingga audio berbunyi	≤ 2 detik

3.7.4 Penngujian Penerimaan Pengguna (UAT)

Pengujian penerimaan pengguna UAT dilakukan dengan melibatkan petani padi sebagai responden utama, karena merekalah pengguna akhir yang paling relevan untuk menilai apakah sistem yang dikembangkan benar-benar sesuai dengan kebutuhan dan mudah dipahami dalam

konteks pertanian. UAT dilaksanakan dengan cara mendemonstrasikan sistem secara langsung kepada responden di lapangan atau di lingkungan yang mensimulasikan kondisi sawah, kemudian responden diminta mengisi kuesioner terstruktur berbasis skala Likert. Pendekatan ini sejalan dengan metodologi UAT yang digunakan oleh (Hermansah *et al.*, 2025) di mana demonstrasi sistem langsung kepada pengguna akhir dilakukan sebelum pengisian kuesioner untuk memastikan responden memiliki pemahaman yang memadai terhadap fungsi sistem yang dinilai.

Kuesioner UAT terdiri dari 15 butir pernyataan kuantitatif yang diukur menggunakan Skala Likert 1 hingga 5, di mana nilai 1 mewakili respons "sangat tidak setuju" dan nilai 5 mewakili "sangat setuju". Butir-butir pernyataan dikelompokkan ke dalam lima dimensi evaluasi, yaitu: (1) Persepsi Manfaat (butir 1, 8, 9, 14, 15), yang menilai apakah responden mempersepsikan sistem sebagai solusi yang bermanfaat dan relevan; (2) Kinerja Deteksi (butir 2, 3, 5), yang menilai apakah kamera dan model deteksi bekerja dengan baik; (3) Kinerja Aktuator (butir 4), yang menilai apakah gerakan kamera membantu pengamatan; (4) Kinerja Audio (butir 6, 7), yang menilai efektivitas suara pengusir; serta (5) Kegunaan Praktis (butir 10, 11, 12, 13), yang menilai kemudahan pemahaman, kelayakan penggunaan, dan kepraktisan sistem. Rancangan kuesioner UAT disajikan pada Tabel 3.8.

Tabel 3.17 Rancangan Kuesioner UAT (Skala Likert 1–5)

No.	Pernyataan	Dimensi
1	Sistem ini membantu mendeteksi hama burung di sawah dengan baik	Persepsi Manfaat
2	Hasil deteksi pada alat ini terlihat cukup akurat	Kinerja Deteksi
3	Kamera pada alat ini mampu menangkap objek burung dengan jelas	Kinerja Deteksi
4	Gerakan kamera (servo) membantu memperluas area pengamatan	Kinerja Aktuator
5	Respons alat terhadap kehadiran burung terlihat cepat	Kinerja Deteksi
6	Suara pengusir yang diputar cukup efektif untuk mengganggu burung	Kinerja Audio
7	Variasi suara yang digunakan membuat alat tidak mudah ditebak burung	Kinerja Audio
8	Sistem ini lebih baik daripada cara manual yang biasa digunakan	Persepsi Manfaat
9	Alat ini dapat mengurangi kebutuhan petani untuk berjaga di sawah	Persepsi Manfaat
10	Alat ini mudah dipahami cara kerjanya setelah didemonstrasikan	Kegunaan Praktis
11	Alat ini layak digunakan di lahan pertanian nyata	Kegunaan Praktis
12	Desain alat ini sudah sesuai dengan kebutuhan pengusiran hama burung	Kegunaan Praktis
13	Penggunaan daya dan sumber listrik alat ini terlihat praktis	Kegunaan Praktis
14	Sistem ini berpotensi membantu mengurangi kerugian akibat hama burung	Persepsi Manfaat
15	Saya bersedia menggunakan alat seperti ini bila tersedia di lahan saya	Persepsi Manfaat

Analisis data UAT dilakukan secara kuantitatif dengan menghitung nilai rata-rata (*mean*) per butir dan per dimensi menggunakan rumus berikut (Aliyah *et al.*, 2024; Hermansah *et al.*, 2025):

$$Mean = \frac{\text{Jumlah Bobot Penilaian Responden}}{\text{Total Responden}} \quad (3.1)$$

$$\text{Persentase}(\%) = \left(\frac{\text{Nilai Mean}}{\text{Bobot Maksimum}} \right) \times 100\% \quad (3.2)$$

Di mana bobot maksimum adalah 5 (nilai tertinggi skala Likert yang digunakan). Nilai mean yang diperoleh dari Persamaan 3.1 kemudian diinterpretasikan berdasarkan kategorisasi rentang interval yang diturunkan dari formula berikut:

$$\text{Lebar Interval} = \frac{(\text{Nilai Maks} - \text{Nilai Min})}{\text{Jumlah Kategori}} = \frac{(5 - 1)}{5} = 0,80 \quad (3.3)$$

Hasil perhitungan dari Persamaan 3.3 menghasilkan lebar interval sebesar 0,80 untuk setiap kategori. Berdasarkan lebar interval tersebut, kategorisasi interpretasi nilai rata-rata UAT ditetapkan sebagaimana disajikan pada Tabel 3.9.

Tabel 3.18 Kategori Interpretasi Nilai Mean UAT

Rentang Nilai Mean	Kategori
4,21 – 5,00	Sangat Baik
3,41 – 4,20	Baik
2,61 – 3,40	Cukup
1,81 – 2,60	Tidak Baik
1,00 – 1,80	Sangat Tidak Baik

Kategorisasi pada Tabel 3.9 digunakan sebagai acuan utama dalam menginterpretasikan seluruh hasil UAT pada Bab IV. Konvensi kategorisasi ini konsisten dengan yang diterapkan dalam penelitian-penelitian UAT sistem informasi dan teknologi terapan sebelumnya

(Hermansah *et al.*, 2025; Kusuma & Yufron, 2024), sehingga memungkinkan perbandingan dan validasi lintas penelitian.

Selain kuesioner kuantitatif, sesi UAT juga dilengkapi dengan bagian evaluasi kualitatif berupa wawancara singkat semi-terstruktur pascademonstrasi. Petani diminta memberikan pendapat tentang kelebihan utama alat, bagian yang perlu diperbaiki, pendapat tentang efektivitas suara pengusir, pendapat tentang gerakan kamera, serta saran perbaikan lainnya. Data kualitatif ini dianalisis secara deskriptif untuk melengkapi interpretasi data kuantitatif kuesioner Likert, sehingga gambaran penerimaan pengguna terhadap sistem dapat dipahami secara lebih komprehensif.

Jumlah responden UAT ditentukan secara *purposive* berdasarkan ketersediaan petani di wilayah penelitian yang bersedia mengikuti sesi demonstrasi dan pengisian kuesioner. Kriteria inklusi responden meliputi: (1) memiliki pengalaman bertani padi minimal dua tahun, dan (2) memiliki pengalaman langsung menghadapi permasalahan hama burung. Data demografis responden mencakup usia, jenis kelamin, lama bertani, dan luas lahan yang dikelola.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Pelatihan Model

4.1.1 Implementasi Skrip Pelatihan

Pelatihan kedua model dilakukan pada mesin komputasi yang dilengkapi GPU NVIDIA GeForce RTX 3050 dengan kapasitas VRAM sebesar 6 GB, menggunakan CUDA 12.1, Python 3.11.14, PyTorch 2.5.1, dan Ultralytics 8.4.14. Seluruh konfigurasi pelatihan dikendalikan melalui *command-line arguments* pada terminal, sehingga kedua model dapat dilatih menggunakan *pipeline* yang identik hanya dengan mengganti parameter bobot awal. Pendekatan ini meminimalkan intervensi manual dan menjamin reproduksibilitas eksperimen.

Inti dari skrip pelatihan terdiri atas fungsi *parse_arguments()* yang mendefinisikan seluruh parameter konfigurasi, dan fungsi *main()* yang mengeksekusi pelatihan. Parameter utama yang relevan meliputi *--config* untuk menentukan dataset, *--init_model* untuk memilih bobot awal, *--epochs* untuk membatasi *epoch* maksimum, dan *--device* untuk menentukan GPU yang digunakan. Parameter batch secara sengaja diatur ke 0.9 untuk mengaktifkan mekanisme AutoBatch pada Ultralytics, yaitu fitur yang menghitung ukuran *batch* optimal secara otomatis berdasarkan 90% utilisasi memori GPU. Mekanisme ini menghasilkan batch size 15 untuk YOLOv8n dan 13 untuk YOLOv11n pada konfigurasi GPU yang digunakan.

Alur implementasi ini dibagi menjadi dua modul utama: pengatur parameter (*parse_arguments*) dan eksekutor pelatihan (*main*).

```
def parse_arguments():
    parser = argparse.ArgumentParser(description="YOLO trainer")
    parser.add_argument("--config", type=str, default="config.yaml")
    parser.add_argument("--init_model", type=str, default="yololln.pt")
    parser.add_argument("--epochs", type=int, default=20)
    parser.add_argument("--device", type=int, default=0)
    parser.add_argument("--image_size", type=str, default="640x640")
    return parser.parse_args()

def main():
    args = parse_arguments()
    model = YOLO(args.init_model)

    image_h, image_w = map(int, args.image_size.split('x'))
    image_size = [image_h, image_w]

    model.train(
        data=args.config,
        epochs=args.epochs,
        imgsz=image_size,
        device=args.device,
        name=args.name,
        batch=-1  # Mengaktifkan AutoBatch untuk optimasi VRAM
```

)

Gambar 4.22 Kode Program Fungsi *parse_arguments()* dan Potongan Fungsi *main()*

Pelatihan dijalankan dengan konfigurasi *epoch* maksimum 500 untuk kedua model. Namun, mekanisme *early stopping* dengan *patience* 100 *epoch* menghentikan pelatihan lebih awal apabila tidak terdapat peningkatan pada metrik validasi selama 100 *epoch* berturut-turut. Hal ini bertujuan menghindari pemborosan komputasi sekaligus mencegah *overfitting* pada tahap akhir pelatihan.

```
# YOLOv8n
python training/yolo_train.py \
  --config data.yaml \
  --init_model yolov8n.pt \
  --epochs 500 \
  --device 0

# YOLOv11n
python training/yolo_train.py \
  --config data.yaml \
  --init_model yolov11n.pt \
  --epochs 500 \
  --device 0
```

Gambar 4.23 *command-line arguments*

```
Speed: 0.2ms preprocess, 2.5ms inference, 0.0ms loss, 0.8ms postprocess per image
Results saved to D:\Ara\rasp pi deteksi burung\runs\detect\deteksi_burung9
ultralytics 8.4.14 Python-3.11.14 torch-2.5.1+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 6GB Laptop GPU, 6144MiB)
Model summary (fused): 73 layers, 3,005,843 parameters, 0 gradients, 8.1 GFLOPs

PyTorch: starting from 'D:\Ara\rasp pi deteksi burung\runs\detect\deteksi_burung9\weights\best.pt' with input shape (1, 3, 640, 640) BCHW and output shape(s) (1, 300, 6) (6.0 MB)

ONNX: starting export with onnx 1.17.0 opset 11...
ONNX: slimming with onnxslim 0.1.82...
ONNX: export success 1.2s, saved as 'D:\Ara\rasp pi deteksi burung\runs\detect\deteksi_burung9\weights\best.onnx' (11.7 MB)

Export complete (1.4s)
Results saved to D:\Ara\rasp pi deteksi burung\runs\detect\deteksi_burung9\weights
Predict: yolo predict task=detect model=D:\Ara\rasp pi deteksi burung\runs\detect\deteksi_burung9\weights\best.onnx imgs=640
Validate: yolo val task=detect model=D:\Ara\rasp pi deteksi burung\runs\detect\deteksi_burung9\weights\best.onnx imgs=640 data=D:\Ara\Skripsi\Deteksi Burung Pemakan Padi\data.yaml
Visualize: https://netron.app
(ml_env) PS D:\Ara\rasp pi deteksi burung>
```

Gambar 4.24 Hasil Eksekusi Pelatihan YOLOv8n dan Parameternya

```

Speed: 0.2ms preprocess, 2.8ms inference, 0.0ms loss, 0.8ms postprocess per image
Results saved to D:\Aрга\rasp pi deteksi burung\runs\detect\deteksi_burung7
Ultralytics 8.4.14 Python-3.11.14 torch-2.5.1+cu121 CUDA:0 (NVIDIA GeForce RTX 3050 6GB Laptop GPU, 6144MiB)
YOLOv11n summary (fused): 101 layers, 2,582,347 parameters, 0 gradients, 6.3 GFLOPs

PyTorch: starting from 'D:\Aрга\rasp pi deteksi burung\runs\detect\deteksi_burung7\weights\best.pt' with input shape (1, 3, 640, 640) BCHW and output shape(s) (1, 300, 6) (5.3 MB)

ONNX: starting export with onnx 1.17.0 opset 11...
ONNX: slimming with onnxslim 0.1.82...
ONNX: export success 1.6s, saved as 'D:\Aрга\rasp pi deteksi burung\runs\detect\deteksi_burung7\weights\best.onnx' (10.1 MB)

Export complete (1.9s)
Results saved to D:\Aрга\rasp pi deteksi burung\runs\detect\deteksi_burung7\weights
Predict: yolo predict task=detect model=D:\Aрга\rasp pi deteksi burung\runs\detect\deteksi_burung7\weights\best.onnx imgsz=640
Validate: yolo val task=detect model=D:\Aрга\rasp pi deteksi burung\runs\detect\deteksi_burung7\weights\best.onnx imgsz=640 data=D:\Aрга\Skripsi\Deteksi Burung Pemakan Padi\data.yaml
Visualize: https://netron.app
(mL_env) PS D:\Aрга\rasp pi deteksi burung>

```

Gambar 4.25 Hasil Eksekusi Pelatihan YOLOv11n dan Parameternya

Berdasarkan hasil eksekusi yang ditunjukkan pada Gambar 4.3 dan Gambar 4.4, berikut adalah perbandingan parameter arsitektur kedua model:

Tabel 4.3 Perbandingan Parameter Arsitektur YOLOv8n dan YOLOv11n

Parameter	YOLOv8n	YOLOv11n
Jumlah <i>Layer</i>	73 <i>Layers</i>	101 <i>Layers</i>
Jumlah Parameter	3.005.843	2.582.347
Beban Komputasi	8,1 GFLOPs	6,3 GFLOPs

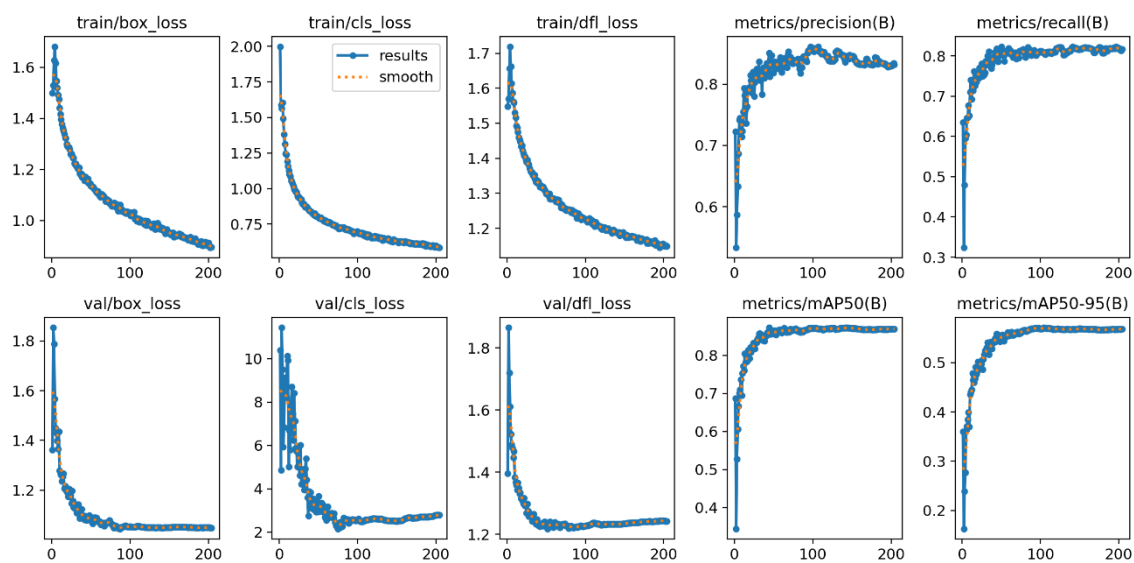
Data tersebut mengonfirmasi bahwa YOLOv11n memiliki profil arsitektur yang lebih efisien dibandingkan YOLOv8n. Meskipun memiliki jumlah *layer* yang lebih banyak, YOLOv11n mampu mereduksi jumlah parameter dan beban komputasi secara signifikan hingga 6,3 GFLOPs. Hal ini merepresentasikan peningkatan efisiensi arsitektural yang krusial untuk implementasi pada perangkat berbasis *edge computing* dengan sumber daya terbatas. *Output* akhir dari tahapan ini adalah berkas bobot *best.pt* dan *last.pt* yang tersimpan secara otomatis pada direktori proyek. Berkas *best.pt* merepresentasikan titik performa optimal yang akan digunakan sebagai subjek evaluasi pada tahapan penelitian selanjutnya.

