



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



**RANCANG BANGUN SISTEM PENGHITUNG BENIH IKAN
OTOMATIS BERBASIS IOT MENGGUNAKAN KECERDASAN
BUATAN**

“Perancangan Alat Penghitung Benih Ikan Otomatis Berbasis IoT”

TUGAS AKHIR

**POLITEKNIK
NEGERI
INNEKE NURUL FAUZIAH
2303332056**

**PROGRAM STUDI TELEKOMUNIKASI
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI JAKARTA
2026**



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta



RANCANG BANGUN SISTEM PENGHITUNG BENIH IKAN OTOMATIS BERBASIS IOT MENGGUNAKAN KECERDASAN BUATAN

“Perancangan Alat Penghitung Benih Ikan Otomatis Berbasis IoT”

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk memperoleh gelar Diploma
Tiga

INNEKE NURUL FAUZIAH
2303332056

PROGRAM STUDI TELEKOMUNIKASI
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI JAKARTA
2026



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

Nama : **Inneke Nurul Fauziah**

NIM : **2303332056**

Tanda Tangan : 

Tanggal : **7 Juli 2026**





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

**LEMBAR PENGESAHAN
TUGAS AKHIR**

Tugas Akhir diajukan oleh :

Nama : Inneke Nurul Fauziah

NIM : 2303332056

Program Studi : D3 - Telekomunikasi

Judul Tugas Akhir : Rancang Bangun Sistem Penghitung Benih Ikan Otomatis Berbasis IoT Menggunakan Kecerdasan Buatan

Telah diuji oleh tim penguji dalam Sidang Tugas Akhir pada 7 Juli 2026 dan dinyatakan **LULUS**.

Pembimbing : Shita Fitria Nurjihan, S.T., M.T.
NIP. 199206202019032028

()

Depok, 28 Juli 2026

Disahkan oleh



Dr. Murie Dwiyanti, S.T., M.T.
NIP. 197803312003122002



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

KATA PENGANTAR

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, penulis dapat menyelesaikan Tugas Akhir ini. Penulisan Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Politeknik. Penelitian ini merancang alat penghitung benih ikan otomatis berbasis IoT secara real-time untuk meningkatkan efisiensi dan akurasi dibandingkan metode manual.

Penulis menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan tugas akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan tugas akhir ini. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Shita Fitria Nurjihan, S.T., M.T., selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan penulis dalam penyusunan tugas akhir ini;
2. Orang tua dan keluarga penulis yang telah memberikan bantuan dukungan material dan moral;
3. Seluruh Bapak dan Ibu dosen Program Studi Telekomunikasi atas segala ilmu pengetahuan dan didikannya selama ini;
4. Fauzan Rayhan Fachrudin selaku rekan Tugas Akhir dan teman teman Telekomunikasi 6C 2023 yang memberikan bantuan dan semangat kepada penulis;
5. Sahabat yang telah banyak membantu penulis dalam menyelesaikan laporan tugas akhir ini.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga Tugas Akhir ini membawa manfaat bagi pengembangan ilmu.

Depok, 15 Juli 2026

Penulis

Inneke Nurul Fauziah



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

RANCANG BANGUN SISTEM PENGHITUNG BENIH IKAN OTOMATIS BERBASIS IOT MENGGUNAKAN KECERDASAN BUATAN

Perancangan Alat Penghitung Benih Ikan Otomatis Berbasis IoT

Abstrak

Budidaya ikan merupakan salah satu sektor perikanan yang memerlukan pengelolaan yang efektif dan efisien, terutama pada tahap penghitungan benih ikan sebelum proses distribusi maupun pembesaran. Penghitungan benih ikan yang masih dilakukan secara manual sering menimbulkan berbagai permasalahan, seperti waktu yang relatif lama, tingkat akurasi rendah, serta potensi ketidaksesuaian jumlah benih antara penjual dan pembeli. Penelitian ini bertujuan untuk merancang, merealisasikan, dan menguji tingkat akurasi sistem penghitung benih ikan otomatis berbasis Internet of Things (IoT) menggunakan webcam sebagai sensor pendeteksi objek, Raspberry Pi 4 sebagai pengendali utama, dan model YOLOv8 untuk mendeteksi serta menghitung benih ikan secara real-time. Sistem diterapkan dengan tiga metode penghitungan yang berjalan paralel, yaitu M1 Zone, M2 Single, dan M3 Dual, serta dilengkapi LCD 7 inci dan dashboard web untuk monitoring hasil secara real-time. Pengujian fungsional terhadap catu daya, modul relay, pompa air, dan LCD menunjukkan tegangan keluaran stabil pada kisaran 5,09–5,14V, serta modul relay dan pompa air berhasil merespons perintah dengan tingkat keberhasilan 100%. Pengujian akurasi yang dilakukan pada 20 variasi jumlah benih ikan menunjukkan metode M1 Zone memberikan akurasi rata-rata tertinggi sebesar 90,26% (46,15%–100%), sedangkan M2 Single dan M3 Dual cenderung mengalami undercount karena bergantung pada garis penghitungan. Kesalahan penghitungan yang ditemukan berupa undercount akibat benih ikan berdempetan dan overcount akibat noise citra maupun pergerakan objek. Dengan hasil tersebut, alat ini terbukti mampu meningkatkan efisiensi dan akurasi proses penghitungan benih ikan dibandingkan metode manual.

Kata kunci: benih ikan, Internet of Things (IoT), pengolahan citra, Raspberry Pi, webcam.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DESIGN AND DEVELOPMENT OF AN IOT-BASED AUTOMATIC FISH FRY COUNTING SYSTEM USING ARTIFICIAL INTELLIGENCE

“Perancangan Alat Penghitung Benih Ikan Otomatis Berbasis IoT”

Abstract

Fish farming is a fisheries sector that requires effective and efficient management, particularly in counting fish fry before distribution or grow-out activities. Manual counting still causes problems such as long processing time, low accuracy, and potential discrepancies in the number of fish fry between sellers and buyers. This study aims to design, implement, and test the accuracy of an IoT-based automatic fish fry counting system, using a webcam as the object detection sensor, a Raspberry Pi 4 as the main controller, and the YOLOv8 model to detect and count fish fry in real time. The system applies three parallel counting methods, M1 Zone, M2 Single, and M3 Dual, and is equipped with a 7-inch LCD and a web dashboard for real-time monitoring. Functional testing of the power supply, relay module, water pump, and LCD showed a stable output of 5.09V–5.14V, while the relay module and water pump responded to commands with a 100% success rate. Accuracy testing carried out on 20 variations in the number of fish fry showed that M1 Zone achieved the highest average accuracy of 90.26% (ranging from 46.15% to 100%), while M2 Single and M3 Dual tended to undercount due to their dependence on counting lines. Errors found were undercounts from overlapping fish fry and overcounts from image noise or object movement. The device has proven capable of improving the efficiency and accuracy of fish fry counting compared to the manual method.

Keywords: digital image processing, fish seed, Internet of Things (IoT), Raspberry Pi, webc



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR ISI

HALAMAN SAMPUL.....	i
HALAMAN JUDUL	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
LEMBAR PENGESAHAN	iv
KATA PENGANTAR.....	v
Abstrak.....	vi
Abstract.....	vii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Perumusan Masalah.....	1
1.3 Tujuan.....	2
1.4 Luaran.....	2
BAB II TINJAUAN PUSTAKA.....	3
2.1 Penghitungan Benih Ikan.....	3
2.2 Internet of Things (IoT).....	3
2.3 Raspberry Pi 4	4
2.4 Webcam	5
2.5 LCD 7 Inchi.....	5
2.6 Power Supply.....	6
2.7 Pompa Air.....	6
2.8 Stepdown	7
2.9 Modul Relay 1 Channel 5V.....	7
2.10 Persamaan Akurasi Benih Ikan.....	8
2.11 Persamaan Rata-rata Akurasi Benih Ikan.....	8
BAB III PERENCANAAN DAN REALISASI.....	9
3.1 Rancangan Alat.....	9
3.1.1 Deskripsi Alat	9
3.1.2 Cara Kerja Alat.....	10
3.1.3 Spesifikasi Alat.....	12
3.1.4 Diagram Blok.....	13
3.1.5 Rancangan Raspberry Pi 4 dan Catu Daya	14
3.1.6 Rancangan Raspberry Pi 4 dan Webcam.....	15
3.1.7 Rancangan Raspberry Pi 4 dan LCD 7inci	16
3.1.8 Rancangan Raspberry Pi 4 dan Modul Relay 1 Channel.....	17
3.2 Realisasi Alat.....	19
3.2.1 Realisasi Alat Penghitung Benih Ikan Otomatis.....	19



