

**SMART WASTE MANAGEMENT SYSTEM: INOVASI TONG SAMPAH
IOT DENGAN MEKANISME OTOMATISASI DAN KEAMANAN
MODERN**

SKRIPSI

DISUSUN OLEH

JAMILUDDIN TANJUNG
NPM. 2109020140



UMSU
Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2025

**SMART WASTE MANAGEMENT SYSTEM: INOVASI TONG SAMPAH
IOT DENGAN MEKANISME OTOMATISASI DAN KEAMANAN
MODERN**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana
Komputer (S.Kom) dalam Program Studi Teknologi Informasi pada
Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas
Muhammadiyah Sumatera Utara**

JAMILUDDIN TANJUNG

NPM. 2109020140

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
MEDAN**

2025

LEMBAR PENGESAHAN

Judul Skripsi : SMART WASTE MANAGEMENT SYSTEM: INOVASI
TONG SAMPAH IOT DENGAN MEKANISME
OTOMATISASI DAN KEAMANAN MODERN

Nama Mahasiswa : JAMILUDDIN TANJUNG

NPM : 2109020140

Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



Yohanni Swara, S.Si., M.Kom

NIDN. 0129108201

Ketua Program Studi



Fatma Sari Hutagalung, M.Kom

NIDN. 0117019301

Dekan



Dr. Al-Khowarizmi, S.Kom., M.Kom.

NIDN. 0127099201

PERNYATAAN ORISINALITAS

SMART WASTE MANAGEMENT SYSTEM: INOVASI TONG SAMPAH IOT DENGAN MEKANISME OTOMATISASI DAN KEAMANAN MODERN

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya.

Medan, 29 September 2025

Yang membuat pernyataan



Jamiluddin Tanjung

2109020140

**PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS**

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Jamiluddin Tanjung
NPM : 2109020140
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bebas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

**SMART WASTE MANAGEMENT SYSTEM: INOVASI TONG SAMPAH
IOT DENGAN MEKANISME OTOMATISASI DAN KEAMANAN
MODERN**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 29 September 2025

Yang membuat pernyataan

Jamiluddin Tanjung

2109020140

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Jamiluddin Tanjung
Tempat dan Tanggal Lahir : Sialang Gatap, 09-01-2000
Alamat Rumah : Jl.Darussalam, Desa Teluk Piai
Telepon/Faks/HP : 085274723257
E-mail : jamiltanjung99@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD	: SDN 117520	TAMAT: 2012
SMP	: MTS YMPI	TAMAT: 2015
SMA	: MA YMPI	TAMAT: 2018

KATA PENGANTAR



Pendahuluan

Penulis tentunya berterima kasih kepada berbagai pihak dalam dukungan serta doa dalam penyelesaian skripsi. Penulis juga mengucapkan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP., Rektor Universitas Muhammadiyah Sumatera Utara (UMSU)
2. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom. Dekan Fakultas Ilmu Komputer dan Teknologi Informasi (FIKTI) UMSU.
3. Ibu Fatma Sari Hutagalung, M.Kom. Ketua Program Studi Teknologi Informasi
4. Bapak Mhd.Basri, S.SI. M.Kom. Sekretaris Program Studi Teknologi Informasi
5. Pembimbing Ibu Yohanni Syahra.S.Si.M.Kom.
6. Terimakasih untuk kedua Orang tua atas cinta dan kasih sayang, doa dan dukungan moral maupun material yang tiada henti.
7. Semua pihak yang telah memberikan bantuan baik secara langsung maupun tidak langsung yang tidak dapat penulis sebutkan satu per satu.

Semua pihak yang terlibat langsung ataupun tidak langsung yang tidak dapat penulis ucapkan satu-persatu yang telah membantu penyelesaian skripsi ini.