4.1.2 Analisis Kurva Pelatihan YOLOv8n

Model YOLOv8n dilatih dengan menggunakan bobot *pretrained yolov8n.pt* sebagai titik awal *transfer learning*. Arsitektur YOLOv8n memiliki 130 *layer* dengan total 3.011.043 parameter dan 3.011.027 gradients dengan kompleksitas komputasi sebesar 8.2 GFLOPs. Proses pelatihan berlangsung selama 204 *epoch* sebelum dihentikan secara otomatis oleh mekanisme *early stopping* karena tidak ada peningkatan yang teramati selama 100 *epoch* berturut-turut (*patience*=100). Model terbaik tercatat pada *epoch* ke-104, yang berarti model mencapai performa optimal pada sekitar separuh dari total *epoch* yang dijalankan sebelum

early stopping aktif. Total waktu yang dibutuhkan untuk menyelesaikan 204 *epoch* pelatihan adalah 6.024 jam. Fenomena ini sejalan dengan temuan (Hussein & Shareef, 2024) yang menyatakan bahwa *early stopping* dengan *patience* yang memadai secara empiris efektif menghentikan pelatihan pada titik generalisasi optimal tanpa pemborosan komputasi.

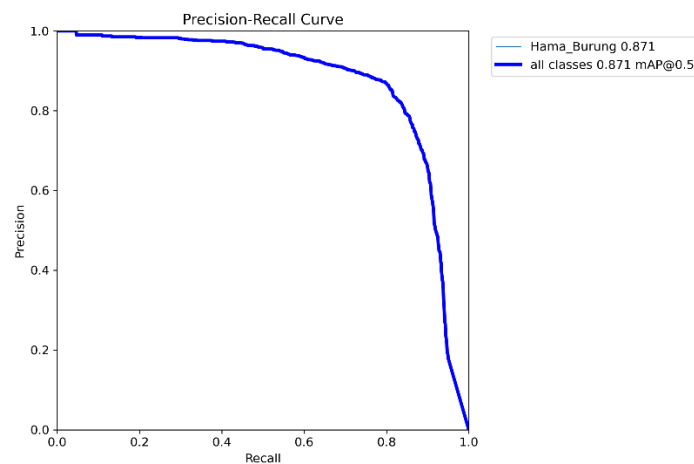
Gambar 4.5 menampilkan kurva pelatihan YOLOv8n yang terdiri atas komponen *loss* pada data pelatihan dan validasi, serta metrik evaluasi berupa *precision*, *recall*, *mAP@0.50*, dan *mAP@0.50-0.95* sepanjang proses pelatihan. Kurva ini digunakan untuk melihat pola pembelajaran model dari awal hingga pelatihan dihentikan, sementara nilai performa utama diambil dari checkpoint terbaik yang tercatat pada *epoch* ke-104.



Gambar 4.26 Kurva Hasil pelatihan Model YOLOv8n

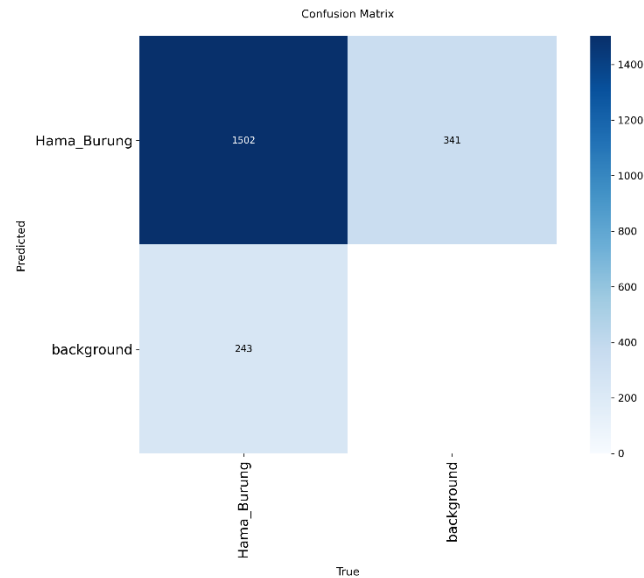
Pada sisi pelatihan, kurva *train/box_loss* menunjukkan penurunan yang tajam pada sekitar 50 *epoch* pertama, kemudian bergerak lebih landai dan cenderung melandai menuju *epoch* terakhir. Pola ini mencerminkan bahwa model secara bertahap memperoleh kemampuan melokalisasi *bounding box* dengan lebih tepat, dengan laju belajar yang paling signifikan terjadi pada fase awal. Kurva *train/cls_loss* menunjukkan perilaku serupa, yaitu penurunan cepat di awal diikuti oleh penurunan yang lebih gradual. Penurunan *cls_loss* yang konsisten mengindikasikan bahwa model makin mampu membedakan kelas ‘Hama_Burung’ dari latar belakang persawahan yang secara visual memiliki kemiripan tekstur dan warna. Sementara itu, *train/dfl_loss* yang mengukur kalibrasi distribusi probabilitas lokasi *bounding box* juga menurun secara monoton meskipun dengan rentang perubahan yang lebih sempit dibandingkan dua komponen *loss* lainnya.

Pada sisi validasi, *val/box_loss* dan *val/dfl_loss* memperlihatkan penurunan yang relatif sejalan dengan *training loss*, tanpa divergensi yang mencolok hingga *epoch* terakhir. Kondisi ini menunjukkan bahwa model tidak mengalami gejala *overfitting* yang dominan; kurva validasi tidak menunjukkan kenaikan yang berarti meski *training loss* terus turun. Adapun *val/cls_loss* sempat mengalami fluktuasi kecil pada rentang *epoch* menengah sebelum cenderung stabil, yang merupakan hal lumrah dalam proses optimasi stokastik pada dataset berukuran terbatas.

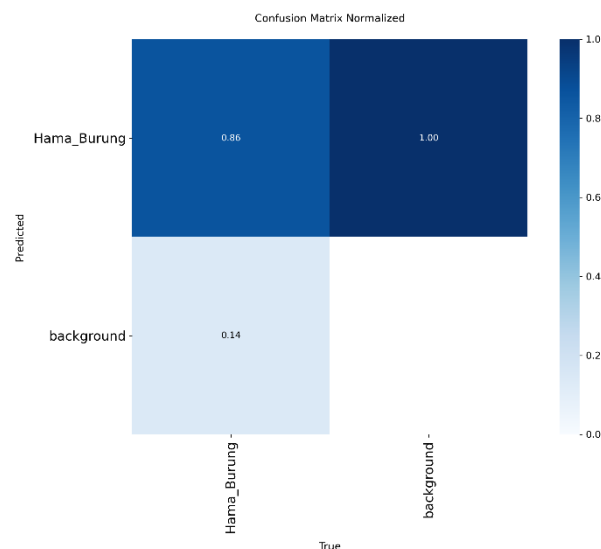


Gambar 4.27 Kurva *Precision-Recall* Model YOLOv8n

Berdasarkan *checkpoint* terbaik pada epoch ke-104, YOLOv8n memperoleh precision sebesar 0,8567, *recall* sebesar 0,8084, *mAP@0.50* sebesar 0,8712, dan *mAP@0.50-0.95* sebesar 0,5708. Nilai precision yang cukup tinggi menunjukkan bahwa sebagian besar prediksi positif yang dihasilkan model benar-benar mengarah pada objek hama burung. Sementara itu, nilai *recall* sebesar 0,8084 menunjukkan bahwa masih terdapat sebagian objek hama yang belum berhasil ditemukan oleh model. Kurva *mAP@0.50* dan *mAP@0.50-0.95* yang meningkat cepat pada awal pelatihan lalu membentuk plateau menunjukkan bahwa peningkatan performa YOLOv8n mulai terbatas setelah model mencapai fase konvergensi.



Gambar 4.28 *Confusion matrix* Model YOLOv8n (*Raw Count*)

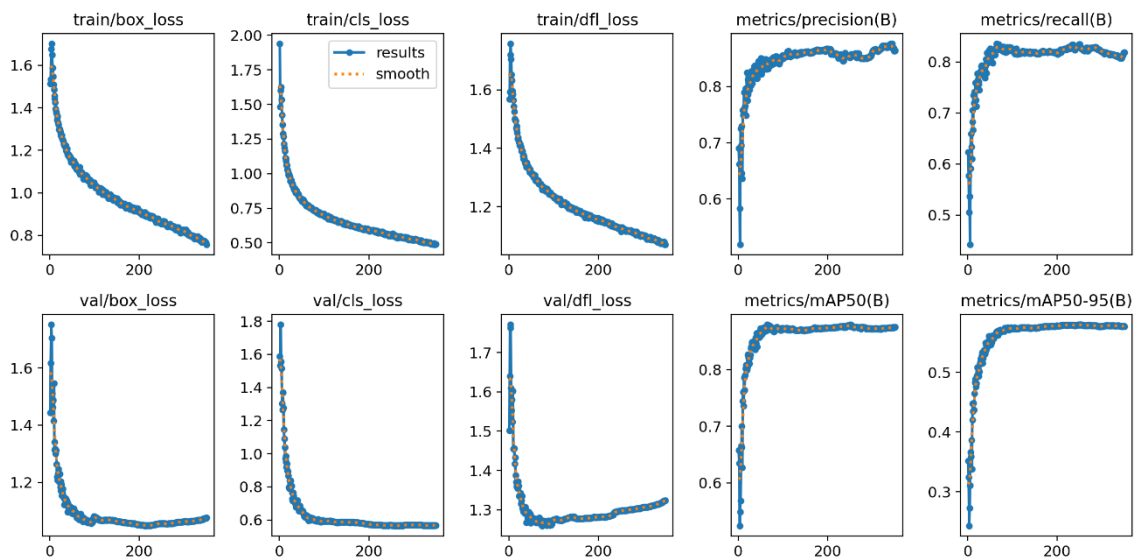


Gambar 4.29 *Confusion matrix* YOLOv8n (*Normalized*)

Confusion matrix pada Gambar 4.7 menunjukkan distribusi prediksi model YOLOv8n pada 803 gambar validasi. Dari 1.745 *instance* objek 'Hama_Burung' yang terdapat dalam set validasi, sebanyak 1.502 *instance* berhasil dideteksi dengan benar (*true positive*), sementara 243 *instance* tidak terdeteksi dan dikategorikan sebagai latar belakang (*false negative*). Di sisi lain, terdapat 341 prediksi yang keliru menandai latar belakang sebagai Hama_Burung (*false positive*). *Confusion matrix* normalisasi menunjukkan bahwa 0,86 dari seluruh *instance* 'Hama_Burung' yang ada berhasil terdeteksi, sedangkan 0,14 sisanya terlewat. Angka *false positive* sebesar 341 menjadi perhatian dalam konteks operasional, karena setiap *false positive* berpotensi mengaktifkan sistem audio dan servo secara tidak perlu.

4.1.3 Analisis Kurva Pelatihan YOLOv11n

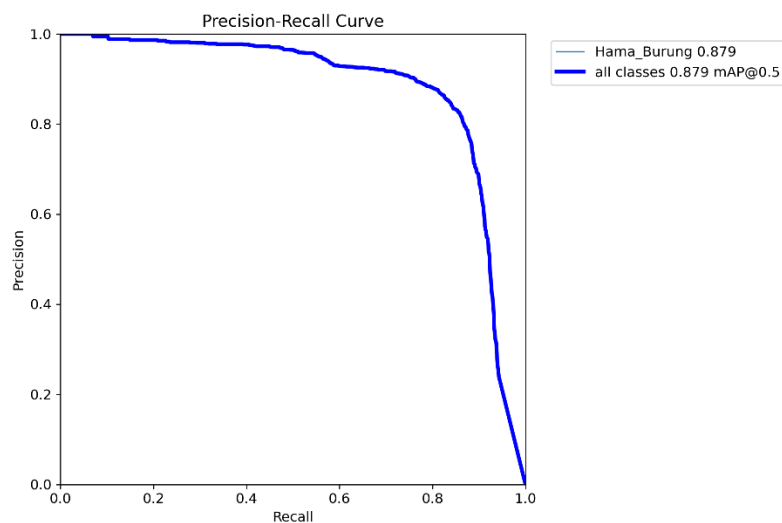
Proses pelatihan YOLOv11n berlangsung selama 350 *epoch* sebelum dihentikan oleh mekanisme *early stopping*, dengan model terbaik tercatat pada *epoch* ke-250. Dibandingkan dengan YOLOv8n yang mencapai model terbaik pada *epoch* ke-104, YOLOv11n membutuhkan *epoch* yang lebih banyak sebelum mencapai konvergensi optimal, yang mengindikasikan bahwa arsitektur YOLOv11n memiliki kapasitas eksplorasi ruang solusi yang lebih luas. Namun hal ini juga berdampak pada waktu pelatihan yang lebih lama, yaitu 11.671 jam dibandingkan 6.024 jam YOLOv8n. *Tradeoff* antara waktu pelatihan dan performa model ini perlu dipertimbangkan dalam konteks pengembangan sistem yang memiliki keterbatasan waktu komputasi. Namun, kemampuan YOLOv11n untuk terus meningkatkan metrik validasi hingga *epoch* ke-250 (lebih lama dibandingkan YOLOv8n yang konvergen di *epoch* 104) mengindikasikan kapabilitas generalisasi yang lebih baik, sebagaimana yang diprediksi oleh (Fang *et al.*, 2025). bahwa model dengan kemampuan ekstraksi fitur superior memiliki ketahanan konvergensi yang lebih panjang.