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

3.2.2	Realisasi Catu Daya	20
3.2.3	Realisasi Sistem Akuisisi Citra	21
3.2.4	Realisasi Sistem Pengolahan Data dan Program Raspberry Pi 4	21
3.2.5	Realisasi Proses Deteksi dan Penghitungan Benih Ikan	23
3.2.6	Realisasi Sistem Monitoring	28
3.2.7	Realisasi Sistem Keseluruhan	29
BAB IV PEMBAHASAN		26
4.1	Pengujian Catu Daya	26
4.1.1	Deskripsi Alat	26
4.1.2	Tujuan Pengujian	26
4.1.3	Set-Up Pengujian Alat	27
4.1.4	Prosedur Pengujian	27
4.1.5	Data Hasil Pengujian	27
4.1.6	Analisa Pengujian	29
4.2	Pengujian Sistem Pengendali	30
4.2.1	Deskripsi Alat	30
4.2.2	Tujuan Pengujian	30
4.2.3	Set-Up Pengujian Alat	30
4.2.4	Prosedur Pengujian	31
4.2.5	Data Hasil Pengujian	31
4.2.6	Analisa Pengujian	33
4.3	Pengujian Sistem Tampilan	34
4.3.1	Deskripsi Alat	34
4.3.2	Tujuan Pengujian	34
4.3.3	Set-Up Pengujian Alat	34
4.3.4	Prosedur Pengujian	35
4.3.5	Data Hasil Pengujian	35
4.3.6	Analisa Pengujian	36
4.4	Pengujian Akurasi Perhitungan Benih Ikan	36
4.4.1	Deskripsi Pengujian	36
4.4.2	Tujuan Pengujian	37
4.4.3	Set-Up Pengujian Alat	37
4.4.4	Prosedur Pengujian	38
4.4.5	Data Hasil Pengujian	38
4.4.6	Analisa Data	45
4.5	Analisa keseluruhan	46
BAB V PENUTUP		47



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

5.1	Kesimpulan.....	47
5.2	Saran.....	47
DAFTAR PUSTAKA		48
DAFTAR RIWAYAT HIDUP		49
LAMPIRAN.....		50





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR TABEL

Tabel 3. 1 Spesifikasi Komponen Alat Penghitung Benih Ikan.....	12
Tabel 3. 2 Koneksi Pin Raspberry pi 4 dan Catu Daya.....	15
Tabel 3. 3 Koneksi Pin Raspberry dan Webcam.....	16
Tabel 3. 4 Koneksi Pin Raspberry dan LCD Display	17
Tabel 3. 5 Koneksi Pin Raspberry dan Modul Relay.....	18
Tabel 3. 6 Koneksi Modul Relay dan Pompa Air DC 12V	19
Tabel 4. 1 Data Hasil Pengujian Tegangan Catu Daya	28
Tabel 4. 2 Hasil Pengujian Kestabilan Tegangan Step-Down (Kondisi Beban).....	28
Tabel 4. 3 Data Hasil Pengujian Modul Relay.....	31
Tabel 4. 4 Hasil Pengujian Switching Relay.....	32
Tabel 4. 5 Data Hasil Pengujian Pompa Air	32
Tabel 4. 6 Data Hasil Pengujian Modul Relay.....	35
Tabel 4. 7 Hasil Pengujian Akurasi Penghitungan Benih Ikan.....	39





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR GAMBAR

Gambar 2. 1 Raspberry Pi 4.....	4
Gambar 2. 2 Webcam Logitech C270.....	5
Gambar 2. 3 Power Supply 20A 12V.....	5
Gambar 2. 4 Power Supply 12V 10A.....	6
Gambar 2. 5 Pompa Air DC 12V	6
Gambar 2. 6 Pompa Air DC 12V	7
Gambar 2. 7 Modul Relay 1 Channel 5V.....	7
Gambar 3. 1 Ilustrasi Alat dan Tampilan Penghitung Bibit Ikan Otomatis	10
Gambar 3. 2 Flowchart Cara Kerja	11
Gambar 3. 3 Diagram Blok Penghitung Bibit Ikan Otomatis	13
Gambar 3. 4 Rangkaian Skematik Hubungan Raspberry Pi 4 dan Power Supply	14
Gambar 3. 5 Rangkaian Skematik Hubungan Raspberry Pi 4 dan Webcam	15
Gambar 3. 6 Rangkaian Skematik Hubungan Raspberry Pi 4 dan LCD 7inci	16
Gambar 3. 7 Rangkaian Skematik Hubungan Raspberry Pi 4 dan Modul Relay	18
Gambar 3. 8 Rangkaian Skematik Alat Penghitung Benih Ikan Otomatis	20
Gambar 3. 9 Realisasi Catu Daya	20
Gambar 3. 10 Webcam sebagai Sistem Akuisisi Citra	21
Gambar 3. 11 Raspberry Pi 4 sebagai Sistem Pengolahan Data	22
Gambar 3. 12 Area deteksi dan penghitungan	24
Gambar 3. 13 Contoh Bounding Box.....	24
Gambar 3. 14 Contoh L1, L2 dan L3.....	26
Gambar 3. 15 Contoh hasil deteksi ketiga metode.....	27
Gambar 3. 16 Contoh hasil deteksi kekurangan hitung	28
Gambar 3. 17 LCD 7inci sebagai Sistem Monitoring.....	28
Gambar 3. 18 Realisasi Sistem Keseluruhan	29
Gambar 4. 1 Set-Up Pengujian Catu Daya.....	27
Gambar 4. 2 Pengukuran Tegangan Catu Daya	28
Gambar 4. 3 Pengukuran Tegangan Keluaran Step-down Percobaan Ke-3	29
Gambar 4. 4 Set-up Pengujian Sistem Pengendali.....	30
Gambar 4. 5 Pengukuran Tegangan Sinyal Trigger Modul Relay.....	32
Gambar 4. 6 Arus Pompa Air.....	33
Gambar 4. 7 Tampilan Perhitungan pada LCD 7Inch.....	36
Gambar 4. 8 Tampilan awal sebelum memulai penghitungan	37
Gambar 4. 9 Set-Up Pengujian Akurasi Penghitung Benih Ikan	38
Gambar 4. 10 Contoh Tampilan Deteksi Benih	38
Gambar 4. 11 Kondisi Benih Ikan Berdempetan pada Percobaan ke-1	42
Gambar 4. 12 Kondisi Kelebihan Hitung Benih Ikan pada Percobaanke- 4.....	43
Gambar 4. 13 Kondisi Kelebihan Hitung Benih Ikan pada Percobaanke- 10.....	43
Gambar 4. 14 Kondisi Sempurna Hitung Benih Ikan pada Percobaanke-20.....	44



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR LAMPIRAN

L - 1 Realisasi Alat.....	50
L - 2 Realisasi Sistem Penghitung Benih Ikan.....	51
L - 3 Skematik Aat	52
L - 4 Tampilan LCD 7inci.....	53
L - 5 Sketch Pendeteksi Ikan.....	54





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi *Internet of Things* (IoT) telah memberikan banyak manfaat dalam berbagai bidang, termasuk pada sektor perikanan. Salah satu kegiatan penting dalam budidaya ikan adalah proses perhitungan benih ikan sebelum dilakukan distribusi atau pembesaran. Namun, proses perhitungan benih ikan masih sering dilakukan secara manual, sehingga membutuhkan waktu yang cukup lama dan berpotensi menimbulkan kesalahan, terutama ketika jumlah benih yang dihitung sangat banyak.

Kesalahan dalam perhitungan benih ikan dapat menyebabkan ketidaksesuaian jumlah antara penjual dan pembeli yang berpotensi menimbulkan kerugian. Selain itu, metode manual juga dinilai kurang efisien karena membutuhkan tenaga kerja lebih banyak dan tingkat akurasi yang kurang optimal. Oleh karena itu, diperlukan suatu sistem yang dapat membantu proses perhitungan benih ikan secara otomatis, cepat, dan akurat.

Pemanfaatan teknologi kamera digital, pengolahan citra, dan IoT memungkinkan proses perhitungan benih ikan dilakukan secara otomatis. Sistem dapat mendeteksi benih ikan yang melewati area pengamatan, kemudian menghitung jumlahnya dan menampilkan hasil perhitungan secara real-time layar LCD sehingga memudahkan pengguna dalam melakukan pemantauan.

Berdasarkan hal tersebut, pada Tugas Akhir ini dirancang alat penghitung benih ikan otomatis berbasis IoT menggunakan webcam dan Raspberry Pi. Sistem ini diharapkan dapat membantu pembudidaya ikan meningkatkan efisiensi kerja, mengurangi kesalahan perhitungan, serta memudahkan proses monitoring hasil perhitungan secara real-time.

1.2 Perumusan Masalah

Rumusan Masalah yang akan dibahas pada tugas akhir ini adalah:

- Bagaimana merancang sistem penghitung benih ikan secara otomatis?
- Bagaimana merealisasikan sistem penghitung benih ikan secara otomatis?
- Bagaimana menguji tingkat akurasi perangkat keras sistem penghitung benih ikan secara otomatis?



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

1.3 Tujuan

Tujuan yang ingin dicapai dalam pembuatan tugas akhir ini adalah:

- a. Merancang sistem penghitung benih ikan secara otomatis.
- b. Merealisasikan sistem penghitung benih ikan secara otomatis.
- c. Menguji tingkat akurasi perangkat keras pada sistem penghitung benih ikan secara otomatis.