SMART WASTE MANAGEMENT SYSTEM: INOVASI TONG SAMPAH IOT DENGAN MEKANISME OTOMATISASI DAN KEAMANAN MODERN

ABSTRAK

Penelitian ini mengembangkan *Smart Waste Management System* berbasis Internet of Things (IoT) dengan integrasi *Convolutional Neural Network* (CNN) untuk meningkatkan efisiensi pemilahan dan pengelolaan sampah. Sistem dirancang menggunakan modul *ESP32-CAM* untuk menangkap citra sampah, kemudian mengirimkannya ke server lokal berbasis Flask untuk diklasifikasikan menjadi empat kategori: organik, anorganik, limbah medis, dan limbah berbahaya (B3). Hasil klasifikasi digunakan untuk menggerakkan tiga motor servo yang mengarahkan sampah ke tong sesuai kategori. Empat sensor ultrasonik dipasang pada setiap tong guna mendeteksi kapasitas volume, sedangkan LED indikator dan buzzer memberikan peringatan lokal ketika tong penuh. Seluruh data status dikirim ke platform *Node-RED* melalui protokol MQTT dan ditampilkan pada dashboard secara real-time. Hasil pengujian menunjukkan sistem mampu mengklasifikasikan sampah dengan akurasi tinggi, mendeteksi kapasitas secara tepat, serta memberikan notifikasi responsif. Inovasi ini diharapkan menjadi solusi efektif dalam mendukung pengelolaan sampah cerdas, efisien, dan berkelanjutan.

Kata Kunci: *Internet of Things, ESP32-CAM, Convolutional Neural Network, Node-RED, Smart Waste Management.*

SMART WASTE MANAGEMENT SYSTEM: IOT BIN INNOVATION WITH MODERN AUTOMATION AND SECURITY MECHANISMS

ABSTRACT

This research develops a Smart Waste Management System based on the Internet of Things (IoT) with Convolutional Neural Network (CNN) integration to improve the efficiency of waste sorting and management. The system is designed using the ESP32-CAM module to capture images of waste, then sends it to a local Flask-based server to be classified into four categories: organic, inorganic, medical waste, and hazardous waste (B3). The classification results are used to drive three servo motors that direct waste to the bins according to the category. Four ultrasonic sensors are installed on each keg to detect volume capacity, while indicator LEDs and buzzers provide local alerts when the keg is full. All status data is sent to the Node-RED platform via the MQTT protocol and displayed on the dashboard in real-time. The test results showed that the system was able to classify waste with high accuracy, detect capacity appropriately, and provide responsive notifications. This innovation is expected to be an effective solution in supporting smart, efficient, and sustainable waste management.

Keywords: *Internet of Things, ESP32-CAM, Convolutional Neural Network, Node-RED, Smart Waste Management.*

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS.....	ii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
RIWAYAT HIDUP	iv
KATA PENGANTAR.....	v
ABSTRAK	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	5
1.4 Tujuan Penelitian.....	6
1.5 Manfaat Penelitian.....	6
BAB II LANDASAN TEORI	8
2.1 Internet of Things (IoT).....	8
2.2 Mikrokontroler ESP32 dan NodeMCU	8
2.3 Sensor Ultrasonik	10
2.4 Kamera ESP32-CAM	10
2.5 Motor Servo.....	11
2.6 Buzzer dan LED	12
2.7 Machine Learning.....	13
2.8 Convolutional Neural Network (CNN)	14
2.9 Klasifikasi Jenis Sampah.....	15
2.10 Node-RED	16
2.11 Flask	17
2.12 Flowchart.....	18
2.13 Penelitian Terkait	19