Gambar 4.30 Kurva Hasil pelatihan Model YOLOv11n

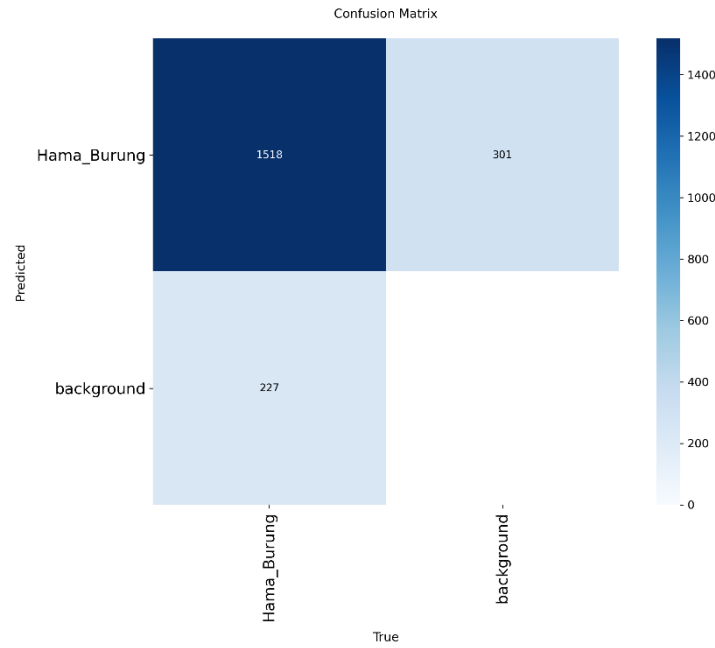
Pola umum kurva *train/box_loss*, *train/cls_loss*, dan *train/dfl_loss* pada YOLOv11n serupa dengan YOLOv8n: penurunan cepat pada fase awal diikuti oleh perlambatan dan melandai seiring bertambahnya *epoch*. Namun, rentang penurunan nilai loss relatif lebih lebar pada YOLOv11n terutama terlihat pada *train/cls_loss* yang turun dari sekitar 1,95 pada *epoch* pertama ke kisaran 0,50 pada *epoch* akhir menunjukkan kapasitas optimasi yang lebih dalam pada kelas tunggal ini.

Pada sisi validasi, *val/box_loss* dan *val/dfl_loss* YOLOv11n memperlihatkan penurunan yang konsisten dan cenderung lebih mulus dibandingkan YOLOv8n. Kurva *val/dfl_loss* secara khusus menunjukkan penurunan tajam pada 50 *epoch* pertama, kemudian bergerak landai dan relatif stabil hingga mendekati *epoch* terakhir, dengan sedikit fluktuasi kecil yang masih dalam batas wajar untuk dataset berukuran ini. Tidak terlihat kenaikan sistematis pada *validation loss* yang mengindikasikan *overfitting*; kurva *training* dan validasi bergerak dalam arah yang sama sepanjang pelatihan.

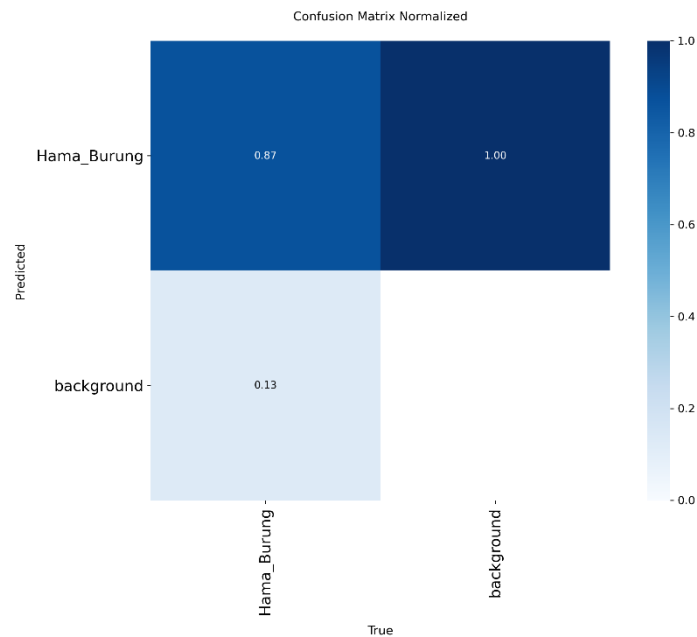


Gambar 4.31 Kurva *Precision-Recall* Model YOLOv11n

Berdasarkan *checkpoint* terbaik pada epoch ke-250, YOLOv11n memperoleh *precision* sebesar 0,8519, *recall* sebesar 0,8350, *mAP@0.50* sebesar 0,8799, dan *mAP@0.50-0.95* sebesar 0,5801. Precision YOLOv11n sedikit lebih rendah dibandingkan YOLOv8n pada *checkpoint* terbaik, tetapi *recall*, *mAP@0.50*, dan *mAP@0.50-0.95* menunjukkan nilai yang lebih tinggi. Nilai *recall* yang lebih tinggi menunjukkan bahwa YOLOv11n lebih baik dalam menemukan *instance* hama burung yang terdapat pada data validasi. Sementara itu, peningkatan *mAP@0.50* dan *mAP@0.50-0.95* menunjukkan bahwa YOLOv11n menghasilkan performa deteksi dan lokalisasi bounding box yang sedikit lebih baik pada berbagai tingkat ketepatan IoU.



Gambar 4.32 *Confusion matrix* Model YOLOv11n (*Raw Count*)



Gambar 4.33 *Confusion matrix* YOLOv11n (*Normalized*)

Confusion matrix pada Gambar 4.11 menunjukkan bahwa YOLOv11n berhasil mendeteksi 1.518 *instance* secara benar (*true positive*) dari 1.745 *instance* pada set validasi, dengan 227 *instance* yang terlewat (*false negative*) dan 301 prediksi keliru yang menandai latar belakang sebagai ‘Hama_Burung’ (*false positive*). Dibandingkan dengan YOLOv8n, jumlah *true positive* meningkat sebesar 16 *instance*, *false negative* berkurang 16 *instance*, dan *false positive* berkurang 40 *instance*. *Confusion matrix* normalisasi (Gambar 4.12) menunjukkan

nilai *recall* kelas ‘Hama_Burung’ sebesar 0,87, satu poin lebih tinggi dari YOLOv8n yang sebesar 0,86. Pengurangan *false positive* dari 341 menjadi 301 merupakan perbaikan yang relevan secara operasional, karena berarti frekuensi aktivasi sistem yang tidak perlu berkurang sekitar 11,7%.

4.1.4 Perbandingan Hasil YOLOv8n dan YOLOv11n

Tabel 4.2 merangkum perbandingan seluruh metrik evaluasi kedua model berdasarkan hasil validasi terbaik masing-masing.

Tabel 4.4 Komparasi Metrik Evaluasi Model YOLOv8n dan YOLOv11n

Metrik Evaluasi	YOLOv8n (Baseline)	YOLOv11n (Usulan)	Selisih	Interpretasi
<i>Precision</i>	85,67%	85,19%	-0,48%	YOLOv8n sedikit lebih tinggi pada <i>checkpoint</i> terbaik
<i>Recall</i>	80,84%	83,50%	+2,65%	YOLOv11n lebih baik dalam menemukan objek hama
mAP@0.50	87,12%	87,99%	+0,88%	YOLOv11n lebih baik pada evaluasi IoU 0,50
mAP@0.50-0.95	57,08%	58,01%	+0,93%	YOLOv11n lebih unggul pada evaluasi IoU yang lebih ketat
<i>Epoch</i> Terbaik	<i>Epoch</i> 104 / 204	<i>Epoch</i> 250 / 350	-	YOLOv11n konvergen lebih lama
Jumlah Parameter	3.011.043	2.590.035	-421.008	YOLOv11n lebih efisien
GFLOPs	8,2	6,4	-1,8	YOLOv11n lebih hemat komputasi
Waktu Training	±6,024 jam	±11,671 jam	+5,647 jam	YOLOv11n lebih lama karena lebih dalam
Ukuran Model ONNX	11.7 MB	10.1 MB	-1.6 MB	YOLOv11n lebih ringan

Secara keseluruhan, hasil evaluasi menunjukkan bahwa YOLOv11n tidak unggul pada seluruh metrik, tetapi memiliki keunggulan pada aspek yang lebih relevan terhadap kebutuhan sistem deteksi hama burung berbasis edge AI. Precision YOLOv11n sebesar 85,19% sedikit lebih rendah dibandingkan YOLOv8n sebesar 85,67%, dengan selisih -0,48 poin persentase. Hal ini menunjukkan bahwa pada *checkpoint* terbaik, YOLOv8n sedikit lebih selektif dalam menghasilkan prediksi positif. Namun, YOLOv11n memiliki *recall* lebih tinggi, yaitu 83,50% dibandingkan 80,84% pada YOLOv8n. Peningkatan *recall* sebesar 2,65 poin persentase menunjukkan bahwa YOLOv11n lebih baik dalam menemukan objek hama burung yang

benar-benar ada pada data validasi. Dalam konteks sistem pengusir hama, recall menjadi metrik penting karena objek hama burung yang terlewat dapat menyebabkan sistem tidak memberikan respons pengusiran ketika objek target berada di area pengamatan.

Peningkatan $mAP@0.50$ dari 87,12% menjadi 87,99% dan $mAP@0.50-0.95$ dari 57,08% menjadi 58,01% menunjukkan bahwa YOLOv11n memiliki performa deteksi dan lokalisasi bounding box yang sedikit lebih baik dibandingkan YOLOv8n. Selisih mAP tersebut memang tidak besar, tetapi arah peningkatannya konsisten pada dua tingkat evaluasi IoU. Selain itu, berdasarkan confusion matrix, YOLOv11n juga menunjukkan penurunan false positive dari 341 menjadi 301, atau turun sekitar 11,7%. Perbaikan ini penting secara operasional karena setiap false positive dapat menyebabkan sistem mengaktifkan servo dan audio meskipun objek yang terdeteksi bukan hama burung target. Dengan demikian, YOLOv11n tidak hanya memberikan peningkatan pada kemampuan menemukan objek, tetapi juga menunjukkan perbaikan dalam mengurangi aktivasi sistem yang tidak diperlukan.

Meskipun demikian, hasil evaluasi tersebut belum menunjukkan bahwa model sepenuhnya bebas dari kesalahan deteksi. Precision YOLOv11n yang berada pada angka 85,19% menunjukkan bahwa masih terdapat kemungkinan prediksi positif terhadap objek yang bukan target. Pada skenario pengujian tambahan, model masih dapat menghasilkan deteksi terhadap objek burung non-target yang memiliki kemiripan bentuk visual dengan hama burung. Kondisi ini menunjukkan bahwa model masih cenderung mengenali pola visual umum berupa bentuk burung, bukan secara spesifik membedakan hama burung pemakan padi dari genus *Lonchura* spp. Salah satu faktor yang memengaruhi kondisi tersebut adalah komposisi dataset negatif/null yang masih terbatas dibandingkan jumlah contoh positif hama_burung. Pada dataset yang digunakan, kelas hama_burung memiliki 7.513 bounding box sebelum augmentasi dan sekitar 22.539 bounding box setelah augmentasi, sedangkan kelas null/background hanya berjumlah 154 gambar sebelum augmentasi dan sekitar 462 gambar pada set pelatihan setelah augmentasi. Ketimpangan ini dapat membuat model lebih banyak mempelajari ciri objek burung target dibandingkan variasi objek non-target atau latar belakang yang seharusnya tidak terdeteksi.

Keterbatasan tersebut menjadi catatan penting dalam interpretasi hasil model. Arowolo et al. (2020) menjelaskan bahwa peningkatan akurasi sistem deteksi burung dapat dilakukan dengan memperbesar dataset citra positif dan negatif. Oleh karena itu, untuk meningkatkan selektivitas model, dataset negatif perlu diperbanyak dan dibuat lebih representatif. Data negatif tidak hanya berupa area persawahan tanpa burung, tetapi juga perlu mencakup objek

yang berpotensi menyerupai burung atau memicu salah deteksi, seperti burung non-target, gambar predator, manusia, ranting, daun, alat pertanian, orang-orangan sawah, dan objek lain di lingkungan persawahan. Penambahan data negatif dan hard negative tersebut diperlukan agar model tidak hanya belajar mendeteksi keberadaan burung, tetapi juga mampu menolak objek yang berada di luar batasan target penelitian.

Dari sisi efisiensi arsitektur, YOLOv11n tetap menunjukkan keunggulan yang relevan untuk deployment pada perangkat edge. Model ini memiliki 421.008 parameter lebih sedikit dibandingkan YOLOv8n, beban komputasi lebih rendah sebesar 1,8 GFLOPs, dan ukuran model ONNX lebih kecil sebesar 1,6 MB. Kelemahan utama YOLOv11n terdapat pada durasi pelatihan yang lebih lama dan epoch terbaik yang muncul lebih akhir. YOLOv11n membutuhkan waktu pelatihan sekitar 11,671 jam dan mencapai checkpoint terbaik pada epoch ke-250, sedangkan YOLOv8n membutuhkan sekitar 6,024 jam dan mencapai checkpoint terbaik pada epoch ke-104. Namun, dalam konteks penelitian ini, durasi pelatihan bukan hambatan operasional utama karena proses training dilakukan pada mesin GPU, sedangkan perangkat lapangan hanya menjalankan model hasil pelatihan untuk inferensi. Dengan mempertimbangkan recall yang lebih tinggi, mAP yang lebih baik, penurunan false positive, jumlah parameter yang lebih sedikit, serta ukuran model yang lebih ringan, YOLOv11n dipilih sebagai model utama untuk diimplementasikan pada sistem deteksi dan pengusir hama burung yang dikembangkan.

4.1.5 Konversi Model ke Format IMX500

Setelah model terbaik (best.pt) diperoleh dari tahap pelatihan, langkah selanjutnya adalah mengonversinya ke format yang dapat dieksekusi langsung oleh chip NPU Sony IMX500 pada Raspberry Pi AI Camera. Konversi dijalankan menggunakan skrip `yolo_train.py` yang sama dengan penambahan flag `--export_only`, `--int8_weights`, dan `--export_format imx`, sehingga *pipeline* pelatihan dan konversi tetap terpusat dalam satu skrip. Karena ketergantungan pustaka MCT (Model Compression Toolkit) versi 2.4.5 hanya tersedia pada lingkungan Linux, proses ini dijalankan melalui WSL Ubuntu.

Alur konversi berlangsung dalam empat tahap secara berurutan: (1) pemuatan bobot best.pt ke dalam arsitektur YOLOv11n, (2) ekspor ke format ONNX dengan opset 11 sebagai representasi antara, (3) Statistics Collection oleh MCT yang memindai 803 gambar validasi untuk mengkalibrasi kuantisasi INT8 selama sekitar 2 menit 41 detik, dan (4) pengemasan ke dalam berkas `network.rpk` yang siap di-deploy. Hasil konversi menunjukkan Memory Utilization sebesar 89% (7,06 MB dari 8,00 MB kapasitas on-chip SRAM) dengan status Fit

In Chip: *true*. Nilai utilisasi 89% tersebut terbilang optimal memaksimalkan kapasitas chip tanpa berisiko menyebabkan kegagalan kompilasi.