1.4 Luaran

Luaran yang diharapkan dari pelaksanaan Tugas Akhir ini meliputi:

- a. Sistem penghitung benih ikan otomatis berbasis IoT yang dapat bekerja secara real-time.
- b. Laporan Tugas Akhir.
- c. Artikel Ilmiah





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan, realisasi, dan pengujian alat yang telah dilakukan, maka dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem penghitung benih ikan otomatis berbasis IoT berhasil dirancang dengan mengintegrasikan Raspberry Pi 4 sebagai pengendali utama, webcam sebagai sensor pendeteksi objek, metode pengolahan citra digital untuk deteksi dan penghitungan objek, modul relay dan pompa air sebagai aktuator pengendali aliran benih ikan, LCD 7 inci sebagai media tampilan lokal, serta dashboard web sebagai sarana monitoring hasil secara real-time.
2. Sistem berhasil direalisasikan dan bekerja dengan baik berdasarkan hasil pengujian perangkat keras. Catu daya menghasilkan tegangan keluaran 12,02V dan modul step-down menghasilkan tegangan stabil pada kisaran 5,09–5,14V sehingga Raspberry Pi 4 dapat beroperasi secara normal. Selain itu, modul relay, pompa air, dan LCD 7 inci juga menunjukkan tingkat keberhasilan fungsi sebesar 100%, sehingga mendukung proses penghitungan benih ikan secara otomatis.
3. Pengujian akurasi pada 20 variasi jumlah benih ikan (5–35 ekor) menunjukkan metode M1 Zone paling mendekati jumlah aktual dengan rata-rata akurasi 90,26% (tertinggi 100%, terendah 46,15%), lebih unggul dibandingkan M2 Single dan M3 Dual yang cenderung menghasilkan jumlah hitungan lebih rendah karena bergantung pada garis referensi. Kesalahan yang masih ditemukan berupa kekurangan hitung akibat benih ikan berdempetan dan kelebihan hitung akibat noise citra maupun perubahan kontur objek saat bergerak.

5.2 Saran

1. Sistem dapat dikembangkan dengan menggunakan kamera beresolusi lebih tinggi serta dilengkapi pengaturan pencahayaan tambahan (misalnya LED ring light), agar kualitas citra yang ditangkap lebih jelas dan proses deteksi objek menjadi lebih stabil.



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

2. Sistem diharapkan dapat diuji pada jumlah benih ikan yang lebih banyak serta pada kondisi budidaya yang berbeda agar performa dan keandalan sistem dapat diketahui secara lebih menyeluruh.





Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR PUSTAKA

- Saputra, D., Putra, R.E. dan Ramadhan, F. (2021). *Penerapan deep learning berbasisTensorFlow Lite pada perangkat embedded untuk deteksi objek*. Jurnal Informatika dan Sistem Informasi, 7(2), pp. 89–96.
- Rahman, A., Yuliani, S. dan Pratama, I.G. (2022). *Sistem deteksi objek real-time berbasis Raspberry Pi dan kamera menggunakan metode convolutional neural network*. Jurnal Ilmiah Teknik Elektro, 14(1), pp. 25–32.
- Putri, N.A. (2020). *Rancang bangun sistem deteksi objek berbasis Raspberry Pi menggunakan metode deep learning*. Tugas Akhir. Politeknik Negeri Bandung.
- Dendi, A. S., & Sunardi. (2021). *Alat penghitung benih ikan lele menggunakan pengolahan citra*. Diploma thesis, Politeknik Manufaktur Negeri Bangka Belitung.
- Hindarto, D. (2023). Exploring YOLOv8 pretrain for realtime detection of Indonesia native fish species. *Jurnal & Penelitian Teknik Informatika (Sinkron)*, 8(4).
- Mahardi, R. D., Sunuharjo, L., Hendrawan, D., 'Atiq, M., Wahyuadi, R. A., & Nugraha, S. P. A. (2024). Desain perancangan buck converter berbasis IC LM2596. *Jurnal Sains dan Ilmu Terapan*, 7(2), 185–192.
- Suharman, & Murti, H. (2018). Sistem monitoring budidaya ikan lele berbasis Internet of Things menggunakan Raspberry Pi. *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIK)*, 5(6).
- Wardana, M. W., Maskuri, I., & Zaim, M. S. (2023). Pengolahan citra digital pada perhitungan ikan hias menggunakan metode Blob. *Jurnal Riset Kajian Teknologi dan Lingkungan (JRKTL)*, 6(2), 118–126.



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

DAFTAR RIWAYAT HIDUP



Inneke Nurul Fauziah

Lahir di Jakarta, 9 Februari 2003. Lulus dari SDN Serdang 11 Pagi 2015. Lalu melanjutkan Pendidikan di SMP Negeri 79 Jakarta dan lulus tahun 2018. Lalu melanjutkan ke jenjang menengah di SMKS TAMANSISWA 1 JAKARTA dan lulus tahun 2021. Lalu melanjutkan pendidikan perguruan tinggi di Politeknik Negeri Jakarta pada Program Studi Jurusan Teknik Telekomunikasi, Elektro dan memperoleh Gelar Diploma Tiga (D3) tahun 2026.

**POLITEKNIK
NEGERI
JAKARTA**



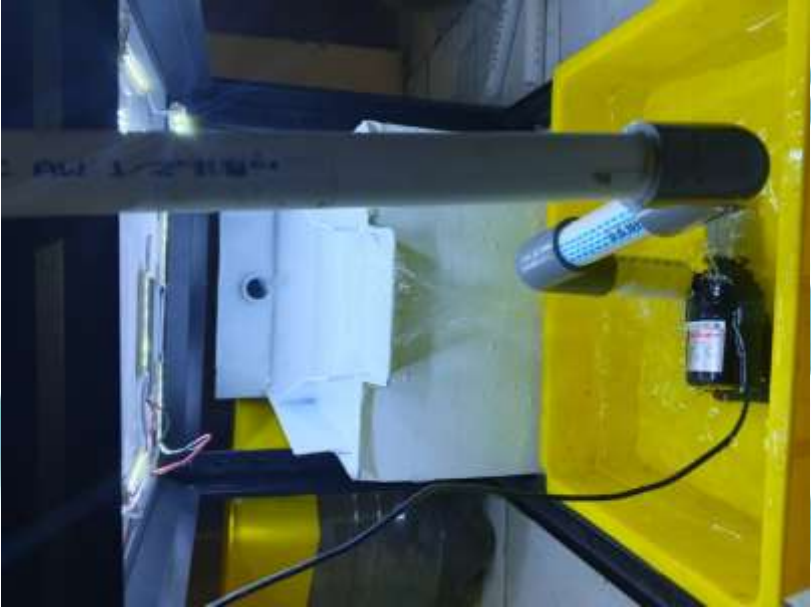
© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

LAMPIRAN

L - 1 Realisasi Alat

		REALISASI ALAT	
		L1	
		POLITEKNIK NEGERI JAKARTA	
Inneke Nurul Fauzan		TT6C	
		28 Juli 2026	



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

L - 2 Realisasi Sistem Penghitung Benih Ikan

L2	 POLITEKNIK NEGERI JAKARTA	
REALISASI SISTEM PENGHITUNG BENIH	Inneke Nurul Fauzan	
TT6C	28 Juli 2026	