BAB III METODOLOGI PENELITIAN	24
3.1 Metode Penelitian.....	24
3.1.1 Karakteristik Metode Observasi	25
3.2 Lokasi dan Waktu Penelitian.....	25
3.2.1 Tempat Pengembangan Sistem.....	26
3.2.2 Tempat Pengujian Prototipe.....	26
3.2.3 Waktu Penelitian.....	27
3.3 Alur Penelitian.....	27
3.3.1 Studi Literatur dan Identifikasi Masalah	27
3.3.2 Perancangan Sistem (Hardware dan Software)	28
3.3.3 Pengumpulan Dataset Gambar Sampah.....	30
3.3.4 Pelatihan Model CNN.....	31
3.3.5 Implementasi Sistem.....	32
3.3.6 Pengujian Sistem.....	33
3.3.7 Evaluasi dan Analisis Hasil	33
3.3.8 Dokumentasi dan Penulisan Laporan	33
3.4 Alat dan Bahan Penelitian	34
3.4.1 Alat Elektronik dan Perangkat Keras.....	34
3.4.2 Tong Sampah Fisik	35
3.4.3 Perangkat Lunak	36
3.5 Perancangan Sistem.....	37
3.5.1 Perancangan Sistem Berbasis IoT	39
3.5.2 Perancangan Model CNN untuk Klasifikasi Sampah.....	40
3.5.3 Integrasi CNN dengan Sistem IoT Secara Utuh	44
3.6 Pengumpulan Data	45
3.7 Flowchart CNN	49
3.8 Pengujian Sistem	50
3.9 Waktu Penelitian	53
BAB IV HASIL DAN PEMBAHASAN	55
4.1 Implementasi Sistem	55
4.1.1 Perangkat Keras	55
4.1.2 Perangkat Lunak	56
4.2 Pengujian Sistem	57
4.2.1 Pengujian Sensor Ultrasonik.....	57
4.2.2 Pengujian Motor Servo	58

4.2.3 Pengujian Modul ESP32-CAM	59
4.2.4 Pengujian Model CNN (Klasifikasi Sampah).....	60
4.2.5 Pengujian Integrasi Sistem IoT (Node-RED Dashboard).....	61
4.3 Pembahasan	62
4.3.1 Pembahasan Sensor Ultrasonik.....	62
4.3.2 Pembahasan Modul ESP32-CAM	63
4.3.3 Pembahasan Model CNN.....	64
4.3.4 Pembahasan Integrasi Node-RED	65
BAB V KESIMPULAN & SARAN	67
5.1 Kesimpulan.....	67
5.2 Saran	67
DAFTAR PUSTAKA	69
Lampiran	71

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait	20
Tabel 3. 1 Waktu Penelitian	53
Tabel 4. 1 Pengujian Sensor Ultrasonik	58
Tabel 4. 2 Pengujian Motor Servo	59
Tabel 4. 3 Pengujian Modul ESP32-CAM.....	60
Tabel 4. 4 Pengujian Model CNN (Klasifikasi Sampah).....	61
Tabel 4. 5 Pengujian Integrasi Sistem IoT (Node-RED Dashboard).....	62

DAFTAR GAMBAR

Gambar 2. 1 Mikrokontroler ESP32 (Universitas Raharja, 2021)	9
Gambar 2. 2 Sensor Ultrasonik (Arduino Indonesia, 2022).....	10
Gambar 2. 3 Kamera ESP32-CAM (CNX Software, 2018)	11
Gambar 2. 4 Motor Servo (Arduino Indonesia, 2022).....	12
Gambar 2. 5 Buzzer (Indobot, 2022)	12
Gambar 2. 6 LED (Setiyo, 2023)	12
Gambar 2. 7 Convolutional Neural Network (Sena, 2017).....	14
Gambar 2. 8 Node-RED (Parker Enfield, 2017).....	16
Gambar 2. 9 Flowchart Sistem (Itbox.id, 2023).....	19
Gambar 3. 1 Diagram Alur Penelitian.....	27
Gambar 3. 2 Diagram Alur Perancangan Sitem.....	38
Gambar 3. 3 Sampel Gambar Sampah Organik	47
Gambar 3. 4 Sampel Gambar Sampah Anorganik	47
Gambar 3. 5 Sampel Gambar Sampah Berbahaya (B3).....	47
Gambar 3. 6 Sampel Gambar Sampah Limbah Medis.....	48
Gambar 3. 7 Flowchart CNN	49
Gambar 3. 8 Flowchart Sistem.....	51
Gambar 4. 1 Aktivasi Sensor Ultrasonik.....	63
Gambar 4. 2 Aktivasi modul esp32-cam.....	63
Gambar 4. 3 Model Machine learning (CNN)	64
Gambar 4. 4 Integrasi Node-red.....	66