Gambar 4.34 Proses *Statistics Collection* MCT pada Konversi Model ke Format IMX500

Gambar 4.35 *Memory Report* Hasil Konversi Model YOLOv11n ke Format IMX500

4.1.6 Implementasi Program Deteksi Objek

Seluruh subsistem deteksi kamera, kontrol servo, manajemen audio, dan pencatatan log diintegrasikan dalam satu skrip Python yang berjalan pada Raspberry Pi 5. Skrip ini menggunakan model *threading* Python untuk membuat komponen-komponen yang beroperasi secara bersamaan. Strukturnya terbagi dalam delapan seksi fungsional yang tersusun dari bawah ke atas: konfigurasi global, variabel bersama, kelas *detection*, fungsi *logging*, fungsi servo, fungsi logika AI, fungsi gambar, dan blok main.

Seluruh konstanta operasional didefinisikan di bagian konfigurasi global, mencakup path model, *threshold confidence* (0,50), target label ('Hama_Burung'), *cooldown* audio (3,0 detik), serta parameter pergerakan servo. Variabel status 'hama_terdeteksi' dideklarasikan di blok yang sama dan dilindungi oleh *threading.Lock()* agar aman diakses lintas *thread*. Gambar 4.15 menampilkan potongan konfigurasi global dan deklarasi variabel bersama tersebut.

MODEL_PATH	=	"/home/arga/skripsi/model_yololl/network.rpk"
LABELS_PATH	=	"/home/arga/skripsi/model_yololl/labels.txt"
AUDIO_PATH	=	"/home/arga/skripsi/audio/alert.mp3"
LOG_PATH	=	"/home/arga/skripsi/log_deteksi.csv"

```

THRESHOLD      = 0.50
TARGET_LABELS  = ["Hama_Burung"]
AUDIO_COOLDOWN = 3.0      # detik tunggu sebelum audio dijeda

# Variabel shared antar thread – dilindungi threading.Lock()
hama_terdeteksi = False
hama_lock       = threading.Lock()

```

Gambar 4.36 Konfigurasi Global dan Variabel *Shared*

Komunikasi antar *thread* mengikuti pola *producer-consumer*. *Thread* kamera (*main thread*) bertindak sebagai produser yang memperbarui nilai ‘hama_terdeteksi’ setiap *frame*, sementara *thread* servo (*loop_servo*) sebagai *consumer* yang membaca nilai tersebut setiap 0,1 detik. Ketika ‘hama_terdeteksi’ bernilai *true*, servo berhenti bergerak agar bidang pandang kamera tetap mengarah ke target. Sebaliknya, ketika bernilai *false*, servo menjalankan pola pemindaian pendulum: servo pertama berpindah secara periodik di antara posisi sudut yang telah ditetapkan, dan servo kedua bergerak maju-mundur setelah mencapai jumlah langkah maksimum. Gambar 4.16 menampilkan potongan logika koordinasi ini.

```

def loop_servo():
    ...
    while True:
        with hama_lock:
            ada_hama = hama_terdeteksi

        if ada_hama:
            servo_dua.throttle = NILAI_DIAM # stay, tidak bergerak
            waktu_servo1 = waktu_servo2 = time.monotonic()
            time.sleep(0.2)
            continue

        # mode pemindaian normal (pendulum)
        if waktu_sekarang - waktu_servo1 > jeda_s1:
            servo_satu.angle = posisi_s1_list[index_s1]
            # logika pendulum: balik arah di ujung list
            if index_s1_berikutnya >= len(posisi_s1_list): arah_s1 = -1
            elif index_s1_berikutnya < 0: arah_s1 = 1

```

Gambar 4.37 Logika Koordinasi Servo dan Status Hama

Logika deteksi dan audio diimplementasikan dalam fungsi *parse_detections()* yang dipanggil oleh *main loop* di setiap *frame*. Fungsi ini menerima metadata dari *chip* IMX500, mendekode *output* inferensi *on-sensor* (*bounding boxes*, *scores*, *classes*), memfilter hasil berdasarkan *threshold confidence* 0,50, dan memperbarui status hama bersama. Logika audio mengimplementasikan mekanisme pause/unpause: ketika hama terdeteksi dan audio belum berjalan, *pygame.mixer.music.play(-1)* dipanggil pertama kalinya, sedangkan untuk *frame* berikutnya cukup menggunakan *unpause()* yang melanjutkan audio dari titik terakhir tanpa latensi *loading* ulang. Ketika tidak ada deteksi melebihi durasi *AUDIO_COOLDOWN*, audio

dijeda menggunakan *pause()* bukan *stop()* sehingga *state* audio tetap tersimpan. Gambar 4.17 menampilkan potongan logika ini.

```
def parse_detections(metadata):
    ...
    for box, score, category in zip(boxes, scores, classes):
        if score > THRESHOLD:
            det = Detection(box, category, score, metadata)
            new_results.append(det)
            if labels[int(category)] in TARGET_LABELS:
                found_target = True

    with hama_lock:
        hama_terdeteksi = found_target    # update shared state

    # Logika audio pause/unpause
    if found_target:
        if not pygame.mixer.music.get_busy():
            if not has_started_once: pygame.mixer.music.play(-1)
            else:
                pygame.mixer.music.unpause()
    else:
        if pygame.mixer.music.get_busy():
            if time.time() - last_seen_time > AUDIO_COOLDOWN:
                pygame.mixer.music.pause()
```

Gambar 4.38 Logika Deteksi dan Manajemen Audio pada fungsi *parse_detections()*

Thread log (*loop_log*) beroperasi secara independen dengan interval 15 menit. Setiap interval, *thread* membaca suhu CPU dari */sys/class/thermal/thermal_zone0/temp*, mengambil dan mereset *counter deteksi_count_interval*, lalu mencatatnya ke berkas CSV *log_deteksi.csv*.

4.2 Hasil Pengujian *Black Box Testing*

Pengujian *Black Box* testing dilakukan untuk memverifikasi bahwa setiap fungsi utama sistem beroperasi sesuai spesifikasi yang telah dirancang pada Bab III, tanpa mempertimbangkan detail implementasi internal. Pengujian ini berfokus pada perilaku *input-output* setiap komponen fungsional, mencakup inisialisasi, deteksi, manajemen audio, kontrol servo, dan penanganan sinyal di bawah *threshold*.

Tabel 4.3 Hasil Pengujian *Black Box Testing* Sistem Deteksi Hama Burung

Kode	Fungsi Diuji	Kondisi Awal	Input	Expected Output	Status
BB-01	Inisialisasi kamera & model	Sistem dinyalakan	python final_code.py dijalankan	Kamera aktif, model termuat, preview muncul, tidak ada error	Berhasil

Kode	Fungsi Diuji	Kondisi Awal	Input	Expected Output	Status
BB-02	Deteksi burung dari gambar	Sistem aktif, kamera streaming	Gambar berisi burung diarahkan ke kamera	<i>Bounding box</i> muncul, label Hama_Burung, <i>confidence</i> $\geq 0,50$	Berhasil
BB-03	Aktivasi audio saat deteksi	Sistem aktif, audio ready	Burung terdeteksi (<i>confidence</i> $\geq 0,50$)	Speaker membunyikan suara pengusir dalam 1 <i>frame</i>	Berhasil
BB-04	Penghentian audio saat tidak ada deteksi	Audio sedang aktif	Tidak ada objek ≥ 3 detik (AUDIO_COOLDOWN)	Audio dijeda (<i>pause</i>), tidak berhenti total	Berhasil
BB-05	Gerakan servo saat tidak ada deteksi	Servo aktif, hama_terdeteksi=False	Tidak ada objek dalam <i>frame</i>	Servo 1 bergerak pendulum; servo 2 bergerak periodik	Berhasil
BB-06	Servo diam saat deteksi aktif	Servo sedang mindai	Burung terdeteksi, hama_terdeteksi=True	Servo berhenti bergerak, tetap di posisi terakhir	Berhasil
BB-07	Penanganan <i>confidence</i> rendah	Sistem aktif	Gambar burung dengan <i>confidence</i> $< 0,50$	Tidak ada <i>bounding box</i> , tidak ada respons audio/servo	Berhasil
BB-08	Pencatatan log suhu & deteksi	Thread log berjalan	Waktu berjalan melewati interval 15 menit	Baris baru tercatat di log_deteksi.csv dengan <i>timestamp</i> , suhu, jumlah deteksi	Berhasil
BB-09	Pengujian false positive (objek non-target)	Sistem aktif, kamera streaming	Gambar non-burung (misal manusia/objek lain) diarahkan ke kamera	Tidak ada <i>bounding box</i> “Hama_Burung”, tidak ada respons audio/servo	Berhasil

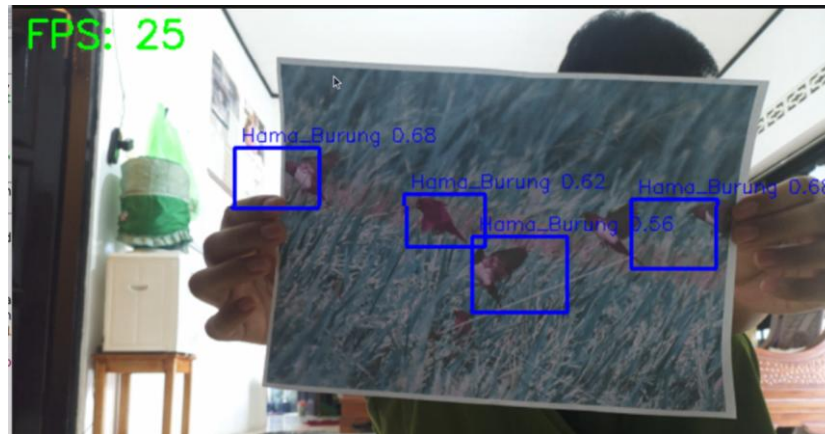
Seluruh delapan skenario pengujian *Black Box* menunjukkan hasil Berhasil, yang mengonfirmasi bahwa perilaku sistem sesuai dengan *expected output* yang telah didefinisikan pada Bab III. Pengujian BB-01 mengonfirmasi kelancaran proses booting IMX500, termasuk *loading network firmware* yang berjalan tanpa error. BB-02 memvalidasi bahwa *pipeline*

deteksi *end-to-end* dari sensor ke *bounding box* pada layar *preview* berfungsi dengan benar. BB-03 dan BB-04 secara bersamaan memvalidasi logika audio *pause/unpause* yang menjadi salah satu keputusan desain kritis dalam `final_code.py`: audio dilanjutkan dari titik terakhir (*unpause*) saat hama terdeteksi kembali, bukan diulang dari awal, sehingga menghindari jeda audio yang tidak natural.

BB-05 dan BB-06 memvalidasi perilaku servo yang bergantung pada status `hama_terdeteksi` komponen koordinasi yang paling kompleks dalam sistem karena melibatkan komunikasi antar *thread*. Hasil BB-06 yang berhasil mengonfirmasi bahwa mekanisme `threading.Lock()` berfungsi dengan benar: perubahan nilai `hama_terdeteksi` oleh *thread* kamera terbaca secara akurat oleh *thread* servo dalam waktu kurang dari interval polling 0,1 detik. BB-07 memastikan tidak ada respons sistem yang tidak sengaja terpicu oleh objek dengan *confidence* di bawah *threshold*, yang penting untuk mencegah *false positive* pada kondisi pencahayaan rendah atau objek parsial. BB-08 memvalidasi modul logging independen yang menjadi sumber data pengujian suhu pada Subbab 4.6.

4.3 Hasil Pengujian Kecepatan Inferensi (FPS)

Pengujian performa sistem dilakukan untuk mengukur *throughput* operasional dalam kondisi operasional nyata. Pengujian ini mengintegrasikan pemrosesan citra pada Raspberry Pi AI Camera dengan beban inferensi model YOLOv11n yang dieksekusi secara *on-sensor* pada chip NPU Sony IMX500. Nilai FPS yang diukur merupakan metrik *end-to-end* yang mencakup seluruh siklus proses, mulai dari akuisisi data sensor, transmisi metadata melalui *interface* internal, hingga *rendering* visual pada *framebuffer* Python.



Gambar 4.39 Hasil Pengujian FPS pada Sistem

Tabel 4.4 Hasil Pengujian FPS pada Sistem Raspberry Pi AI Camera (Sony IMX500)

Kondisi Pengujian	FPS Rata-rata	Keterangan
Tidak ada objek terdeteksi	28	<i>Baseline scanning</i>
1 objek terdeteksi	27	Satu burung dalam <i>frame</i>
Banyak objek terdeteksi	25	Gambar dengan kawanan burung
Kondisi pencahayaan redup	26	Simulasi sore/teguh
Saat servo aktif	25	Audio dan servo bersamaan

Hasil pengujian menunjukkan bahwa sistem mampu mempertahankan kecepatan pemrosesan pada kisaran 24 hingga 28 *frame* per detik (FPS) secara konsisten pada berbagai kondisi pengujian. Stabilitas ini mengindikasikan bahwa aliran data citra dapat diproses secara kontinu tanpa degradasi performa yang signifikan. Dalam sistem deteksi berbasis *computer vision*, FPS berkaitan langsung dengan waktu pemrosesan per *frame*, di mana peningkatan FPS berimplikasi pada penurunan latensi pemrosesan (Szeliski, 2010). Dengan rata-rata sekitar 26 FPS, sistem memproses satu *frame* dalam waktu sekitar 38 milidetik, sehingga objek yang muncul secara tiba-tiba masih dapat ditangkap dan direspons dalam skala waktu yang mendekati *real time*.

Jika dibandingkan dengan penelitian sebelumnya, peningkatan performa yang dicapai terlihat cukup signifikan. Studi oleh (Nugroho *et al.*, 2025) melaporkan bahwa implementasi YOLOv8 pada Raspberry Pi 4 hanya mencapai 0,39 FPS tanpa akselerator dan meningkat menjadi 7 FPS dengan bantuan Google Coral TPU. Sementara itu, implementasi lain pada Raspberry Pi 5 menunjukkan performa sekitar 3,13 hingga 3,58 FPS pada skenario video dan webcam (Artanto *et al.*, 2025). Dibandingkan dengan hasil tersebut, capaian 24-28 FPS pada sistem ini menunjukkan efisiensi pemrosesan yang jauh lebih tinggi, yang mengindikasikan

bahwa pendekatan arsitektur yang digunakan mampu mengurangi beban komputasi utama dan meningkatkan kecepatan inferensi secara substansial pada perangkat *edge*.