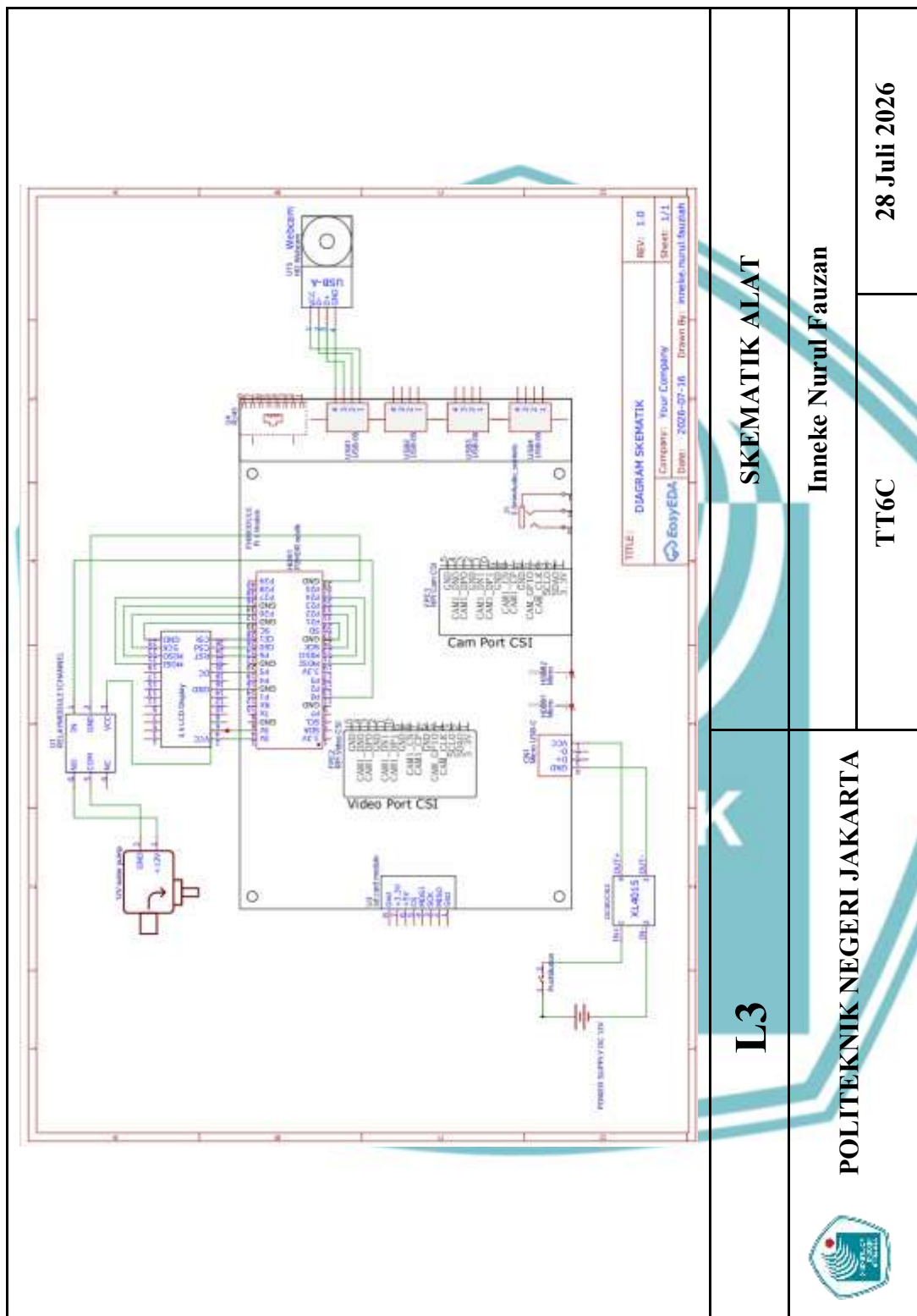


© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

L - 3 Skematik Aat





© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

- Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
- Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

Fish Counter AI — ONNX Runtime FISH COUNTER AI Engine: ONNX Runtime FPS: 37 Avg FPS: 40.73 Durasi: 00:00 M1 Zone: 0 M2 Single: 0 M3 Dual: 0 Status: RUNNING Pompa: MENYALA Nama Pembeli: Inne Jumlah Aktual: 20 START RESET IDE MONITOR (Esc keli EXIT		TAMPILAN LCD 7INCI	
L5 POLITEKNIK NEGERI JAKARTA	Inneke Nurul Fauzan		
	TT6C		
28 Juli 2026			

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
import cv2
import numpy as np
import time
import threading
import tkinter as tk
from PIL import Image, ImageTk
from supabase import create_client, Client
from datetime import datetime

# ===== ONNX RUNTIME (Saran Eka Puji)
# =====
# Install: pip install onnxruntime
# Untuk Raspberry Pi: pip install onnxruntime (versi ARM)
try:
    import onnxruntime as ort

    sess_options = ort.SessionOptions()
    sess_options.intra_op_num_threads = 4 # pakai semua core Pi 4
    # PENTING: inter_op_num_threads HARUS 1 kalau intra_op sudah
    pakai semua core.
    # Kombinasi 4+2 = bisa minta sampai 8 thread rebutan cuma 4
    core fisik,
    # bikin overhead context-switching dan malah MEMPERLAMBAT
    inference.
    sess_options.inter_op_num_threads = 1
    sess_options.graph_optimization_level =
ort.GraphOptimizationLevel.ORT_ENABLE_ALL
    sess_options.enable_mem_pattern = True
    sess_options.enable_cpu_mem_arena = True
    sess_options.execution_mode = (
        ort.ExecutionMode.ORT_SEQUENTIAL
    ) # lebih stabil utk model kecil single-stream
    session = ort.InferenceSession(
        "best320.onnx", sess_options,
        providers=["CPUExecutionProvider"]
    )
    input_name = session.get_inputs()[0].name
    USE_ORT = True
    print("✅ ONNX Runtime aktif")
except ImportError:
    net = cv2.dnn.readNetFromONNX("best320.onnx")
    USE_ORT = False
    print("⚠️ Fallback ke cv2.dnn - install onnxruntime untuk
    performa lebih baik")

# ===== SUPABASE =====
SUPABASE_URL = "https://hndejxzmoodipqmmrto.supabase.co"
SUPABASE_KEY =
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJzdXBhYmFzZSIsInJl
ZiI6ImhuZGVqehptbW9vZGlwcWltnRvIiwicm9sZSI6ImFub24iLCJpYXQiOiE3Nz
E2ODQ2MDYsImV4cCI6MjA4NDYyMjA4NDYyMjA4NDYyMjA4NDYyMjA4NDYyMjA4NDYy
KyJQZQMjVlAo"
supabase: Client = create_client(SUPABASE_URL, SUPABASE_KEY)

# ===== KONTROL POMPA AIR (GPIO) =====
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Status on/off dikontrol dari Admin Panel (cloud) lewat tabel
device_control
# di Supabase. Pi cek status ini secara berkala lalu
menyalakan/mematikan
# relay pompa. Try/except supaya tetap bisa dites di laptop tanpa
GPIO.
PIN_RELAY_POMPA = 17 # GANTI sesuai pin relay yang dipakai
POMPA_POLL_INTERVAL_MS = 3000 # cek status tiap 3 detik
pompa_status_terakhir = None

try:
    import RPi.GPIO as GPIO

    GPIO.setmode(GPIO.BCM)
    GPIO.setup(PIN_RELAY_POMPA, GPIO.OUT)
    GPIO.output(PIN_RELAY_POMPA, GPIO.LOW)
    GPIO_AVAILABLE = True
    print("✅ GPIO relay pompa siap (pin", PIN_RELAY_POMPA, ")")
except ImportError:
    GPIO_AVAILABLE = False
    print("⚠️ RPi.GPIO tidak ditemukan - mode simulasi (tidak
menyalakan relay fisik)")

# ===== CAMERA / VIDEO SOURCE =====
# Untuk testing pakai file video: isi VIDEO_SOURCE dengan path
filenya.
# Untuk kembali ke webcam nanti: kosongkan lagi (None) atau
comment baris ini.
CAM_INDEX = 1
VIDEO_SOURCE = None # None -> pakai webcam

IS_VIDEO_FILE = bool(VIDEO_SOURCE)
cap = cv2.VideoCapture(VIDEO_SOURCE if IS_VIDEO_FILE else
CAM_INDEX)

if not IS_VIDEO_FILE:
    # Setting ini cuma relevan untuk webcam fisik, tidak
berpengaruh ke file video
    cap.set(
        cv2.CAP_PROP_FOURCC, cv2.VideoWriter_fourcc(*"MJPG")
    ) # jauh lebih cepat di Pi dibanding YUYV
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
    cap.set(cv2.CAP_PROP_FPS, 30)
    cap.set(
        cv2.CAP_PROP_BUFFERSIZE, 1
    ) # jangan numpuk frame lama, selalu ambil frame terbaru

if IS_VIDEO_FILE and not cap.isOpened():
    print(f"⚠️ Gagal membuka video: {VIDEO_SOURCE} - cek lagi
path-nya.")

# ===== INPUT SIZE =====
# Sesuaikan dengan model yang diexport:
# best320.onnx → 320 | best160.onnx → 160 | best96.onnx → 96
# Saran Eka Puji: export dengan imgsz=160,160 untuk Raspberry Pi
INPUT_SIZE = 320
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang menggunakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# ===== LETTERBOX (Saran Eka Puji) =====
def letterbox(img, new_shape=320):
    """
    Resize dengan mempertahankan aspect ratio – standar YOLO.
    Lebih akurat dari resize biasa karena tidak distorsi.
    Area kosong diisi abu-abu (114,114,114).
    """
    h, w = img.shape[:2]
    r = min(new_shape / h, new_shape / w)
    nw, nh = int(w * r), int(h * r)
    img = cv2.resize(img, (nw, nh),
        interpolation=cv2.INTER_LINEAR)

    dw = (new_shape - nw) / 2
    dh = (new_shape - nh) / 2
    top, bottom = int(round(dh - 0.1)), int(round(dh + 0.1))
    left, right = int(round(dw - 0.1)), int(round(dw + 0.1))
    img = cv2.copyMakeBorder(
        img, top, bottom, left, right, cv2.BORDER_CONSTANT,
        value=(114, 114, 114)
    )
    return img, r, (dw, dh)