BAB I

PENDAHULUAN

1.1.Latar Belakang Masalah

Perkembangan teknologi digital dan konektivitas nirkabel telah melahirkan inovasi yang signifikan dalam bidang otomatisasi, salah satunya melalui pemanfaatan Internet of Things (IoT). IoT memungkinkan objek fisik terhubung ke jaringan dan saling bertukar data secara real-time. Dalam konteks pengelolaan lingkungan, teknologi ini dapat diimplementasikan untuk menciptakan sistem tempat sampah pintar yang mampu beroperasi secara otomatis dan efisien (Ashari et al., 2024). Penggabungan berbagai sensor, mikrokontroler, dan jaringan internet menghadirkan solusi yang lebih adaptif terhadap tantangan pengelolaan sampah di era modern.

Beberapa penelitian sebelumnya telah mengembangkan tempat sampah otomatis berbasis mikrokontroler seperti ESP32 dan NodeMCU. Sistem ini dilengkapi dengan sensor ultrasonik untuk mengukur kapasitas sampah, serta mekanisme pemilahan yang digerakkan oleh motor servo. Saat volume sampah mendekati penuh, sistem dapat memberikan informasi melalui indikator visual atau platform monitoring (Ashari et al., 2024). Fitur ini meningkatkan higienitas dan kenyamanan pengguna, serta membantu petugas dalam memantau kondisi tempat sampah secara efisien.

Namun, persoalan pengelolaan sampah tidak berhenti pada efisiensi pembuangan saja. Tantangan yang lebih kompleks muncul dari kurangnya pemisahan jenis sampah, yang sering kali menyebabkan pencampuran antara

sampah organik, non-organik, bahkan limbah berbahaya atau medis. Studi yang dilakukan oleh Firdaus et al. (2025) menunjukkan bahwa sistem klasifikasi sampah berbasis pengenalan citra digital dengan algoritma Convolutional Neural Network (CNN) memiliki akurasi tinggi dalam membedakan jenis sampah infeksius secara otomatis. Teknologi ini memungkinkan tempat sampah pintar melakukan fungsi pemilahan dengan lebih cermat dan mengurangi risiko kontaminasi.

Permasalahan limbah medis dan non-medis juga menjadi isu krusial di berbagai fasilitas kesehatan. Berdasarkan hasil penelitian di Rumah Sakit Budi Agung Kota Palu, diketahui bahwa masih banyak limbah organik dan non-organik yang disatukan dalam satu wadah, serta kurangnya pemahaman petugas terhadap prosedur pemilahan yang aman (Kotika et al., 2023). Minimnya pewadahan yang sesuai dan pengawasan yang ketat memperbesar potensi bahaya kesehatan dan pencemaran lingkungan. Oleh karena itu, dibutuhkan sistem yang tidak hanya otomatis, tetapi juga cerdas dan adaptif terhadap jenis sampah yang dibuang.

Di sisi lain, persoalan pengelolaan sampah di masyarakat masih menjadi tantangan yang signifikan, khususnya di ruang-ruang publik seperti sekolah, rumah sakit, perkantoran, dan pasar. Kurangnya fasilitas tempat sampah yang mendukung sistem pemisahan jenis sampah menyebabkan sampah seringkali tercampur tanpa klasifikasi. Dalam praktiknya, limbah organik, non-organik, hingga limbah medis dan berbahaya masih sering ditemukan dalam satu wadah yang sama. Hal ini tentu berpotensi menimbulkan pencemaran lingkungan, meningkatkan risiko penularan penyakit, dan membahayakan keselamatan, terutama ketika limbah yang dibuang bersifat mudah terbakar atau beracun (Kotika et al., 2023).

Sistem pengawasan tempat sampah yang masih bergantung pada kontrol manual juga menyebabkan banyak tempat sampah terabaikan dalam kondisi sudah penuh atau menimbulkan bau tak sedap sebelum ditangani. Minimnya pemantauan dan keterlambatan dalam pengosongan menyebabkan sampah meluber dan mencemari lingkungan sekitarnya. Hal ini diperburuk oleh tidak adanya sistem notifikasi volume secara real-time, sehingga efisiensi kerja petugas kebersihan menjadi rendah dan tidak tepat sasaran (Ashari et al., 2024). Oleh karena itu, perlu dirancang sebuah sistem yang mampu melakukan deteksi volume sampah secara otomatis serta mendukung pemantauan jarak jauh melalui sistem digital.