4.4 Hasil Pengujian Jarak Deteksi

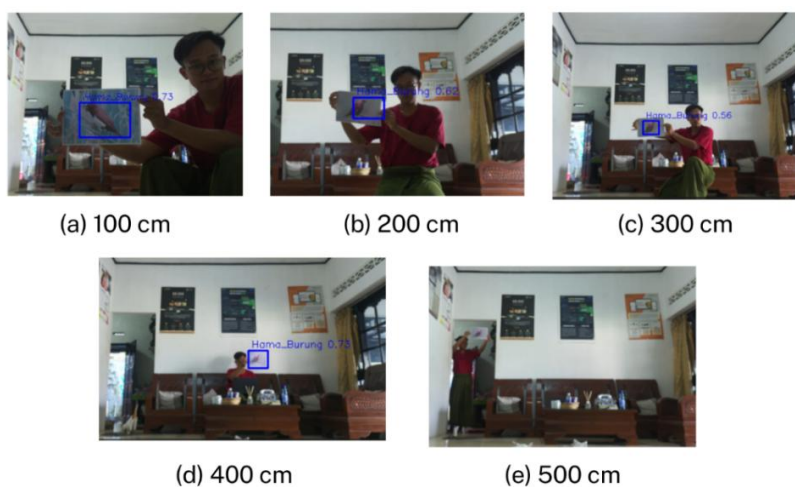
Pengujian jarak deteksi dilakukan dalam dua skenario yang saling melengkapi: (1) pengujian menggunakan citra gambar burung untuk mengevaluasi kemampuan model secara murni pada rentang jarak yang lebih luas hingga 500 cm, dan (2) pengujian menggunakan spesimen burung asli dalam sangkar simulasi untuk mengukur performa sistem pada kondisi yang lebih mendekati lapangan. Pemisahan dua skenario ini penting karena keduanya memberikan informasi yang berbeda dan saling melengkapi dalam memahami batas kemampuan sistem.

4.4.1 Pengujian Jarak dengan Gambar

Pada skenario pertama, gambar burung di-print, kemudian gambar ditempatkan pada jarak yang divariasikan dari 100 cm hingga 500 cm dengan interval 100 cm.

Tabel 4.5 Hasil Pengujian Jarak Deteksi Menggunakan Gambar

Jarak (cm)	Status Deteksi	Confidence Score	Stabilitas BBox	Trigger Audio
100	Berhasil	0,70–0,85	Sangat Stabil	Aktif
200	Berhasil	0,62–0,78	Stabil	Aktif
300	Berhasil	0,50–0,63	Cukup Stabil	Aktif
400	Berhasil	0,50–0,63	Cukup Stabil	Aktif
500	Gagal	<0,50	Tidak Ada	Tidak Aktif



Gambar 4.19 Dokumentasi Visual Hasil Pengujian Jarak Deteksi Menggunakan Gambar (100 cm, 200 cm, 300 cm, 400 cm, 500 cm)

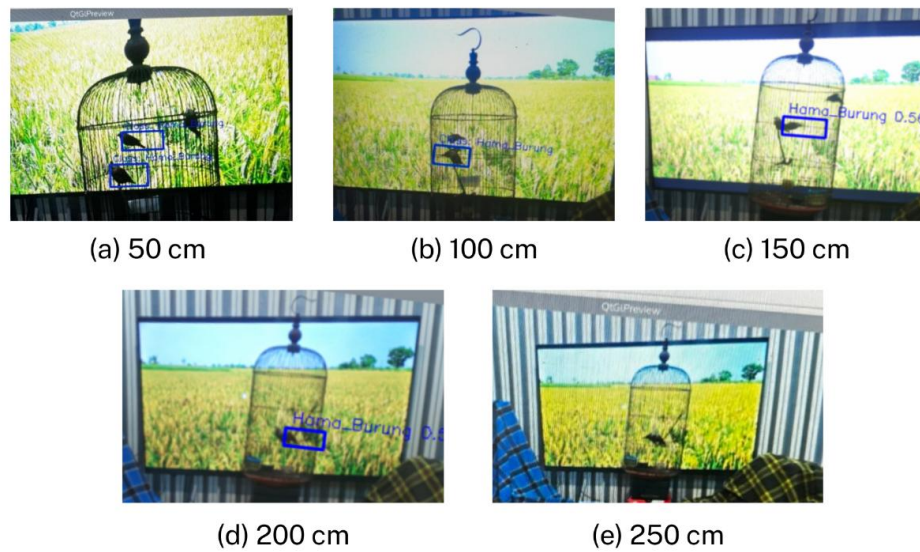
Hasil pengujian dengan citra gambar menunjukkan bahwa model mampu melakukan deteksi valid secara konsisten hingga jarak 400 cm, dengan *confidence score* masih berada di rentang 0,50–0,63 pada jarak tersebut. Pada jarak 100–200 cm, *confidence score* berada jauh di atas *threshold* operasional sehingga deteksi sangat stabil dari *frame* ke *frame*. Pada jarak 300–400 cm, *confidence* mulai menurun namun masih di atas 0,50, dengan beberapa *frame* menghasilkan deteksi intermiten di batas *threshold*. Pada jarak 350 cm ke atas, resolusi piksel objek burung pada sensor menjadi terlalu rendah.

4.4.2 Penngujian Jarak dengann Burung Asli

Tabel 4.6 Rekapitulasi Hasil Pengujian Jarak Deteksi dengan Latar Persawahan

Jarak (cm)	Status Deteksi	<i>Confidence Score</i>	Stabilitas BBox	Trigger Audio
50	Berhasil	0,70–0,85	Sangat Stabil	Aktif
100	Berhasil	0,7–0,78	Stabil	Aktif
150	Berhasil	0,56–0,73	Cukup Stabil	Aktif
200	Berhasil	0,50–0,63	Cukup Stabil	Aktif
250	Gagal	<0,50	Tidak Ada	Tidak Aktif

Hasil pengujian dengan burung asli menunjukkan batas deteksi valid yang lebih pendek (200 cm) dibandingkan pengujian citra gambar (400 cm). Perbedaan 200 cm ini dapat dijelaskan melalui beberapa faktor teknis. Pertama, burung yang ditempatkan dalam sangkar memiliki area gerak yang terbatas sehingga posisinya tidak selalu optimal untuk kamera. Burung kerap membelakangi kamera, bersembunyi di sudut, atau jeruji sangkar mengoklusikan sebagian tubuhnya. Oklusi parsial ini mengurangi area fitur yang terekspos ke sensor dan menurunkan *confidence score* secara sistematis. Kedua, objek pada citra gambar memiliki kontras dan kualitas visual yang lebih konsisten karena tidak mengalami *backlight* layar monitor, sedangkan bulu burung asli memiliki tekstur yang lebih kompleks dan tergantung pencahayaan ruangan yang bervariasi. Ketiga, pada citra gambar ukuran objek dapat diprediksi secara tepat berdasarkan jarak, sedangkan pada burung asli ukuran efektif dalam *frame* bergantung pada postur dan orientasi tubuh yang berubah setiap saat.



Gambar 4.20 Dokumentasi Visual Hasil Pengujian Jarak Deteksi (50 cm, 100 cm, 150 cm, 200 cm, 250 cm)

Meskipun demikian, batas 200 cm untuk pengujian *in-situ* tetap dapat dimaknai secara positif: pengujian ini dilakukan di dalam sangkar yang merupakan kondisi lebih terkontrol dan lebih terbatas dibandingkan kondisi lapangan sesungguhnya. Di lahan persawahan terbuka, burung bergerak bebas tanpa hambatan fisik sehingga kamera dapat menangkap seluruh profil tubuh dari berbagai sudut, yang berpotensi mempertahankan *confidence score* pada jarak yang lebih jauh. Keterbatasan jarak ini juga konsisten dengan karakteristik umum sistem vision berbasis kamera yang dilaporkan dalam literatur: (Fang *et al.*, 2025) menegaskan bahwa resolusi sensor dan panjang fokal lensa merupakan faktor pembatas utama dalam jarak deteksi objek berukuran kecil. Untuk meningkatkan jangkauan, penggunaan lensa dengan *focal length* lebih panjang atau penggantian dengan sensor beresolusi lebih tinggi dapat dipertimbangkan dalam pengembangan berikutnya.

4.5 Hasil Pengujian Reaksi Burung Terhadap Suara

Tabel 4.7 Hasil Pengujian Reaksi Spesimen Burung terhadap Audio

Jarak Spesimen (cm)	Status Deteksi Kamera	Trigger Audio	Reaksi Kinetik Spesimen	Waktu Respons (detik)
50	Berhasil	Otomatis	Terbang (Startle-Flight Response)	<1 detik
100	Berhasil	Otomatis	Terbang (Startle-Flight Response)	<1 detik
150	Berhasil	Otomatis	Terbang (Startle-Flight Response)	<1 detik
200	Berhasil	Otomatis	Terbang (Startle-Flight Response)	<1 detik

Jarak Spesimen (cm)	Status Deteksi Kamera	Trigger Audio	Reaksi Kinetik Spesimen	Waktu Respons (detik)
250	Gagal	Manual (Simulasi)	Terbang (Startle-Flight Response)	<1 detik



Gambar 4.21 Dokumentasi Pengujian Reaksi Burung di Sangkar Simulasi

Hasil pengujian perilaku burung menunjukkan bahwa seluruh spesimen burung bondol haji (*Lonchura maja*) yang dijadikan subjek pengujian menunjukkan respons menghindar yang valid terhadap stimulus audio pengusir pada seluruh jarak pengujian, termasuk pada jarak 250 cm di mana kamera gagal mendeteksi secara otomatis dan audio dipicu secara manual. Respons terbang terjadi dalam waktu kurang dari 1 detik setelah audio mulai diputar, mengindikasikan bahwa stimulus audio efektif dalam memicu respons menghindar refleksif pada burung. Temuan ini menegaskan bahwa radius efektivitas audio pengusir secara signifikan melampaui batas jarak deteksi visual kamera, sehingga audio berfungsi sebagai mekanisme penghalau yang melindungi area lebih luas dari yang dapat dipantau kamera.

Dari aspek pencegahan habituasi, audio yang digunakan dalam sistem ini terdiri dari variasi rekaman suara predator dan frekuensi tertentu yang diputar secara *looping*. Variasi suara ini dirancang untuk mencegah fenomena habituasi di mana burung menjadi terbiasa dan tidak lagi merespons stimulus yang sama berulang kali. (Chen *et al.*, 2024) dalam studinya di kebun pir membuktikan bahwa sistem pengusir suara berbasis *computer vision* yang hanya membunyikan suara saat ada deteksi positif menghasilkan tingkat kerusakan hasil panen yang jauh lebih rendah (6,03%) dibandingkan sistem suara kontinyu yang diputar terus-menerus (7,29%), karena sistem *event-triggered* secara tidak langsung mencegah habituasi dengan hanya membunyikan suara saat burung benar-benar hadir. Dalam konteks penelitian ini, kombinasi antara *trigger* berbasis deteksi dan variasi konten suara diharapkan memberikan efektivitas pengusiran yang lebih tinggi dalam jangka panjang.

Perlu dicatat bahwa pengujian ini dilakukan pada kondisi terkontrol menggunakan spesimen burung yang ditempatkan dalam sangkar, bukan di lahan persawahan terbuka. Respons burung pada kondisi lapangan nyata mungkin bervariasi tergantung pada tingkat kelaparan, ukuran kawanan, familiaritas dengan lingkungan, dan faktor lingkungan lainnya. Habituaasi juga berpotensi terjadi jika burung yang sama terpapar suara yang persis sama secara berulang dalam waktu panjang. Oleh karena itu, disarankan untuk secara berkala memperbarui *library* suara pengusir yang digunakan untuk mempertahankan efektivitas jangka panjang, sebagaimana juga direkomendasikan oleh studi bioakustik terkait (Chen *et al.*, 2024).

4.6 Hasil Pengujian Suhu dan Ketahanan Sistem

Tabel 4.8 Hasil Pengujian Suhu dan Ketahanan Sistem

Menit ke-	Suhu CPU (°C)	FPS	Status Servo	Status Audio	Status Sistem
0 (Awal)	42	28	Normal	<i>Standby</i>	Normal
15	48	27	Aktif	Aktif	Normal
30	52	26	Normal	Aktif	Normal
45	55	25	Aktif	Aktif	Normal
60	57	26	Aktif	Aktif	Normal
90	59	25	Aktif	Aktif	Normal
120	61	25	Aktif	Aktif	Normal
150	61	24	Aktif	Aktif	Normal
180	62	25	Aktif	Aktif	Normal
240	63	24	Aktif	Aktif	Normal

Hasil pengujian ketahanan selama 240 menit menunjukkan bahwa sistem beroperasi secara stabil tanpa mengalami *crash*, *hang*, atau gangguan performa yang signifikan sepanjang durasi pengujian. Suhu CPU Raspberry Pi 5 mengalami peningkatan bertahap dari 42°C pada awal pengujian menjadi 63°C setelah 240 menit operasional. Peningkatan suhu sebesar 21°C ini masih berada dalam batas operasional yang aman karena Raspberry Pi 5 dirancang untuk dapat beroperasi pada suhu CPU hingga 85°C sebelum melakukan *thermal throttling*, dan *shutdown* darurat baru terjadi di atas 95°C. Suhu stabil di kisaran 60-63°C setelah menit ke-120 mengindikasikan bahwa sistem mencapai keseimbangan termal (*thermal equilibrium*),

yaitu kondisi di mana panas yang dihasilkan komponen setara dengan panas yang dibuang melalui *heatsink* dan *fan*.

Dari sisi performa, FPS mengalami penurunan bertahap dari 28 FPS pada awal pengujian menjadi 24-25 FPS pada menit ke-150 hingga 240. Penurunan ini bersifat gradual, dengan variasi kecil (fluktuatif antara 24 dan 25 FPS), sehingga tidak menunjukkan degradasi performa yang tajam. Penurunan sekitar 3-4 FPS ini masih dalam batas toleransi operasional sistem *real time*, karena keseluruhan pengujian menunjukkan FPS tetap berada ≥ 24 FPS. Aktivitas servo dan audio tercatat aktif secara konsisten sejak menit ke-15 hingga akhir pengujian tanpa adanya anomali status. Tidak ditemukan indikasi gangguan seperti *delay respons*, *jitter* pada servo, maupun gangguan audio. Secara keseluruhan, sistem menunjukkan ketahanan operasional yang baik untuk penggunaan jangka panjang, dengan catatan bahwa pada kondisi lingkungan lapangan yang lebih ekstrem, potensi kenaikan suhu tetap perlu diantisipasi melalui manajemen termal tambahan.

4.7 Hasil User Acceptance Testing (UAT)

User Acceptance Testing (UAT) merupakan tahap evaluasi akhir yang bertujuan untuk mengukur tingkat penerimaan dan persepsi pengguna akhir terhadap sistem yang dikembangkan. Sebagaimana ditetapkan pada rancangan pengujian di Bab III, evaluasi UAT dilakukan menggunakan kuesioner berbasis Skala Likert 1–5 dengan 15 butir pernyataan yang dikelompokkan dalam lima dimensi, dan nilai rata-rata (*mean*) tiap dimensi diinterpretasikan berdasarkan kategorisasi rentang interval pada Tabel 3.9. Dalam konteks penelitian ini, pengguna akhir yang dimaksud adalah petani yang secara langsung berhadapan dengan permasalahan hama burung di lahan mereka. UAT dilaksanakan dengan mendemonstrasikan sistem secara langsung kepada petani di lokasi pengujian, kemudian setiap petani diminta mengisi formulir evaluasi berbasis *Google Form* yang terdiri dari pertanyaan kuantitatif dengan Skala Likert 1–5 dan pertanyaan kualitatif terbuka.