# ===== INFERENCE (Saran Eka Puji) =====
def run_inference(image):
    """
    Inferensi dengan ONNX Runtime atau fallback cv2.dnn.
    ONNX Runtime pakai letterbox + np.ascontiguousarray
    untuk memory layout optimal – lebih cepat dari blobFromImage.
    """
    sz = INPUT_SIZE
    if USE_ORT:
        img_lb, ratio, (dw, dh) = letterbox(image, sz)
        img_in = cv2.cvtColor(img_lb, cv2.COLOR_BGR2RGB)
        img_in = img_in.astype(np.float32) / 255.0
        img_in = np.ascontiguousarray(img_in.transpose(2, 0,
1)) [np.newaxis]
        out = session.run(None, {input_name:
img_in}) [0] [0].transpose()
        return out, ratio, dw, dh
    else:
        blob = cv2.dnn.blobFromImage(
            image, 1 / 255.0, (sz, sz), swapRB=True, crop=False
        )
        net.setInput(blob)
        out = net.forward() [0].transpose()
        h, w = image.shape[:2]
        ratio = min(sz / h, sz / w)
        dw = (sz - w * ratio) / 2
        dh = (sz - h * ratio) / 2
        return out, ratio, dw, dh

# ===== SORT TRACKER =====
class KalmanTrack:
    count = 1

    def __init__(self, bbox, obj_id=None):
        self.kf = cv2.KalmanFilter(4, 2)
```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

self.kf.measurementMatrix = np.array([[1, 0, 0, 0], [0, 1,
0, 0]], np.float32)
self.kf.transitionMatrix = np.array(
    [[1, 0, 1, 0], [0, 1, 0, 1], [0, 0, 1, 0], [0, 0, 0,
1]], np.float32
)
self.kf.processNoiseCov = (
    np.eye(4, dtype=np.float32) * 0.03
) # diturunkan agar prediksi lebih smooth
self.kf.measurementNoiseCov = (
    np.eye(2, dtype=np.float32) * 0.1
) # diturunkan agar koreksi lebih responsif
self.kf.errorCovPost = np.eye(4, dtype=np.float32)

cx, cy = self._bbox_to_center(bbox)
self.kf.statePost = np.array([[cx], [cy], [0], [0]],
np.float32)

self.id = obj_id if obj_id else KalmanTrack.count
if obj_id is None:
    KalmanTrack.count += 1

self.bbox = bbox
self.age = 0
self.hits = 1
self.time_since_update = 0

self.passed_line1 = False
self.counted_dual = False
self.counted_single = False
self.counted_zone = False
self.prev_cy = cy

# ===== State hysteresis (anti-jitter) untuk deteksi
lintas garis =====
# 'atas' / 'bawah' / None(belum diketahui) - dipakai
supaya centroid
# yang goyang (jitter) tepat di garis TIDAK dihitung
berkali-kali,
# dan supaya lintasan yang lambat/erratic tetap TERHITUNG
(tidak
# kelewat) karena harus jelas-jelas berada di sisi yang
berlawanan
# dulu sebelum status berpindah.
self.side_l1 = None
self.side_l2 = None
self.zone_state = None # 'luar_atas' / 'dalam' /
'luar_bawah'

def _bbox_to_center(self, bbox):
    x1, y1, x2, y2 = bbox
    return (x1 + x2) / 2.0, (y1 + y2) / 2.0

def predict(self):
    self.kf.predict()
    self.age += 1
    self.time_since_update += 1

def update(self, bbox):

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        cx, cy = self._bbox_to_center(bbox)
        self.kf.correct(np.array([[np.float32(cx)],
[ np.float32(cy) ]]))
        self.bbox = bbox
        self.hits += 1
        self.time_since_update = 0

    def get_state(self):
        st = self.kf.statePost
        cx, cy = float(st[0][0]), float(st[1][0])
        x1, y1, x2, y2 = self.bbox
        bw = (x2 - x1) / 2
        bh = (y2 - y1) / 2
        return (int(cx - bw), int(cy - bh), int(cx + bw), int(cy +
bh))

    def get_center(self):
        st = self.kf.statePost
        return int(float(st[0][0])), int(float(st[1][0]))

class SORTTracker:
    def __init__(self, max_age=25, min_hits=1,
iou_threshold=0.12):
        self.max_age = max_age
        self.min_hits = min_hits
        self.iou_threshold = iou_threshold
        self.tracks = []

    def reset(self):
        self.tracks = []
        KalmanTrack.count = 1

    @staticmethod
    def _iou(boxA, boxB):
        ax1, ay1, ax2, ay2 = boxA
        bx1, by1, bx2, by2 = boxB
        ix1 = max(ax1, bx1)
        iy1 = max(ay1, by1)
        ix2 = min(ax2, bx2)
        iy2 = min(ay2, by2)
        inter = max(0, ix2 - ix1) * max(0, iy2 - iy1)
        if inter == 0:
            return 0.0
        aA = (ax2 - ax1) * (ay2 - ay1)
        aB = (bx2 - bx1) * (by2 - by1)
        return inter / float(aA + aB - inter)

    def _hungarian_match(self, detections):
        if not self.tracks or not detections:
            return [], list(range(len(detections))),
list(range(len(self.tracks)))

        cost = np.zeros((len(self.tracks), len(detections)),
dtype=np.float32)
        for t, track in enumerate(self.tracks):
            tb = track.get_state()
            for d, det in enumerate(detections):
                cost[t, d] = -self._iou(tb, det[:4])

```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

try:
    from scipy.optimize import linear_sum_assignment

    row_ind, col_ind = linear_sum_assignment(cost)
except ImportError:
    # Greedy fallback jika scipy tidak ada
    matched, unm_d, unm_t = [],
    list(range(len(detections))), []
    for t in range(len(self.tracks)):
        best_iou, best_d = 0, -1
        for d in range(len(detections)):
            if d in [m[1] for m in matched]:
                continue
            iou = -cost[t, d]
            if iou > best_iou:
                best_iou, best_d = iou, d
        if best_d >= 0 and best_iou >= self.iou_threshold:
            matched.append((t, best_d))
            unm_d.remove(best_d)
        else:
            unm_t.append(t)
    return matched, unm_d, unm_t

matched, unmatched_dets, unmatched_trks = [], [], []
for d in range(len(detections)):
    if d not in col_ind:
        unmatched_dets.append(d)
for t in range(len(self.tracks)):
    if t not in row_ind:
        unmatched_trks.append(t)
for r, c in zip(row_ind, col_ind):
    if -cost[r, c] < self.iou_threshold:
        unmatched_dets.append(c)
        unmatched_trks.append(r)
    else:
        matched.append((r, c))
return matched, unmatched_dets, unmatched_trks

def update(self, detections):
    for track in self.tracks:
        track.predict()
    matched, unmatched_dets, _ =
self._hungarian_match(detections)
    for t_idx, d_idx in matched:
        self.tracks[t_idx].update(detections[d_idx][:4])
    for d_idx in unmatched_dets:
        self.tracks.append(KalmanTrack(detections[d_idx][:4]))
    self.tracks = [t for t in self.tracks if
t.time_since_update <= self.max_age]
    return [t for t in self.tracks if t.hits >= self.min_hits]

# ===== INISIALISASI TRACKER =====
tracker = SORTTracker(
    max_age=25, min_hits=1, iou_threshold=0.12
) # min_hits diturunkan ke 1 (seperti fishcountingONNX.py) supaya
track langsung
# aktif sejak deteksi pertama – ikan yang cuma sekilas
lewat/motion blur tetap

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# kehitung. Logika counting M1/M2/M3 (hysteresis anti-jitter)
TIDAK diubah.

# ===== STATE =====
running = False
session_id = (
    None # id baris di Supabase untuk sesi yang sedang berjalan
    (realtime save)
)
last_synced_counts = (
    0,
    0,
    0,
) # (count_zone, count_single, count_dual) terakhir dikirim

count_zone = 0
count_single = 0
count_dual = 0

frame_count = 0

MIN_AREA = 300 # diturunkan dari 500 agar ikan yang lebih kecil
tetap terdeteksi

prev_time = 0
fps_total = 0
fps_samples = 0
avg_fps = 0

durasi_display = 0
confidence_total = 0
confidence_count = 0
avg_confidence = 0

gui_render_counter = (
    0 # untuk selang-seling update gambar GUI (hemat CPU, deteksi
    tetap tiap frame)
)