Penerapan sistem berbasis pengolahan citra digital dan klasifikasi sampah memerlukan platform monitoring andal untuk memantau kinerja secara jarak jauh. Node-RED menjadi salah satu platform IoT yang efektif karena mampu menerima data dari broker MQTT, menampilkannya secara real-time melalui dashboard, dan menyimpan data historis untuk analisis. Platform ini memungkinkan pemantauan kondisi tempat sampah dari browser mana pun, serta merespons cepat saat kapasitas penuh atau terdeteksi bahaya. Keunggulannya yang open-source, ringan, dan mudah dikustomisasi membuatnya ideal untuk pengelolaan sampah responsif dan berbasis data (Anwar et al., 2025). Berdasarkan kajian dan pertimbangan teknis, penulis meyakini penggunaan Node-RED dalam penelitian ini adalah solusi tepat dan berkelanjutan bagi sistem monitoring pengelolaan sampah.

Dengan demikian, dibutuhkan suatu sistem yang mengintegrasikan berbagai komponen mulai dari sensor ultrasonik, kamera ESP32-CAM, mikrokontroler ESP32, motor servo, hingga algoritma CNN, serta platform Node-RED untuk menciptakan Smart Waste Management System yang tidak hanya mampu

mendeteksi, mengklasifikasi, dan mengelola sampah secara otomatis, tetapi juga menyediakan sistem monitoring modern. Inovasi ini diharapkan dapat menjawab kebutuhan pengelolaan sampah yang cerdas, efisien, dan berkelanjutan di masa depan.

1.2 Rumusan Masalah

Pengembangan sistem tempat sampah pintar memerlukan perumusan masalah yang jelas agar proses perancangan, implementasi, dan pengujian dapat berjalan terarah. Berdasarkan latar belakang, rumusan masalah penelitian ini adalah :

1. Perancangan sistem tempat sampah pintar berbasis ESP32-CAM yang mengambil gambar sampah dan mengirimkannya ke server untuk diklasifikasikan menggunakan model Convolutional Neural Network (CNN) menjadi empat kategori, yaitu organik, anorganik, limbah medis, dan berbahaya.
2. Implementasi mekanisme pemilahan otomatis menggunakan tiga motor servo berdasarkan hasil klasifikasi, sehingga sampah diarahkan ke tong sesuai kategori.
3. Pemanfaatan sensor ultrasonik untuk mendeteksi volume sampah pada setiap tong dan penggunaan buzzer serta LED indikator sebagai peringatan lokal saat tong penuh.
4. Integrasi sistem dengan platform Node-RED untuk menampilkan hasil klasifikasi dan kapasitas tong secara real-time.

1.3 Batasan Masalah

Batasan masalah diperlukan agar penelitian tetap fokus pada lingkup yang telah ditentukan, sehingga pengembangan sistem dapat dilakukan secara efektif sesuai tujuan. Batasan masalah penelitian ini adalah :

1. Sistem hanya mengklasifikasikan sampah menjadi empat kategori, yaitu organik, anorganik, limbah medis, dan berbahaya (B3).
2. Proses klasifikasi dilakukan pada server lokal berbasis Flask menggunakan model CNN yang telah dilatih sebelumnya, bukan langsung di ESP32-CAM.
3. ESP32-CAM berfungsi untuk mengambil gambar sampah dan mengirimkannya ke server, sedangkan kontrol pemilahan dan sensor volume dijalankan oleh ESP32.
4. Sensor ultrasonik digunakan hanya untuk mendeteksi ketinggian/volume sampah dalam tong, tanpa analisis komposisi atau berat.
5. LED indikator digunakan sebagai tanda status kapasitas penuh, sedangkan buzzer berfungsi sebagai peringatan suara saat kapasitas penuh.
6. Node-RED hanya digunakan sebagai media monitoring hasil klasifikasi dan status volume tong secara real-time, tanpa mengendalikan perangkat keras.
7. Sistem tidak dilengkapi dengan mekanisme buka-tutup otomatis berbasis sensor PIR atau fitur notifikasi kesalahan pembuangan sampah.
8. Penelitian ini dilakukan dalam skala laboratorium, sehingga kondisi lingkungan uji tidak sepenuhnya merepresentasikan kondisi lapangan yang sebenarnya.