4.7.1 Hasil Evaluasi Kuantitatif UAT

Responden UAT terdiri dari 6 (enam) orang petani yang semuanya berjenis kelamin laki-laki, dengan profil usia berkisar antara 35 hingga 45 tahun (rata-rata 40.8 tahun) dan pengalaman bertani antara 13 hingga 23 tahun (rata-rata 18.5 tahun). Luas lahan yang dikelola bervariasi mulai dari kurang dari 0.5 hektar hingga lebih dari 2 hektar, yang mencerminkan keragaman skala usaha pertanian. Seluruh responden berlokasi di wilayah Cipetir dan Undrus Binangun, Kecamatan Kadudampit, Kabupaten Sukabumi, yang merupakan daerah pertanian

padi aktif dengan riwayat serangan hama burung yang signifikan. Homogenitas lokasi responden memastikan bahwa seluruh peserta memiliki konteks permasalahan yang serupa sehingga evaluasi mereka dapat dibandingkan secara adil.

Kuesioner UAT terdiri dari 15 butir pernyataan yang dikelompokkan dalam empat dimensi evaluasi: (1) Persepsi Manfaat (butir 1, 8, 9, 14, 15), (2) Kinerja Deteksi (butir 2, 3, 5), (3) Kinerja Aktuator Servo (butir 4), dan (4) Kinerja Audio (butir 6, 7), serta (5) Kegunaan Praktis (butir 10, 11, 12, 13). Data skor mentah tiap responden untuk setiap butir pertanyaan disajikan pada Tabel 4.7.

Tabel 4.9 Data Hasil Kuisisioner UAT

No.	Pertanyaan (Dimensi)	Budi	Ruslan	Deni	Kholid	Yusuf	Asap	Rata-rata
1	Sistem membantu mendeteksi hama burung (Manfaat)	5	4	4	5	4	4	4.33
2	Hasil deteksi terlihat akurat (Kinerja Deteksi)	4	5	4	5	4	4	4.33
3	Kamera mampu menangkap burung dengan jelas (Kinerja Deteksi)	4	5	4	4	4	4	4.17
4	Gerakan servo memperluas area pengamatan (Aktuator)	5	4	5	4	5	4	4.50
5	Respons alat terhadap burung terlihat cepat (Kinerja Deteksi)	5	5	4	4	4	4	4.33
6	Suara pengusir cukup efektif (Kinerja Audio)	4	3	3	5	4	5	4.00
7	Variasi suara mencegah habituasi burung (Kinerja Audio)	4	4	4	4	4	5	4.17
8	Lebih baik dari cara manual (Manfaat)	5	5	5	5	5	4	4.83
9	Mengurangi kebutuhan berjaga di sawah (Manfaat)	4	5	5	5	4	5	4.67
10	Mudah dipahami cara kerjanya (Kegunaan Praktis)	3	4	3	4	3	4	3.50

No.	Pertanyaan (Dimensi)	Budi	Ruslan	Deni	Kholid	Yusuf	Asap	Rata-rata
11	Layak digunakan di lahan pertanian nyata (Kegunaan Praktis)	4	4	4	3	4	5	4.00
12	Desain sesuai kebutuhan pengusiran (Kegunaan Praktis)	4	4	4	4	4	4	4.00
13	Penggunaan daya terlihat praktis (Kegunaan Praktis)	3	3	3	4	4	3	3.33
14	Berpotensi mengurangi kerugian hama burung (Manfaat)	5	4	4	5	5	4	4.50
15	Bersedia menggunakan alat ini (Manfaat)	4	4	4	4	5	4	4.17

Berdasarkan data pada Tabel 4.7, perhitungan nilai rata-rata per dimensi dilakukan menggunakan Persamaan 3.1 ($\text{Mean} = \text{Jumlah Bobot Penilaian} / \text{Total Responden}$), dengan total responden $N = 6$. Sebagai contoh perhitungan untuk dimensi Persepsi Manfaat (butir 1, 8, 9, 14, 15):

$$\text{Butir 1: Mean} = (5+4+4+5+4+4) / 6 = 26 / 6 = 4,33$$

$$\text{Butir 8: Mean} = (5+5+5+5+5+4) / 6 = 29 / 6 = 4,83$$

$$\text{Butir 9: Mean} = (4+5+5+5+4+5) / 6 = 28 / 6 = 4,67$$

$$\text{Butir 14: Mean} = (5+4+4+5+5+4) / 6 = 27 / 6 = 4,50$$

$$\text{Butir 15: Mean} = (4+4+4+4+5+4) / 6 = 25 / 6 = 4,17$$

$$\text{Mean Dimensi Persepsi Manfaat} = (4,33 + 4,83 + 4,67 + 4,50 + 4,17) / 5 = 22,50 / 5 = 4,50$$

Seluruh perhitungan rata-rata per dimensi dilakukan dengan cara yang sama. Rekapitulasi hasil perhitungan nilai rata-rata per dimensi beserta interpretasinya berdasarkan Tabel 3.9 disajikan pada Tabel 4.8.

Tabel 4.10 Data Hasil Kuisisioner UAT

Matriks Evaluasi	Item Pertanyaan	Rata-rata Dimensi	Persentase (%)	Interpretasi
Persepsi Manfaat	1, 8, 9, 14, 15	4.50	90,0%	Sangat Baik

Matriks Evaluasi	Item Pertanyaan	Rata-rata Dimensi	Persentase (%)	Interpretasi
Kinerja Deteksi	2, 3, 5	4.28	85,6%	Sangat Baik
Kinerja Aktuator Servo	4	4.50	90,0%	Sangat Baik
Kinerja Audio Pengusir	6, 7	4.08	81,7%	Baik
Kegunaan Praktis	10, 11, 12, 13	3.71	74,2%	Baik
Rata-Rata Keseluruhan	1–15	4.10	82,0%	Baik

Nilai persentase pada Tabel 4.8 dihitung menggunakan Persamaan 3.2 (Persentase = $(\text{Mean} / 5) \times 100\%$). Sebagai contoh, untuk dimensi Persepsi Manfaat: Persentase = $(4,50 / 5) \times 100\% = 90,0\%$. Rata-rata keseluruhan dihitung dari jumlah seluruh 15 butir dibagi 15: Mean keseluruhan = $(4,33 + 4,33 + 4,17 + 4,50 + 4,33 + 4,00 + 4,17 + 4,83 + 4,67 + 3,50 + 4,00 + 4,00 + 3,33 + 4,50 + 4,17) / 15 = 61,83 / 15 = 4,12 \approx 4,10$.

Berdasarkan Tabel 4.8, sistem mendapatkan penilaian yang positif di seluruh dimensi evaluasi. Rata-rata keseluruhan UAT sebesar 4,10 berada dalam kategori "Baik" (rentang 3,41–4,20 sesuai Tabel 3.9), yang menandakan bahwa petani secara umum menerima sistem ini dengan baik setelah menyaksikan demonstrasinya. Dimensi Persepsi Manfaat dan Kinerja Aktuator Servo mendapatkan skor tertinggi dengan rata-rata 4,50 (kategori "Sangat Baik"), mengindikasikan bahwa petani sangat menghargai kemampuan sistem untuk mengurangi beban kerja manual dan fitur gerakan servo yang memperluas area pemantauan. Nilai 4,83 pada butir "Lebih baik dari cara manual" merupakan nilai tertinggi dalam seluruh kuesioner, yang secara kuat mengonfirmasi bahwa sistem ini memiliki keunggulan yang jelas dan dirasakan langsung oleh petani dibandingkan metode konvensional.

Dimensi Kinerja Deteksi mendapatkan rata-rata 4,28 (kategori "Sangat Baik"), yang mengonfirmasi persepsi petani bahwa sistem mampu menangkap dan mendeteksi burung dengan akurasi yang memuaskan. Sementara itu, dimensi Kinerja Audio mendapatkan rata-rata 4,08 (kategori "Baik"), dengan variasi nilai yang cukup beragam di antara responden. Responden Deni memberikan nilai 3 (Netral) untuk efektivitas suara, yang sejalan dengan komentarnya bahwa suara sudah meyakinkan namun kerasan speaker perlu dipertimbangkan lebih lanjut. Dimensi Kegunaan Praktis mendapatkan skor terendah dengan rata-rata 3,71 (masih dalam kategori "Baik"), terutama didorong oleh nilai rendah pada butir penggunaan

daya (rata-rata 3,33). Ini merupakan temuan yang konsisten dan kritis: seluruh petani menyoroti ketersediaan sumber listrik PLN sebagai keterbatasan utama sistem, mengingat sebagian besar sawah tidak memiliki akses listrik langsung.

Hasil ini konsisten dengan penelitian UAT pada sistem teknologi terapan lainnya. Hermansah, Murhadi, & Saputro (2025) melaporkan rata-rata penerimaan sistem informasi *softskill* sebesar 78,3% dengan kategori "Baik", sedangkan Aliyah, Hartono, & Muin (2025) memperoleh rata-rata 87% (kategori "Sangat Baik") untuk sistem informasi keuangan dan inventaris. Hasil UAT penelitian ini sebesar 4,10 (82,0%) yang masuk kategori "Baik" menunjukkan tingkat penerimaan yang kompetitif untuk sebuah prototipe teknologi baru yang diperkenalkan kepada petani untuk pertama kalinya.

4.7.2 Hasil Evaluasi Kualitatif UAT

Selain penilaian kuantitatif, kuesioner UAT juga memuat lima pertanyaan kualitatif terbuka yang memberikan kesempatan kepada responden untuk mengekspresikan pendapat, saran, dan kritik secara lebih bebas. Masukan kualitatif ini sangat berharga karena mengungkapkan dimensi penilaian yang tidak selalu tertangkap oleh pertanyaan terstruktur dengan skala Likert. Berikut adalah rangkuman tema-tema utama yang muncul dari jawaban kualitatif responden.

Kelebihan utama yang paling sering disebutkan oleh responden adalah kemampuan sistem untuk bekerja secara otomatis tanpa membutuhkan kehadiran manusia secara terus-menerus. Petani Budi menyebutkan bahwa "alat ini otomatis digunakan sehingga bisa mengurangi tenaga kerja untuk orang berjaga di sawah", sementara Yusuf mengapresiasi bahwa "ketika ada burung bisa cepat nyala dan membunyikan suara tanpa delay". Petani Asap menggunakan kata "canggih dan otomatis" untuk mendeskripsikan kesan pertamanya terhadap sistem. Apresiasi terhadap otomatisasi ini sangat konsisten dengan temuan dari survei kebutuhan awal yang menunjukkan bahwa seluruh petani responden (100%) menyatakan membutuhkan alat yang dapat bekerja otomatis tanpa dijaga terus-menerus.

Masukan paling dominan berkaitan dengan aspek sumber daya listrik. Seluruh responden atau sebesar 100% dari peserta UAT menyebutkan keterbatasan akses listrik PLN sebagai bagian yang perlu diperbaiki atau sebagai saran pengembangan. Petani Budi menyarankan penggunaan panel surya, sementara petani Yusuf menyatakan "sumber listriknya pake baterai atau panel surya jangan pake listrik PLN langsung". Temuan ini secara konsisten menunjukkan bahwa integrasi sumber energi alternatif (solar panel atau baterai) merupakan kebutuhan yang paling mendesak untuk membuat sistem ini dapat diterapkan secara praktis di lapangan.

Masukan kedua yang muncul dari beberapa responden (Ruslan dan Kholid) adalah mengenai jangkauan deteksi kamera dan volume speaker yang perlu ditingkatkan agar satu unit alat dapat mencakup area sawah yang lebih luas.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, dapat disimpulkan bahwa sistem deteksi dan pengusir hama burung berbasis edge AI berhasil dikembangkan sebagai prototipe yang mampu bekerja secara otomatis dan real time. Sistem yang dibangun mampu mengintegrasikan komponen deteksi berbasis kamera, pemrosesan pada perangkat edge, serta mekanisme aktuasi berupa gerakan pan-tilt dan respons audio dalam satu alur kerja yang berjalan tanpa intervensi manual. Pengujian fungsional menunjukkan bahwa seluruh komponen sistem dapat beroperasi sesuai dengan skenario yang dirancang.

Kemampuan sistem dalam mengenali hama burung menunjukkan hasil yang baik pada kondisi pengujian, di mana objek burung dari genus *Lonchura* dapat dideteksi secara konsisten pada jarak hingga 400 cm pada pengujian citra dan 200 cm pada pengujian dengan objek nyata. Respons pengusiran berbasis audio yang diaktifkan secara event-driven juga mampu memicu reaksi menghindar pada burung dalam waktu singkat. Hasil ini menunjukkan bahwa sistem tidak hanya mampu melakukan deteksi visual, tetapi juga mampu memberikan respons pengusiran yang relevan terhadap keberadaan objek target.

Dari sisi model deteksi, hasil evaluasi menunjukkan bahwa YOLOv11n lebih sesuai untuk diterapkan pada sistem yang dikembangkan dibandingkan YOLOv8n. YOLOv11n memiliki recall, mAP@0.50, dan mAP@0.50-0.95 yang lebih tinggi, jumlah false positive yang lebih rendah, serta ukuran model dan beban komputasi yang lebih ringan. Model ini juga mampu mempertahankan kecepatan inferensi pada kisaran 24-28 FPS setelah diimplementasikan pada Raspberry Pi AI Camera berbasis Sony IMX500. Dengan demikian, sistem telah memenuhi kebutuhan pemrosesan real time untuk aplikasi deteksi objek pada lingkungan persawahan.

Meskipun demikian, sistem masih memiliki keterbatasan pada aspek selektivitas deteksi. Nilai precision YOLOv11n sebesar 85,19% menunjukkan bahwa masih terdapat kemungkinan deteksi terhadap objek yang bukan target. Pada skenario pengujian tambahan, model masih dapat merespons objek burung non-target yang memiliki kemiripan visual dengan hama burung. Hal ini menunjukkan bahwa model belum sepenuhnya mampu membedakan hama burung pemakan padi dari genus *Lonchura* spp. dengan objek burung lain yang tidak termasuk dalam batasan penelitian. Kondisi tersebut dipengaruhi oleh komposisi dataset negatif/null yang masih terbatas dibandingkan jumlah data positif hama_burung. Oleh karena itu, hasil

penelitian ini menunjukkan bahwa sistem sudah mampu menjalankan fungsi deteksi dan pengusiran secara otomatis, tetapi selektivitas model masih perlu ditingkatkan melalui penambahan data negatif dan hard negative yang lebih representatif.

Tingkat penerimaan pengguna terhadap sistem juga menunjukkan hasil yang baik berdasarkan pengujian *User Acceptance Testing* (UAT). Pengguna menilai bahwa sistem mampu membantu mengurangi kebutuhan penjagaan manual serta memberikan respons yang lebih konsisten dibandingkan metode konvensional. Namun, ketergantungan sistem pada sumber listrik PLN masih menjadi keterbatasan yang perlu diperhatikan pada pengembangan selanjutnya agar sistem dapat digunakan secara lebih fleksibel di berbagai kondisi lahan persawahan.

5.2 Saran

Berdasarkan keterbatasan penelitian dan peluang pengembangan yang teridentifikasi selama proses penelitian, terdapat beberapa saran yang dapat dijadikan acuan bagi penelitian lanjutan.

Pertama, pengujian lapangan jangka panjang perlu dilakukan pada skala sawah yang lebih luas dan dalam rentang waktu yang mencakup seluruh fase kritis tanaman padi, terutama fase pembungaan hingga pematangan bulir yang merupakan periode serangan hama paling intensif. Pengujian jangka panjang diperlukan untuk mengetahui konsistensi performa sistem terhadap variasi kondisi lingkungan, intensitas cahaya, perubahan cuaca, jarak objek, serta dinamika perilaku burung di lahan persawahan.