start_time = None
waktu_mulai = None

line1_y = 190
line2_y = 230
line3_y = 270

zone_top = 188
zone_bottom = 232

# Margin hysteresis (px) - centroid harus jelas-jelas di satu sisi
(di luar
# margin ini) sebelum status "atas/bawah" dianggap berubah. Ini
yang bikin
# hitungan STABIL: jitter kecil di sekitar garis tidak dihitung
berkali-kali,
# tapi lintasan asli (meski gerak ikan lambat/erratic) tetap
tertangkap.
LINE_MARGIN = 3 # diturunkan dari 5 agar deteksi lintas garis
lebih sensitif
```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# ===== MODE MONITOR (ringan, tanpa browser)
=====
# Menampilkan ulang frame yang SUDAH ada di memori (img_tk), tidak
ada
# encode/decode tambahan sama sekali – aman untuk performa
Raspberry Pi.
monitor_window = None
monitor_label_widget = None

def buka_mode_monitor():
    global monitor_window, monitor_label_widget

    if monitor_window is not None:
        return # sudah terbuka, jangan buka doble

    monitor_window = tk.Toplevel(root)
    monitor_window.attributes("-fullscreen", True)
    monitor_window.configure(bg="black")
    monitor_window.config(cursor="none") # sembunyikan kursor di
    layar publik
    monitor_window.update_idletasks() # paksa hitung ukuran layar
    segera (penting untuk LCD 7")

    monitor_label_widget = tk.Label(monitor_window, bg="black")
    monitor_label_widget.pack(expand=True, fill="both")

    def tutup_monitor(event=None):
        global monitor_window, monitor_label_widget
        if monitor_window is not None:
            monitor_window.destroy()
            monitor_window = None
            monitor_label_widget = None

    # Tombol ESC di layar (selain shortcut keyboard Escape) –
    supaya mode
    # monitor bisa ditutup lewat sentuh/klik langsung di layar
    publik
    # (LCD 7" dsb yang biasanya tidak dipasang keyboard).
    btn_esc = tk.Button(
        monitor_window,
        text="X ESC",
        font=("Arial", 12, "bold"),
        bg="#ef4444",
        fg="white",
        bd=0,
        activebackground="#dc2626",
        activeforeground="white",
        command=tutup_monitor,
    )
    btn_esc.place(relx=1.0, rely=0.0, anchor="ne", x=-12, y=12)

    monitor_window.bind("<Escape>", tutup_monitor)
    monitor_window.protocol("WM_DELETE_WINDOW", tutup_monitor)

def sinkron_ke_monitor(img_pil):
    if monitor_label_widget is None:
        return
```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

win_w = monitor_window.winfo_width()
win_h = monitor_window.winfo_height()
if win_w <= 1 or win_h <= 1:
    return # ukuran window belum siap, lewati frame ini

# Perbesar mengikuti ukuran layar (LCD 7" dsb) sambil
mempertahankan
# aspect ratio asli supaya gambar tidak gepeng/distorsi.
src_w, src_h = img_pil.size
scale = min(win_w / src_w, win_h / src_h)
new_w, new_h = max(1, int(src_w * scale)), max(1, int(src_h *
scale))
resized = img_pil.resize(
    (new_w, new_h), Image.NEAREST
) # NEAREST = paling ringan di Raspberry Pi

# Tempel di tengah dengan latar hitam (letterbox) mengisi
penuh layar
canvas = Image.new("RGB", (win_w, win_h), (0, 0, 0))
canvas.paste(resized, ((win_w - new_w) // 2, (win_h - new_h)
// 2))

monitor_img_tk = ImageTk.PhotoImage(image=canvas)
monitor_label_widget.imgtk = (
    monitor_img_tk # simpan reference, wajib agar tidak
hilang dari memori
)
monitor_label_widget.configure(image=monitor_img_tk)

HARGA_PER_EKOR_DEFAULT = 1000
harga_per_ekor_aktif = HARGA_PER_EKOR_DEFAULT

def ambil_harga_per_ekor():
    """Ambil harga per ekor TERKINI dari Supabase (diisi lewat
Admin Panel),
    supaya kalau admin ubah harga karena mengikuti harga pasar,
sesi
    berikutnya otomatis pakai harga baru tanpa perlu update
aplikasi ini.

    Asumsi nama tabel/kolom: tabel "pengaturan", baris id=1, kolom
    "harga_per_ekor" (mengikuti pola tabel "device_control" id=1
yang sudah
    dipakai untuk kontrol pompa di file ini). KALAU nama
tabel/kolom di
    Supabase kamu berbeda, sesuaikan baris di bawah ini -
dibungkus
    try/except supaya kalau gagal/tidak ketemu, aplikasi TETAP
jalan pakai
    harga default (tidak crash).
    """
    global harga_per_ekor_aktif
    try:
        res = supabase.table("pengaturan").select("*").eq("id",
1).execute()
        if res.data:
            harga_per_ekor_aktif = int(
                res.data[0].get("harga_per_ekor") or
HARGA_PER_EKOR_DEFAULT

```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

    )
    print(f"[PENGATURAN] Harga per ekor aktif: Rp
{harga_per_ekor_aktif}")
    else:
        harga_per_ekor_aktif = HARGA_PER_EKOR_DEFAULT
    except Exception as e:
        harga_per_ekor_aktif = HARGA_PER_EKOR_DEFAULT
    print(
        "[PENGATURAN] Gagal ambil harga per ekor dari
Supabase, "
        f"pakai default Rp {HARGA_PER_EKOR_DEFAULT}. Error:",
        e,
    )
    return harga_per_ekor_aktif

# ===== GUI FUNCTIONS =====
def start_counting():
    global running, start_time, waktu_mulai
    global count_zone, count_single, count_dual
    global frame_count
    global fps_total, fps_samples, avg_fps
    global confidence_total, confidence_count, avg_confidence
    global session_id, last_synced_counts

    nama = name_entry.get().strip()
    if not nama:
        status_label.config(text="Nama wajib diisi!", fg="orange")
        return

    count_zone = count_single = count_dual = 0
    frame_count = 0
    fps_total = fps_samples = avg_fps = 0
    confidence_total = confidence_count = avg_confidence = 0
    last_synced_counts = (0, 0, 0)
    tracker.reset()

    start_time = time.time()
    waktu_mulai = datetime.utcnow()
    running = True

    # Sinkron harga per ekor terkini dari Admin Panel di awal sesi
    ambil_harga_per_ekor()

    # ===== REALTIME SAVE: insert baris sesi SEKARANG (bukan
    nunggu Stop) =====
    # Kolom count_zone/single/dual dimulai dari 0, lalu di-UPDATE
    LENGKAP
    # (count, harga, status, durasi) setiap kali ada perubahan -
    # lihat process_frame -> push_realtime_update.
    harga_awal = hitung_harga(0)
    try:
        row = {
            "nama": nama,
            "aktual": 0,
            "count_zone": 0,
            "count_single": 0,
            "count_dual": 0,
            "harga": harga_awal,
            "status": "Sedang Berjalan",

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        "durasi": 0,
        "waktu_mulai": waktu_mulai.isoformat(),
        "waktu_selesai": None,
        "metode": f"SORT Kalman + {'ONNX Runtime' if USE_OR
else 'cv2.dnn'}",
    }
    result =
supabase.table("history_count").insert(row).execute()
    session_id = result.data[0]["id"]
    print(f"[REALTIME] Sesi dibuat di Supabase,
id={session_id}")
    except Exception as e:
        session_id = None
        print("[REALTIME] Gagal membuat baris sesi awal:", e)

    status_label.config(text="Status: RUNNING", fg="green")
    m1_label.config(text="M1 Zone: 0")
    m2_label.config(text="M2 Single: 0")
    m3_label.config(text="M3 Dual: 0")
    print("START COUNTING")

def hitung_harga(cz):
    """Harga total = jumlah ikan (count_zone) x harga per ekor
yang sedang
    aktif (diambil dari Admin Panel saat sesi dimulai). Tidak ada
    lagi
    konsep paket kecil/besar – semua dihitung satuan per ekor."""
    return int(cz) * harga_per_ekor_aktif

def push_realtime_update(id_sesi, payload):
    """Kirim update LENGKAP (count, paket, harga, status, durasi,
dst) ke
    Supabase di thread terpisah supaya TIDAK memblokir loop
    deteksi/FPS."""

    def _ kirim():
        try:
            supabase.table("history_count").update(payload).eq("id
", id_sesi).execute()
        except Exception as e:
            print("[REALTIME] Gagal push update:", e)

    threading.Thread(target=_ kirim, daemon=True).start()

# ===== Tombol STOP/SAVE terpisah DIHAPUS =====
# Setiap ikan yang lewat zona counting SUDAH langsung ter-update
    penuh ke
# Supabase secara realtime (count, paket, harga, status, durasi,
    dst) lewat
# push_realtime_update() di process_frame(). Baris di Supabase
    SELALU
# mencerminkan kondisi terkini, bukan cuma di akhir sesi.
#
# RESET di bawah ini polos: cuma reset counter & tampilan lokal ke
    0, TIDAK
# perlu kirim apa pun lagi ke Supabase – karena data terakhir yang
    sudah
# terkirim (sebelum Reset ditekan) itulah hasil akhir sesi yang
    tersimpan.