1.4 Tujuan Penelitian

Tujuan penelitian merinci hasil yang ingin dicapai melalui pengembangan sistem yang dirancang, sehingga setiap proses dapat diukur keberhasilannya.

Tujuan penelitian ini adalah :

1. Merancang dan membangun sistem tempat sampah pintar berbasis ESP32-CAM dan ESP32 yang dapat melakukan pengambilan gambar, klasifikasi, dan pemilahan sampah secara otomatis.
2. Mengimplementasikan model CNN pada server lokal untuk mengklasifikasikan sampah menjadi empat kategori sesuai citra yang dikirimkan oleh ESP32-CAM.
3. Menerapkan mekanisme pemilahan otomatis menggunakan tiga motor servo yang diarahkan sesuai hasil klasifikasi.
4. Mengintegrasikan sensor ultrasonik, LED indikator, dan buzzer untuk memantau kapasitas tong dan memberikan tanda ketika penuh.
5. Menghubungkan sistem dengan platform Node-RED untuk menampilkan informasi hasil klasifikasi dan status volume tong secara real-time.

1.5 Manfaat Penelitian

Manfaat penelitian menggambarkan kontribusi yang dapat diberikan oleh sistem yang dibangun. Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Menjadi referensi akademis dalam penerapan teknologi IoT dan pengolahan citra digital untuk sistem pemilahan sampah.
2. Memberikan contoh implementasi integrasi antara mikrokontroler, sensor, aktuator, dan platform monitoring Node-RED dalam satu sistem.

3. Membantu proses pemilahan sampah secara otomatis untuk penerapan di lingkungan publik maupun fasilitas tertentu.
4. Mempermudah pemantauan kondisi tong sampah melalui dashboard real-time serta memberikan peringatan lokal menggunakan buzzer dan LED indikator.
5. Mengurangi risiko pencemaran lingkungan dan kontaminasi akibat pencampuran sampah yang tidak sesuai kategori.

BAB II

LANDASAN TEORI

2.1 Internet of Things (IoT)

Internet of Things (IoT) adalah konsep yang menghubungkan perangkat fisik ke jaringan internet untuk saling bertukar data secara real-time. Dalam sistem otomatisasi, IoT menjadi jembatan antara sensor, mikrokontroler, dan platform monitoring yang memungkinkan sistem merespons kondisi lingkungan secara otomatis tanpa interaksi manual. Pada sistem tempat sampah pintar, IoT digunakan untuk membaca kapasitas sampah, mendeteksi gerakan, serta mengirimkan data dan peringatan ke aplikasi pemantau melalui internet (Ananda et al., 2021).

Penerapan teknologi Internet of Things (IoT) dalam pengelolaan sampah memberikan efisiensi yang tinggi karena memungkinkan pemantauan kapasitas tempat sampah secara real-time melalui koneksi internet. Sensor seperti ultrasonik dan PIR digunakan untuk mendeteksi volume sampah serta pergerakan di sekitar tong, sehingga sistem dapat merespons dengan membuka tutup tong secara otomatis dan memberikan informasi visual melalui platform monitoring berbasis web. Pemanfaatan teknologi ini mendukung otomatisasi dan transparansi dalam sistem pengelolaan sampah, serta memungkinkan pemantauan jarak jauh yang memudahkan pengawasan dan pengambilan keputusan berbasis data (Pratama et al., 2024).

2.2 Mikrokontroler ESP32 dan NodeMCU

Mikrokontroler merupakan unit kontrol utama yang berfungsi sebagai pengolah data dari berbagai sensor dan pengendali aktuator dalam sistem elektronik

berbasis IoT. Dalam pengembangan tempat sampah pintar, mikrokontroler yang banyak digunakan adalah NodeMCU ESP32, karena memiliki keunggulan berupa konektivitas Wi-Fi bawaan, jumlah pin yang cukup banyak, serta konsumsi daya yang efisien. Perangkat ini dapat langsung menghubungkan sensor ke jaringan internet tanpa modul tambahan, sehingga cocok untuk sistem monitoring yang membutuhkan koneksi real-time (Ariyanti et al., 2021).