Kedua, dataset negatif/null perlu diperbanyak pada tahap pelatihan, validasi, dan pengujian agar distribusi data menjadi lebih seimbang. Data negatif tidak cukup hanya berupa gambar sawah tanpa burung, tetapi juga perlu mencakup objek lingkungan pertanian yang berpotensi memicu salah deteksi, seperti manusia, alat pertanian, ranting, daun, orang-orangan sawah, serta burung non-target seperti elang, burung hantu, atau jenis burung lain yang tidak termasuk dalam genus *Lonchura* spp. Penambahan data negatif dan hard negative tersebut penting agar model tidak hanya belajar mengenali keberadaan burung, tetapi juga mampu membedakan hama burung pemakan padi dari objek lain yang memiliki kemiripan visual.

Ketiga, penelitian selanjutnya dapat melakukan pelatihan ulang model dengan dataset yang lebih representatif dan menambahkan skenario pengujian khusus untuk objek non-target. Pengujian ini diperlukan untuk mengukur kemampuan model dalam menolak objek yang bukan hama burung target. Selain itu, penyesuaian nilai confidence threshold juga dapat dilakukan

untuk menekan false positive tanpa menurunkan kemampuan sistem dalam mendeteksi objek hama secara signifikan. Jika sistem ingin diarahkan lebih spesifik terhadap hama burung dari genus *Lonchura* spp., pengembangan berikutnya dapat mempertimbangkan skema klasifikasi yang membedakan antara burung target dan burung non-target.

Keempat, ketergantungan sistem pada sumber daya listrik PLN merupakan keterbatasan yang perlu diatasi agar sistem dapat dioperasikan di lokasi sawah yang tidak memiliki akses jaringan listrik. Pengembangan sumber daya alternatif berbasis panel surya yang dikombinasikan dengan baterai penyimpanan energi merupakan solusi yang perlu dieksplorasi pada penelitian lanjutan, sehingga sistem dapat beroperasi secara mandiri dan lebih andal di berbagai kondisi geografis.

Kelima, integrasi sistem dengan fitur notifikasi dan pemantauan jarak jauh dapat menjadi nilai tambah pada penelitian lanjutan. Penambahan modul komunikasi nirkabel yang memungkinkan petani memantau status sistem dan menerima notifikasi serangan hama secara real time melalui perangkat seluler akan meningkatkan ketergunaan sistem bagi pengguna akhir. Selain itu, fitur pemantauan juga dapat membuka peluang pengumpulan data serangan hama secara lebih luas untuk mendukung analisis pola serangan pada pengembangan berikutnya.

DAFTAR PUSTAKA

- Aboltins, A., Pikulins, D., Grizans, J., & Tjukovs, S. (2021). Piscivorous Bird Deterrent Device Based on a Direct Digital Synthesis of Acoustic Signals. *Elektronika Ir Elektrotechnika*, 27(6), 42–48. <https://doi.org/10.5755/j02.eie.28977>
- Adafruit. (2023). Adafruit IO documentation.
- Afif, M. H., Sanjaya, R., Sauri, S., & Prasetyo, S. M. (2023). *Sistem Perangkat Pengusir Hama Burung Emprit Atau Pipit Berbasis Sensor PIR Dan IoT*. <https://journal.mediapublikasi.id/index.php/logic>
- Aliyah, Nahrin Hartono, & Asrul Azhari Muin. (2024). Penggunaan User Acceptance Testing (UAT) Pada Pengujian Sistem Informasi Pengelolaan Keuangan Dan Inventaris Barang. *Switch : Jurnal Sains Dan Teknologi Informasi*, 3(1), 84–100. <https://doi.org/10.62951/switch.v3i1.330>
- Ardjansyah, A., Budi Hernowo, J., & Priyambodo, S. (2017). PENGARUH SERANGAN BURUNG BONDOL TERHADAP KERUSAKAN TANAMAN PADI DI BOGOR (The Influences of Bondol Attack Against Paddy Damage in Bogor). In *Agustus* (Vol. 22, Number 2).
- Arowolo, O., Adekunle, A. A., & Ade-Omowaye, J. A. (2020). A Real time Image Processing Bird Repellent System Using Raspberry Pi. *FUOYE Journal of Engineering and Technology*, 5(2). <https://doi.org/10.46792/fuoyejet.v5i2.496>
- Chen, Q., Xie, J., Yu, Q., Liu, C., Ding, W., Li, X., & Zhou, H. (2024). An experimental study of acoustic bird repellents for reducing bird encroachment in pear orchards. *Frontiers in Plant Science*, 15. <https://doi.org/10.3389/fpls.2024.1365275>
- Fabre, W., Haroun, K., Lorrain, V., Lepecq, M., & Sicard, G. (2024). From Near-Sensor to In-Sensor: A State-of-the-Art Review of Embedded AI Vision Systems. In *Sensors* (Vol. 24, Number 16). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/s24165446>
- Fang, S., Shen, Y., Zou, H., Yin, Y., Jin, W., & Zhou, H. (2025). Birds-YOLO: A Bird Detection Model for Dongting Lake Based on Modified YOLOv11. *Biology*, 14(11). <https://doi.org/10.3390/biology14111515>
- Fawwaz, F. M. R., Fadhlurrahman, R., Eka Putri, N., Akmalano, R., Alif Artanto, G., & Hidayat, R. (2024). Radar Wing Defender (RWD): Sistem untuk membantu para petani dalam mendeteksi dan mengusir hama burung. *Jurnal Komputer Dan Elektro Sains*, 3(1), 16–22. <https://doi.org/10.58291/komets.v3i1.180>

- Fiolka, J., Bernacki, K., Farah, A., & Popowicz, A. (2023). Multi-Wavelength Biometric Acquisition System Utilizing Finger Vasculature NIR Imaging. *Sensors*, 23(4). <https://doi.org/10.3390/s23041981>
- Hasugian, H. (2023). *USER ACCEPTANCE TESTING (UAT) PADA ELECTRONIC DATA PREPROCESSING GUNA MENGETAHUI KUALITAS SISTEM*. 4(1), 20–27.
- Hermansah, L., Tjahjo Saputro, W., Studi Teknologi Informasi, P., Teknik, F., Muhamamdiyah Purworejo, U., Ahmad Dahlan, J. K., Purworejo, K., & Tengah, J. (2025). *Sistemasi: Jurnal Sistem Informasi User Acceptance Testing Guna mengetahui Reseptivitas Pengguna terhadap Sistem Informasi Pelatihan Softskill User Acceptance Testing to Assess User Receptiveness Toward a Soft Skills Training Information System* (Vol. 14, Number 5). <http://sistemasi.ftik.unisi.ac.id>
- Hobbs, M. T., & Brehme, C. S. (2017). An improved camera trap for amphibians, reptiles, small mammals, and large invertebrates. *PLoS ONE*, 12(10). <https://doi.org/10.1371/journal.pone.0185026>
- Hussain, M., & Khanam, R. (2024). In-Depth Review of YOLOv1 to YOLOv10 Variants for Enhanced Photovoltaic Defect Detection. In *Solar* (Vol. 4, Number 3, pp. 351–386). Multidisciplinary Digital Publishing Institute (MDPI). <https://doi.org/10.3390/solar4030016>
- Hussein, B. M., & Shareef, S. M. (2024). An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning. *ITM Web of Conferences*, 64, 01003. <https://doi.org/10.1051/itmconf/20246401003>
- Irianto, K. D. (2022). Design of Smart Farm Irrigation Monitoring System Using IoT and LoRA. *Jurnal RESTI*, 6(1), 47–56. <https://doi.org/10.29207/resti.v6i1.3707>
- Jalaludin, R., & Laksmiati, D. (2023). Perancangan Sistem Kendali Irigasi Otomatis dan Pengusir Hama Burung Dengan Menggunakan Sensor PIR. *Jurnal Ilmiah Telsinas Elektro, Sipil Dan Teknik Informasi*, 6(2), 122–134. <https://doi.org/10.38043/telsinas.v6i2.4565>
- J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91. keywords: {Computer architecture;Microprocessors;Object detection;Training;Real-time systems;Neural networks;Pipelines},

- Kusuma, A. P., & Yufron, A. (2024). ANALYSIS OF USER ACCEPTANCE TESTING ON A SHIPPING APPLICATION TO DETERMINE THE QUALITY OF THE SYSTEM. *Antivirus : Jurnal Ilmiah Teknik Informatika*, 18(2), 234–243. <https://doi.org/10.35457/antivirus.v18i2.4002>
- Maspupah, A. (2024). Literature Review: Advantages and Disadvantages of Black Box and White Box Testing Methods. *Techno Nusa Mandiri: Journal of Computing and Information Technology*, 21(2), 151 hingga 162. DOI: 10.33480/techno.v21i2.5776.
- Minani, J. B. (2025). *Functional End-to-End Testing of IoT Systems*.
- MPLK Politeknik Negeri Kupang. (2023). *Burung pemakan padi Lonchura sp.* Diakses dari <https://mplk.politanikoe.ac.id/index.php/opt-tanaman/vertebrata-hama-pertanian/1369-burung-pemakan-padi-lonchura-sp-lainnya>
- Muhammad, D. S. A., Husodo, A. Y., & Huwae, R. B. (2023). *Rancang Bangun Smart Farming System Monitoring dan Pengusir Hama Burung Berbasis Raspberry Pi 4 dengan Pengaplikasian Algoritma YOLOv5 Design and Development of Smart Farming System for Monitoring and Bird Pest Control Based on Raspberry Pi 4 with Implementation of YOLOv5 Algorithm*.
- Wuli, N. R., Puspita, V. A., Noa, M., Tinggi, S., Flores, P., Program, B.-N.-N., & Agroteknologi, S. (2025). IDENTIFIKASI JENIS HAMA PADA TANAMAN PADI (ORYZA SATIVA L) VARIETAS INPARI 40 DI DESA PAPE KECAMATAN BAJAWA KABUPATEN NGADA. *JURNAL PERTANIAN UNGGUL*, 4. <https://ejournal.stiperfb.ac.id/index.php/jurnalpertanianunggul/article/view/168>
- Nugroho, B., Azhar, N. F., Pratama, B. M., Syakbani, A. R. A., Wibowo, D. R., & Syam, A. M. A. R. (2025). Implementation of Sparrow Pest Detection Using YOLOv8 Method on Raspberry Pi and Google Coral USB Accelerator. *Sebatik*, 29(1), 129–135. <https://doi.org/10.46984/sebatik.v29i1.2558>
- Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. *International Journal of Scientific and Research Publications*, 13(8), 120–124. <https://doi.org/10.29322/ijsrp.13.08.2023.p14015>
- Puryono, D. A., & Pangestu, E. P. (2023). Application of Rice Pest Repellent Using ESP32 Microcontroller and Internet of Things Based Sound Wave. *East Asian Journal of*

- Multidisciplinary Research*, 2(10), 4261–4272.
<https://doi.org/10.55927/eajmr.v2i10.6228>
- Royce, W.W. (1970) Managing the Development of Large Software Systems. Proceedings of IEEE WESCON, 26, 328-388
- Safinatun, F., Dan Nor, N., & Astriyani, R. (n.d.). *Sistem Kontrol dan Monitoring Tinggi Air Serta Pengusir Hama Burung pada Tanaman Padi Berbasis Internet of Things Dan Android*.
- Saravanos, A., & Curinga, M. X. (2023). Simulating the Software Development Lifecycle: The Waterfall Model. *Applied System Innovation*, 6(6). <https://doi.org/10.3390/asi6060108>
- Shende, A. P., & Anande, R. S. (2019). Motion Imitation Robotic Arm, (MIRA). *International Research Journal of Engineering and Technology*. www.irjet.net
- Storms, R. F., Carere, C., Musters, R., Van Gasteren, H., Verhulst, S., & Hemelrijk, C. K. (2022). Deterrence of birds with an artificial predator, the RobotFalcon. *Journal of the Royal Society Interface*, 19(195). <https://doi.org/10.1098/rsif.2022.0497>
- Syarif Hidayat, A., & Kusuma Raharja, W. (2022). Design and Build a Camera Directional Movement Control Device based on Sound Source Position Detection. In *International Journal of Computer Applications* (Vol. 183, Number 51).
- Szeliski, R. (2010). *Computer vision: Algorithms and applications*. Springer.
<https://doi.org/10.1007/978-1-84882-935-0>
- Thabibi, H., Wati, S. F. A., & Rinjeni, T. P. (2025). Implementasi User Acceptance Testing (UAT) Pada Website E-Commerce UMKM BBhealthy. *Adopsi Teknologi Dan Sistem Informasi (ATASI)*, 4(1), 19–26. <https://doi.org/10.30872/atasi.v4i1.2904>
- Tuti, H. K., Sepe, M., Romadhan, P., Rido, M., & Regina, N. A. (2024). EDUKASI PENDEKATAN PARTISIPATIF DALAM PENGENDALIAN HAMA BURUNG DI PERTANAMAN PADI DI KECAMATAN TAMBAN CATUR KABUPATEN KAPUAS KALIMANTAN TENGAH. *BAKIRA: Jurnal Pengabdian Kepada Masyarakat*, 5(2), 115–123. <https://doi.org/10.30598/bakira.2024.5.2.115-123>
- Ultralytics, "YOLOv8 Documentation: Dataset Format and Split Recommendations," Ultralytics Docs, 2024. [Online]. Available: <https://docs.ultralytics.com>.
- Ultralytics. (2024). YOLO11 Overview. Diakses dari <https://docs.ultralytics.com/models/yolo11/>
- Wahyuni, E. D., Kom, S., Kom, M., Ramadha, F. N., Deo, D., Septa, V., Mi'raj, L., & Rachmadi, F. (2024). *SDLC Big Bang dan Waterfall : Perbandingan Pendekatan dalam*

Pengembangan Perangkat Lunak (Vol. 18).
<https://journal.fkom.uniku.ac.id/ilkom41> Terakreditasi SINTA 5

LAMPIRAN

Lampiran Observasi Lapangan



Lampiran Transkrip Wawancara Observasi

Lokasi Lokasi: Desa Undrus Binangun, Cipetir, Kecamatan Kadudampit, Kabupaten Sukabumi

Narasumber

- Narasumber 1: Lukmanul Hakim (Usia: 39 Tahun, Pengalaman Bertani: ± 19 Tahun, Luas Lahan: ± 1 Hektar)
- Narasumber 2: Jamaludin (Usia: 40 Tahun, Pengalaman Bertani: ± 20 Tahun, Luas Lahan: ± 0.5 Hektar)

Transkrip Wawancara

Arga: Selamat siang Bapak. Mohon sebutkan nama Bapak berdua. **Jamaludin:** Nama saya Jamaludin. **Lukmanul Hakim:** Nama saya Lukmanul Hakim.

Arga: Berapa usia Bapak saat ini dan sudah berapa lama Bapak bekerja sebagai petani?

Lukmanul Hakim: Usia saya 39 tahun. Saya sudah bertani sejak usia sekitar 20 tahun.

Jamaludin: Saya berusia 40 tahun dan juga sudah mulai bertani sejak usia sekitar 20 tahun.

Arga: Berapa luas lahan pertanian yang Bapak kelola saat ini? **Lukmanul Hakim:** Saya mengelola lahan seluas kurang lebih 1 hektar. **Jamaludin:** Saya mengelola lahan sekitar setengah hektar.