```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
def reset_counting():
    global running, count_zone, count_single, count_dual
    global start_time, waktu_mulai
    global fps_total, fps_samples, avg_fps
    global confidence_total, confidence_count, avg_confidence,
    frame_count
    global session_id, last_synced_counts

    # Update status sesi aktif ke Selesai sebelum reset
    if session_id is not None:
        try:
            durasi_akhir = 0 if start_time is None else
            int(time.time() - start_time)
            supabase.table("history_count").update(
                {
                    "status": "Selesai",
                    "harga": hitung_harga(count_zone),
                    "durasi": durasi_akhir,
                    "waktu_selesai":
                    datetime.utcnow().isoformat(),
                }
            ).eq("id", session_id).execute()
            print(f"[RESET] Sesi {session_id} ditandai Selesai.")
        except Exception as e:
            print("[RESET] Gagal update status sesi:", e)

    running = False
    count_zone = count_single = count_dual = 0
    start_time = waktu_mulai = None
    fps_total = fps_samples = avg_fps = 0
    confidence_total = confidence_count = avg_confidence = 0
    frame_count = 0
    session_id = None
    last_synced_counts = (0, 0, 0)
    tracker.reset()

    durasi_label.config(text="Durasi: 00:00")
    m1_label.config(text="M1 Zone: 0")
    m2_label.config(text="M2 Single: 0")
    m3_label.config(text="M3 Dual: 0")
    status_label.config(text="Status: STOPPED", fg="red")
    aktual_entry.delete(0, tk.END)
    print("RESET")

def exit_app():
    """Tutup aplikasi. Kalau ada sesi yang masih berjalan (status
    'Sedang Berjalan'),
    update statusnya jadi 'Selesai' dulu ke Supabase sebelum
    keluar
    supaya di History web tidak stuck 'Sedang Berjalan'
    selamanya."""
    global running, session_id

    # Update status sesi aktif ke Selesai sebelum keluar
    if session_id is not None:
        try:
            durasi_akhir = 0 if start_time is None else
            int(time.time() - start_time)
            harga_akhir = hitung_harga(count_zone)
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

payload_akhir = {
    "status": "Selesai",
    "harga": harga_akhir,
    "durasi": durasi_akhir,
    "waktu_selesai": datetime.utcnow().isoformat(),
}
# Kirim SINKRON (bukan thread) karena app mau langsung
ditutup
supabase.table("history_count").update(payload_akhir).
eq(
    "id", session_id
).execute()
print(f"[EXIT] Sesi {session_id} ditandai Selesai di
Supabase.")
except Exception as e:
    print("[EXIT] Gagal update status sesi:", e)

running = False
session_id = None

cap.release()
if GPIO_AVAILABLE:
    GPIO.output(PIN_RELAY_POMPA, GPIO.LOW)
    GPIO.cleanup()
root.destroy()

def cek_status_pompa():
    """Polling berkala ke Supabase untuk sinkronisasi status pompa
    yang dikontrol dari Admin Panel (cloud). Dijalankan lewat
    root.after
    supaya tidak memblokir loop utama Tkinter."""
    global pompa_status_terakhir
    try:
        res =
supabase.table("device_control").select("*").eq("id", 1).execute()
        if res.data:
            status = res.data[0].get("pompa_status", "off")
            if status != pompa_status_terakhir:
                pompa_status_terakhir = status
                if GPIO_AVAILABLE:
                    GPIO.output(
                        PIN_RELAY_POMPA, GPIO.HIGH if status ==
"on" else GPIO.LOW
                    )
                print(
                    f"[POMPA] Status berubah -> {status}"
                    f"{' ' if GPIO_AVAILABLE else ' ' (simulasi, GPIO
tidak aktif)'}"
                )
                pompa_label.config(
                    text=f"Pompa: {'MENYALA' if status == 'on'
else 'MATI'}",
                    fg="#21c55d" if status == "on" else "#9ca3af",
                )
            except Exception as e:
                print("[POMPA] Gagal cek status:", e)

root.after(POMPA_POLL_INTERVAL_MS, cek_status_pompa)

```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# ===== DETECTION & TRACKING =====
def process_frame():
    global count_zone, count_single, count_dual
    global frame_count
    global prev_time, fps_total, fps_samples, avg_fps
    global confidence_total, confidence_count, avg_confidence
    global durasi_display
    global session_id, last_synced_counts

    ret, frame = cap.read()
    if not ret:
        if IS_VIDEO_FILE:
            # Video habis -> ulang dari awal, supaya bisa terus
            # perlu restart program berkali-kali.
            cap.set(cv2.CAP_PROP_POS_FRAMES, 0)
            root.after(10, process_frame)
            return
        root.after(10, process_frame)
        return

    frame_count += 1
    image = cv2.resize(frame, (320, 320))
    h, w = image.shape[:2]

    # ===== INFERENCE =====
    outputs, ratio, dw, dh = run_inference(image)

    detections = []
    for row in outputs:
        conf = float(row[4])
        if conf > 0.30: # diturunkan dari 0.40 agar lebih banyak
            ikan terdeteksi
            x, y, bw, bh = row[:4]

            if USE_ORT:
                # Kembalikan koordinat ke skala image dengan
                letterbox offset
                x1 = int((x - bw / 2 - dw) / ratio)
                y1 = int((y - bh / 2 - dh) / ratio)
                x2 = int((x + bw / 2 - dw) / ratio)
                y2 = int((y + bh / 2 - dh) / ratio)
            else:
                x1 = int((x - bw / 2) * w / INPUT_SIZE)
                y1 = int((y - bh / 2) * h / INPUT_SIZE)
                x2 = int((x + bw / 2) * w / INPUT_SIZE)
                y2 = int((y + bh / 2) * h / INPUT_SIZE)

            x1 = max(0, x1)
            y1 = max(0, y1)
            x2 = min(w, x2)
            y2 = min(h, y2)
            area = (x2 - x1) * (y2 - y1)
            if area < MIN_AREA:
                continue
            detections.append([x1, y1, x2, y2, conf])
            if running:
                confidence_total += conf
                confidence_count += 1
```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.

b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta

2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        avg_confidence = confidence_total /
confidence_count

    # NMS
    boxes = [[d[0], d[1], d[2] - d[0], d[3] - d[1]] for d in
detections]
    scores = [d[4] for d in detections]
    filtered = []
    if boxes:
        idxs = cv2.dnn.NMSBoxes(
            boxes, scores, 0.30, 0.40
        ) # threshold diturunkan, NMS dinaikkan
        if len(idxs) > 0:
            for i in (idxs.flatten() if hasattr(idxs, "flatten")
else idxs):
                filtered.append(detections[int(i)])
    detections = filtered

    # ===== SORT TRACKER UPDATE =====
    active_tracks = tracker.update(detections)

    # ===== COUNTING =====
    # Semua metode counting sekarang berbasis PER-ID + hysteresis,
    bukan lagi
    # perbandingan satu-frame mentah. Ini menghilangkan 2 sumber
    utama hasil
    # tidak stabil sebelumnya:
    # 1) M1 (zone) dulu dihitung dari selisih jumlah objek di
    zona antar-frame
    # -> ikan yang goyang/jitter keluar-masuk zona bisa
    dihitung berkali2.
    # Sekarang: 1 ID cuma dihitung SEKALI saat pertama kali
    stabil masuk zona.
    # 2) M2 dulu pakai cooldown GLOBAL antar-ikan
    (COUNT_COOLDOWN) -> kalau
    # 2 ikan lewat garis hampir bersamaan, yang kedua bisa
    tidak terhitung
    # sama sekali. Sekarang setiap track py hitung independen
    (tidak saling
    # memblokir), cukup dijaga oleh flag per-ID
    (counted_single/counted_dual).
    if running:
        for track in active_tracks:
            cx, cy = track.get_center()

            # ---- M1: ZONA (hysteresis, hitung sekali per ID) ---
            -

            if cy > zone_bottom + LINE_MARGIN:
                zone_state = "luar_bawah"
            elif cy < zone_top - LINE_MARGIN:
                zone_state = "luar_atas"
            elif zone_top <= cy <= zone_bottom:
                zone_state = "dalam"
            else:
                zone_state = track.zone_state # di dead-zone
            margin, pertahankan