Arga: Terkait budidaya padi, apakah lahan Bapak sering mengalami serangan hama burung?

Lukmanul Hakim: Ya, sering. Terutama serangan burung pipit.

Arga: Seberapa sering burung tersebut datang dan pada jam berapa biasanya mereka menyerang? **Jamaludin:** Biasanya mulai sekitar pukul 10 pagi. **Lukmanul Hakim:** Serangan terjadi setiap hari. Waktunya tidak menentu, tetapi umumnya dari pagi hingga siang hari. Sore hari jarang terjadi.

Arga: Pada fase apa serangan burung ini terjadi? Apakah sejak awal penanaman? **Lukmanul Hakim:** Tidak, pada awal tanam biasanya yang menyerang adalah ulat. Serangan burung terjadi secara spesifik pada fase pengisian gabah.

Arga: Spesies burung apa saja yang biasanya menjadi hama utama di sini? **Lukmanul Hakim:** Burung pipit, bondol, dan gereja.

Arga: Apakah ada hama lain selain burung? **Lukmanul Hakim:** Ada hama kungkang. Bentuknya kurus dan panjang, hama ini menyerang bagian batang padi secara langsung.

Arga: Jika lahan diserang hama burung, berapa persentase kerusakan atau penurunan hasil panen yang terjadi? **Jamaludin:** Kerugiannya bisa sangat besar, mencapai 40 persen hingga 50 persen dari total panen. Jika tidak dijaga sama sekali, hasil panen bisa habis.

Arga: Metode apa yang saat ini Bapak gunakan untuk menanggulangi serangan burung tersebut? **Lukmanul Hakim:** Kami menggunakan jaring pelindung di atas lahan. **Jamaludin:** Selain itu, kami juga membunyikan kaleng secara manual untuk menakuti mereka.

Arga: Seberapa efektif metode jaring dan bunyi kaleng tersebut? **Lukmanul Hakim:** Efektivitasnya hanya sekitar 50 persen. Burung-burung tersebut perlahan terbiasa dengan suara kaleng dan tetap kembali.

Arga: Saya saat ini sedang merancang alat pengusir burung otomatis. Sistem ini menggunakan kamera untuk mendeteksi kedatangan burung. Saat terdeteksi, alat ini akan mengeluarkan suara yang bervariasi secara otomatis agar burung tidak terbiasa dengan satu jenis suara. Dari sudut pandang Bapak, fitur apa yang paling dibutuhkan dari alat seperti ini? **Lukmanul Hakim:** Fitur suara yang berubah-ubah itu yang paling dibutuhkan.

Arga: Untuk operasional alat, apakah terdapat sumber listrik yang dapat diakses di sekitar area persawahan ini? **Lukmanul Hakim:** Tidak ada. Jarak ke sumber listrik sangat jauh dari lahan kami.

Arga: Sebagai penutup, apakah ada hama lain yang tingkat kerusakannya sama parahnya dengan burung? **Jamaludin:** Hama tikus. Tikus sangat berbahaya karena mereka memotong langsung batang padi. Serangannya bisa menghabiskan banyak lahan sekaligus. Tingkat keparahannya bisa lebih tinggi dibandingkan burung.

Narasumber: Dahlan (Usia: 66 Tahun)

Transkrip Wawancara

Arga: Mohon sebutkan nama asli Bapak. **Dahlan:** Nama asli saya Dahlan.

Arga: Berapa usia Bapak saat ini? **Dahlan:** Saat ini usia saya sudah mencapai 66 tahun.

Arga: Sejak usia berapa Bapak memulai profesi sebagai petani? **Dahlan:** Saya mulai belajar bertani dari orang tua sekitar usia 15 tahun, tepat setelah lulus Sekolah Dasar. Saya mulai bertani secara penuh pada rentang usia 20 hingga 25 tahun.

Arga: Apakah Bapak secara rutin membudidayakan padi? **Dahlan:** Ya, saya rutin menanam padi.

Arga: Selama masa budidaya padi tersebut, apakah lahan Bapak secara konsisten mengalami gangguan dari hama burung? **Dahlan:** Pasti. Burung adalah hama yang konsisten.

Arga: Selain burung, hama jenis apa lagi yang secara signifikan merusak lahan pertanian di sini? **Dahlan:** Hama tikus. Selain itu, ada juga hama sejenis serangga yang merusak tanaman hingga bagian tengahnya menjadi berlubang.

Arga: Apakah serangga tersebut adalah hama kungkang? Berdasarkan pengalaman Bapak, di antara tikus, burung, atau kungkang, mana yang dampaknya paling destruktif? **Dahlan:** Tikus. Serangan tikus adalah yang paling parah dan merusak.

Arga: Bagaimana dengan serangan burung? Apakah kerugiannya juga bisa mencapai tingkat yang sangat parah? **Dahlan:** Tingkat keparahannya sangat bergantung pada pola tanam serempak. Jika penanaman di suatu wilayah tidak dilakukan bersamaan, serangan burung akan bertahap dan terfokus pada lahan yang siap panen terlebih dahulu. Jika penanaman serempak, kawanan burung akan berpindah tempat secara menyebar, sehingga kerugian terbagi. Konsep penyebarannya sama seperti hama pada komoditas sayuran.

Arga: Pada periode waktu kapan biasanya kawanan burung tersebut menyerang? Pagi, siang, atau sore hari? **Dahlan:** Bisa terjadi pada pagi maupun sore hari. Hal ini sangat bergantung

pada kondisi cuaca. Jika cuaca sangat terik, burung tidak akan muncul. Mereka justru sangat aktif menyerang saat cuaca sedang mendung.

Arga: Apakah serangan secara spesifik terjadi mulai dari fase pengisian gabah? **Dahlan:** Benar. Begitu gabah mulai terisi, kawanan burung akan mulai memakan hasilnya sedikit demi sedikit secara berkelanjutan.

Arga: Spesies burung apa yang biasanya paling mendominasi serangan di lahan ini? **Dahlan:** Umumnya didominasi oleh burung pipit.

Arga: Secara persentase atau volume, berapa banyak estimasi kerugian hasil panen akibat serangan burung tersebut? **Dahlan:** Seperti yang saya jelaskan sebelumnya, persentase kerugian berbanding lurus dengan status penanaman serempak. Jika penanaman serempak di berbagai lahan, volume kerugian per lahan tidak begitu besar. Namun, jika hanya sedikit lahan yang siap panen di satu area, kerugiannya sangat parah dan panen bisa habis total. Risikonya sangat besar.

Arga: Pendekatan atau metode apa yang biasa Bapak terapkan untuk menanggulangi hama tersebut? **Dahlan:** Pada masa lalu, saya pernah menggunakan pecahan karbit yang dilempar ke tengah lahan pertanian sebagai pengusir. Untuk saat ini, penanganan hama burung difokuskan pada penggunaan jaring pelindung secara fisik.

Arga: Sebelum ketersediaan jaring pelindung memadai, bagaimana Bapak menangani kawanan burung? **Dahlan:** Dahulu kami hanya mengandalkan instrumen penghasil suara bising secara manual. Namun, metode itu kurang efektif karena burung akan tetap kembali jika lahan tidak dijaga secara terus menerus oleh manusia.

Arga: Apakah terdapat infrastruktur jaringan listrik yang dapat diakses di area lahan persawahan ini? **Dahlan:** Belum ada akses listrik di tengah lahan. Fasilitas listrik hanya tersedia untuk lahan yang posisinya sangat dekat dengan area pemukiman warga.

Arga: Saat ini saya sedang merancang sistem alat pengusir burung menggunakan visi komputer. Kamera akan mendeteksi objek burung secara otomatis, kemudian memicu keluaran

suara yang bervariasi seperti suara manusia atau predator melalui pelantang suara agar burung tidak mengalami habituasi. Alat ini dikembangkan sebagai pemenuhan kewajiban akademis untuk manajemen penanggulangan hama. **Dahlan:** Sebagai gambaran realitas di lapangan saat ini, setiap lahan padi yang mendekati masa panen hampir pasti tertutup oleh jaring. Mayoritas petani menghamparkan jaring di atas tanaman. Meskipun demikian, tetap saja ada satu atau dua ekor burung yang berhasil menembus masuk, hanya saja skalanya tidak separah dahulu. Burung memiliki kecerdasan untuk lari dan bersembunyi di area rerumputan sekitarnya. Oleh karena itu, tenaga manusia tetap dibutuhkan secara berkala untuk berjaga di lahan.

Narasumber: Yusuf (Usia: 39 Tahun)

Transkrip Wawancara

Arga: Mohon sebutkan nama Bapak. **Yusuf:** Nama saya Yusuf.

Arga: Berapa usia Bapak saat ini? **Yusuf:** 39 tahun.

Arga: Apakah Bapak memiliki pengalaman langsung dalam menanam dan mengelola komoditas padi? **Yusuf:** Ya, saya memiliki pengalaman bertani padi dari kampung halaman.

Arga: Selama masa tanam padi, apakah Bapak pernah mengalami kendala atau gangguan dari hama burung? **Yusuf:** Tentu. Hama yang paling umum adalah burung pipit.

Arga: Selain burung pipit, apakah ada spesies burung lain yang sering menyerang? **Yusuf:** Ya, ada burung gereja. Jadi umumnya burung pipit dan burung gereja.

Arga: Bagaimana pola kedatangan burung tersebut? Apakah mereka datang setiap hari, sesekali, atau memiliki jadwal tertentu? **Yusuf:** Mereka datang setiap hari, tetapi hanya pada masa menjelang panen. Jika padi belum memasuki masa panen, burung tidak ada.

Arga: Pada waktu apa biasanya kawanan burung tersebut datang menyerang? Pagi, siang, atau sore? **Yusuf:** Umumnya serangan terjadi pada pagi dan sore hari.

Arga: Artinya, serangan terfokus pada fase pengisian bulir padi (masak susu). Seberapa besar dampak kerugian yang ditimbulkan oleh hama burung ini terhadap hasil panen? **Yusuf:** Dampaknya sangat signifikan. Hasil panen yang didapatkan bisa sangat sedikit jika diserang.

Arga: Metode apa yang biasanya Bapak gunakan untuk mengusir burung? Apakah menggunakan orang-orangan sawah atau membunyikan kaleng? **Yusuf:** Biasanya kami menggunakan orang-orangan sawah.

Arga: Apakah Bapak pernah mencoba menggunakan instrumen penghasil suara seperti kaleng? **Yusuf:** Ya, saya pernah mencobanya.

Arga: Bagaimana dengan penggunaan jaring pelindung (net)? **Yusuf:** Pernah juga. Terkadang kami mengombinasikan jaring dengan metode suara.

Arga: Dari ketiga metode tersebut (orang-orangan sawah, suara, dan jaring), mana yang menurut Bapak paling efektif? **Yusuf:** Menggunakan jaring adalah yang paling efektif karena menutupi area sawah.

Arga: Apakah metode jaring tersebut sudah sepenuhnya efektif menghentikan burung, atau masih ada burung yang berhasil menembus dan memakan padi? **Yusuf:** Dengan jaring, burung hampir tidak mungkin masuk. Namun, untuk hasil maksimal, tetap diperlukan tenaga manusia untuk menjaga dan mengusir sisa burung yang mendekat secara manual.

Arga: Saya sedang mengembangkan inovasi alat pengusir burung otomatis berbasis kamera pintar. Sistem ini dapat mendeteksi kedatangan burung dan secara otomatis mengeluarkan suara untuk mengusirnya. Menurut perspektif Bapak sebagai petani, apakah alat ini memiliki potensi yang baik? **Yusuf:** Menurut saya, itu ide yang sangat bagus.

Arga: Jika alat tersebut diimplementasikan di lahan, di titik mana penempatan yang paling optimal? Di tengah sawah atau di area pinggir? **Yusuf:** Lebih optimal ditempatkan di tengah sawah. Dengan begitu, alat tersebut bisa berputar dan menjangkau seluruh area.

Narasumber: ling (Usia: 36 Tahun)

Transkrip Wawancara

Arga: Mohon sebutkan nama dan usia Bapak. **ling:** Nama saya ling. Usia saya 36 tahun.

Arga: Terkait budidaya padi, hama jenis apa yang umumnya menyerang tanaman? **ling:** Hama wereng dan kungkang.

Arga: Bisa dijelaskan lebih spesifik mengenai hama kungkang? **ling:** Hama tersebut sejenis serangga yang memiliki anatomi kaki panjang.

Arga: Selain serangga, apakah Bapak pernah mengalami kerugian akibat serangan hama burung? **ling:** Ya, pernah.

Arga: Kapan biasanya kawanan burung tersebut datang menyerang? Apakah setiap hari atau pada periode tertentu saja? **ling:** Serangan terjadi pada fase masak susu, yaitu ketika bulir padi sudah mulai terisi.

Arga: Pada rentang waktu kapan burung tersebut menyerang dalam sehari? Pagi, siang, atau sore? **ling:** Serangannya terjadi secara terus-menerus sepanjang hari.

Arga: Spesies burung apa saja yang biasanya menjadi hama utama? **ling:** Burung pipit dan burung gereja. Saat ini semua jenis tersebut ada, padahal dahulu burung gereja jarang ditemukan di wilayah ini.

Arga: Seberapa besar estimasi kerugian yang ditimbulkan oleh hama burung ini terhadap hasil panen Bapak? **ling:** Persentase kerugian bervariasi. Jika serangannya parah, hasil panen bisa berkurang hingga 50 persen.

Arga: Metode apa yang saat ini Bapak gunakan untuk mengusir atau menanggulangi kawanan burung tersebut? **ling:** Kami menggunakan jaring penutup.

Arga: Apakah penggunaan jaring membutuhkan biaya operasional yang besar? **Iing:** Saya kurang mengetahui harga pastinya. Namun biayanya cukup tinggi karena jaring harus menutupi seluruh area lahan secara menyeluruh.

Arga: Apakah metode jaring tersebut efektif dalam mencegah serangan burung? **Iing:** Ya, cukup efektif.

Arga: Apakah terdapat akses jaringan listrik di area persawahan ini? **Iing:** Tidak ada akses listrik sama sekali.

Arga: Saya sedang mengembangkan sistem pengusir burung menggunakan kamera cerdas yang dapat mendeteksi kedatangan burung dan secara otomatis mengeluarkan suara peringatan. Bagaimana pendapat Bapak mengenai inovasi ini? **Iing:** Konsep tersebut bagus. Namun menurut n saya, akan lebih optimal jika alat tersebut dapat terbang seperti drone.

Arga: Konsep perangkat saya saat ini adalah perangkat statis yang dipasang pada tiang penyangga, dilengkapi modul kamera dengan kapabilitas rotasi. **Iing:** Saya kira alatnya bisa terbang seperti drone. Mobilitas drone akan sangat berguna karena kawanan burung pipit tidak menetap di satu titik, mereka berpindah-pindah area secara konstan.

Arga: Dengan sistem tiang, jangkauan penglihatan kamera memang harus diatur agar luas. Kamera akan dikonfigurasi untuk dapat bergerak memantau sekeliling seperti sistem pengawasan CCTV. Terima kasih atas informasi dan waktu Bapak, Bapak Iing. **Iing:** Sama-sama.

Lampiran Dokumentasi Pengujian UAT