            if zone_state == "dalam" and not track.counted_zone:
                count_zone += 1

```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

        track.counted_zone = True
        track.zone_state = zone_state

        # ---- Sisi terhadap L1 (hysteresis) ----
        if cy > line1_y + LINE_MARGIN:
            side_l1 = "bawah"
        elif cy < line1_y - LINE_MARGIN:
            side_l1 = "atas"
        else:
            side_l1 = track.side_l1

        # ---- Sisi terhadap L2 (hysteresis) ----
        if cy > line2_y + LINE_MARGIN:
            side_l2 = "bawah"
        elif cy < line2_y - LINE_MARGIN:
            side_l2 = "atas"
        else:
            side_l2 = track.side_l2

        # ---- M2: SINGLE LINE (L2) - hitung saat transisi
        atas -> bawah ----
        if (
            not track.counted_single
            and track.side_l2 == "atas"
            and side_l2 == "bawah"
        ):
            count_single += 1
            track.counted_single = True

        # ---- M3: DUAL LINE (harus lewat L1 dulu, baru L2) --
        --
        if (
            not track.passed_line1
            and track.side_l1 == "atas"
            and side_l1 == "bawah"
        ):
            track.passed_line1 = True

        if (
            track.passed_line1
            and not track.counted_dual
            and track.side_l2 == "atas"
            and side_l2 == "bawah"
        ):
            count_dual += 1
            track.counted_dual = True

        track.side_l1 = side_l1
        track.side_l2 = side_l2
        track.prev_cy = cy

        # ===== REALTIME SAVE: push update LENGKAP setiap ada
        perubahan apa pun =====
        # Dicek setelah semua metode counting (M1/M2/M3) selesai
        diproses di atas,
        # supaya count_single/count_dual yang bertambah di tengah loop
        (bukan cuma
        # count_zone) juga ikut ter-sync, bukan cuma count_zone
        seperti sebelumnya.

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

if running and session_id is not None:
    counts_sekarang = (count_zone, count_single, count_dual)
    if counts_sekarang != last_synced_counts:
        last_synced_counts = counts_sekarang
        harga = hitung_harga(count_zone)

    try:
        aktual_input = int(aktual_entry.get().strip())
    except ValueError:
        aktual_input = 0

    durasi_sekarang = 0 if start_time is None else
int(time.time() - start_time)

    payload = {
        "aktual": aktual_input,
        "count_zone": count_zone,
        "count_single": count_single,
        "count_dual": count_dual,
        "harga": harga,
        "status": "Sedang Berjalan",
        "durasi": durasi_sekarang,
        # waktu_selesai dipakai sebagai "waktu update
terakhir" selama
        # sesi masih realtime berjalan (tidak ada lagi
tombol Stop
        # yang menandai akhir sesi secara eksplisit).
        "waktu_selesai": datetime.utcnow().isoformat(),
    }
    push_realtime_update(session_id, payload)

# ===== DRAW =====
cv2.rectangle(image, (0, zone_top), (w, zone_bottom), (255,
180, 0), 1)
cv2.line(image, (0, line1_y), (w, line1_y), (0, 255, 255), 2)
cv2.line(image, (0, line2_y), (w, line2_y), (255, 0, 0), 2)
cv2.line(image, (0, line3_y), (w, line3_y), (0, 0, 255), 2)
cv2.putText(
    image, "L1", (5, line1_y - 5), cv2.FONT_HERSHEY_SIMPLEX,
0.4, (0, 255, 255), 1
)
cv2.putText(
    image, "L2", (5, line2_y - 5), cv2.FONT_HERSHEY_SIMPLEX,
0.4, (255, 0, 0), 1
)
cv2.putText(
    image, "L3", (5, line3_y - 5), cv2.FONT_HERSHEY_SIMPLEX,
0.4, (0, 0, 255), 1
)

# Engine label di pojok kanan atas video
cv2.putText(
    image,
    "ORT" if USE_ORT else "DNN",
    (w - 35, 12),
    cv2.FONT_HERSHEY_SIMPLEX,
    0.38,
    (0, 255, 128) if USE_ORT else (0, 165, 255),
    1,

```




Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

)

for track in active_tracks:
    x1, y1, x2, y2 = track.get_state()
    cx, cy = track.get_center()

    color = (
        (0, 255, 0)
        if track.counted_dual
        else (0, 165, 255) if track.passed_line1 else (255,
255, 0)
    )

    cv2.rectangle(image, (x1, y1), (x2, y2), color, 2)
    label = f"ID:{track.id}"
    cv2.rectangle(image, (x1, y1 - 16), (x1 + len(label) * 8,
y1), color, -1)
    cv2.putText(
        image, label, (x1 + 2, y1 - 3),
cv2.FONT_HERSHEY_SIMPLEX, 0.38, (0, 0, 0), 1
    )
    cv2.circle(image, (cx, cy), 3, (0, 0, 255), -1)

cv2.putText(
    image,
    f"Frame:{frame_count}",
    (5, 12),
    cv2.FONT_HERSHEY_SIMPLEX,
    0.38,
    (200, 200, 200),
    1,
)

# ===== FPS =====
current_time = time.time()
elapsed = current_time - prev_time
fps = 1 / elapsed if prev_time != 0 and elapsed > 0 else 0
prev_time = current_time

if running and fps > 0:
    fps_total += fps
    fps_samples += 1
    avg_fps = fps_total / fps_samples

if running and start_time is not None:
    durasi_display = int(time.time() - start_time)
    m = durasi_display // 60
    s = durasi_display % 60
    durasi_label.config(text=f"Durasi: {m:02d}:{s:02d}")

fps_label.config(text=f"FPS: {int(fps)}")
avg_fps_label.config(text=f"Avg FPS: {round(avg_fps, 2)}")
m1_label.config(text=f"M1 Zone: {count_zone}")
m2_label.config(text=f"M2 Single: {count_single}")
m3_label.config(text=f"M3 Dual: {count_dual}")

image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
img_pil = Image.fromarray(image_rgb)

```



Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```
# Deteksi & counting SELALU jalan tiap frame di atas (tidak
berubah).
# Yang di-skip cuma render ke widget GUI, karena Tkinter cukup
terasa
# mulus di ~15 FPS meski kamera/inference jalan lebih cepat
dari itu.
global gui_render_counter
gui_render_counter += 1
if gui_render_counter % 2 == 0:
    imgTk = ImageTk.PhotoImage(image=img_pil)
    video_label.imgtk = imgTk
    video_label.configure(image=imgTk)
    sinkron_ke_monitor(
        img_pil
    ) # kirim versi PIL supaya bisa di-upscale untuk layar
monitor

root.after(1, process_frame)

# ===== GUI LAYOUT =====
root = tk.Tk()
root.title(f"Fish Counter AI - {'ONNX Runtime' if USE_OR_T else
'cv2.dnn'}")
root.geometry("760x680")
root.resizable(False, False)

tk.Label(root, text="FISH COUNTER AI", font=("Arial", 20,
"bold")).pack(pady=8)

# Engine badge di bawah title
engine_color = "#48bb78" if USE_OR_T else "#f59e0b"
engine_text = (
    "Engine: ONNX Runtime ✅ "
    if USE_OR_T
    else "Engine: cv2.dnn ⚠️ (install onnxruntime)"
)
tk.Label(root, text=engine_text, font=("Arial", 9),
fg=engine_color).pack(pady=(0, 4))

main_frame = tk.Frame(root)
main_frame.pack()

video_label = tk.Label(main_frame)
video_label.grid(row=0, column=0, rowspan=12, padx=15, pady=10)

side_frame = tk.Frame(main_frame)
side_frame.grid(row=0, column=1, padx=15, sticky="n")

fps_label = tk.Label(side_frame, text="FPS: 0", font=("Arial", 12,
"bold"))
fps_label.pack(pady=3)
avg_fps_label = tk.Label(side_frame, text="Avg FPS: 0",
font=("Arial", 12, "bold"))
avg_fps_label.pack(pady=3)
durasi_label = tk.Label(side_frame, text="Durasi: 00:00",
font=("Arial", 12, "bold"))
durasi_label.pack(pady=3)
```




© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

tk.Frame(side_frame, height=1, bg="#444").pack(fill="x", pady=6)

m1_label = tk.Label(
    side_frame, text="M1 Zone: 0", font=("Arial", 13, "bold"),
    fg="#4a90e2"
)
m1_label.pack(pady=3)
m2_label = tk.Label(
    side_frame, text="M2 Single: 0", font=("Arial", 13, "bold"),
    fg="#f0a500"
)
m2_label.pack(pady=3)
m3_label = tk.Label(
    side_frame, text="M3 Dual: 0", font=("Arial", 14, "bold"),
    fg="#48bb78"
)
m3_label.pack(pady=3)

tk.Frame(side_frame, height=1, bg="#444").pack(fill="x", pady=6)

status_label = tk.Label(
    side_frame, text="Status: STOPPED", font=("Arial", 12,
    "bold"), fg="red"
)
status_label.pack(pady=4)

pompa_label = tk.Label(
    side_frame, text="Pompa: MATI", font=("Arial", 11, "bold"),
    fg="#9ca3af"
)
pompa_label.pack(pady=(0, 4))

tk.Label(side_frame, text="Nama Pembeli:", font=("Arial",
11)).pack(pady=(10, 2))
name_entry = tk.Entry(side_frame, font=("Arial", 11), width=22)
name_entry.pack(pady=3)

tk.Label(side_frame, text="Jumlah Aktual:", font=("Arial",
11)).pack(pady=(6, 2))
aktual_entry = tk.Entry(side_frame, font=("Arial", 11), width=22)
aktual_entry.pack(pady=3)

tk.Button(
    side_frame,
    text="START",
    font=("Arial", 12, "bold"),
    width=18,
    bg="#21c55d",
    fg="white",
    command=start_counting,
).pack(pady=4)
tk.Button(
    side_frame,
    text="RESET",
    font=("Arial", 12, "bold"),
    width=18,
    bg="#f59e0b",
    fg="white",
    command=reset_counting,

```



© Hak Cipta milik Politeknik Negeri Jakarta

Hak Cipta :

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber :
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian , penulisan karya ilmiah, penulisan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar Politeknik Negeri Jakarta
2. Dilarang mengumunkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin Politeknik Negeri Jakarta

```

).pack(pady=4)
tk.Button(
    side_frame,
    text="MODE MONITOR (Esc keluar)",
    font=("Arial", 11, "bold"),
    width=18,
    bg="#0ea5e9",
    fg="white",
    command=buka_mode_monitor,
).pack(pady=4)
tk.Button(
    side_frame,
    text="EXIT",
    font=("Arial", 12, "bold"),
    width=18,
    bg="#374151",
    fg="white",
    command=exit_app,
).pack(pady=4)

process_frame()
cek_status_pompa()
root.mainloop()

```

**POLITEKNIK
NEGERI
JAKARTA**