Gambar 2. 1 Mikrokontroler ESP32 (Universitas Raharja, 2021)

Dalam implementasinya, NodeMCU ESP32 memegang peranan sebagai pusat kendali yang mengintegrasikan berbagai sensor untuk membentuk sistem monitoring tempat sampah yang otomatis dan real-time. Sensor ultrasonik digunakan untuk mengukur ketinggian sampah sebagai indikator kapasitas, sementara modul Wi-Fi internal ESP32 memungkinkan pengiriman data ke aplikasi pemantauan yang terhubung melalui jaringan internet. Ketika kapasitas sampah mencapai ambang batas tertentu, sistem secara otomatis mengirimkan notifikasi ke aplikasi pengguna, sehingga mempercepat respons petugas dalam pengelolaan sampah. Kemampuan ESP32 dalam menangani pengolahan data sensor sekaligus konektivitas internet menjadikannya komponen utama yang sangat andal dalam pengembangan tempat sampah pintar berbasis IoT (Muliadi et al., 2020).

2.3 Sensor Ultrasonik

Sensor ultrasonik merupakan komponen yang bekerja dengan memancarkan gelombang suara berfrekuensi tinggi (di atas 20 kHz) untuk mendeteksi keberadaan atau jarak suatu objek berdasarkan waktu pantulan gelombang yang diterima kembali oleh sensor. Dalam sistem tempat sampah pintar, sensor ini digunakan untuk mengukur ketinggian sampah di dalam wadah guna mengetahui tingkat kepenuhan tempat sampah. Ketika jarak antara sensor dan permukaan sampah berada di bawah ambang tertentu, sistem akan mengaktifkan peringatan berupa suara dan notifikasi digital melalui platform pemantauan. Dengan memanfaatkan prinsip kerja dua unit pemancar dan penerima, sensor ultrasonik dapat memberikan informasi akurat secara real-time kepada mikrokontroler yang selanjutnya diteruskan ke aplikasi antarmuka seperti Telegram atau Blynk (Kristanti et al., 2022).



Gambar 2. 2 Sensor Ultrasonik (Arduino Indonesia, 2022)

2.4 Kamera ESP32-CAM

ESP32-CAM merupakan modul mikrokontroler yang dilengkapi dengan kamera terintegrasi dan konektivitas Wi-Fi, sehingga sangat cocok digunakan dalam sistem keamanan berbasis IoT. Pada penerapannya dalam tempat sampah pintar, ESP32-CAM berfungsi untuk mengambil gambar objek atau individu yang

terdeteksi oleh sensor gerak sebagai bagian dari sistem keamanan. Setelah gambar diambil, data dikirimkan secara real-time ke aplikasi pesan seperti Telegram untuk memberikan notifikasi visual kepada pengguna. Kemampuan ESP32-CAM dalam menggabungkan fungsi pengawasan visual dan transmisi data menjadikannya komponen penting dalam sistem tempat sampah pintar yang membutuhkan fitur keamanan tambahan berbasis gambar (Wisnu et al., 2021).



Gambar 2. 3 Kamera ESP32-CAM (CNX Software, 2018)

2.5 Motor Servo

Motor servo merupakan aktuator yang berfungsi untuk menggerakkan atau mengontrol posisi objek secara presisi melalui sudut tertentu. Dalam sistem tempat sampah pintar, komponen ini berperan sebagai penggerak otomatis yang membuka dan menutup tutup tempat sampah berdasarkan perintah dari mikrokontroler setelah menerima sinyal dari sensor. Ketika sensor mendeteksi adanya objek atau sampah pada jarak tertentu, sistem akan mengaktifkan motor servo untuk membuka tutup, kemudian menutup kembali setelah selang waktu tertentu secara otomatis. Penggunaan motor servo memberikan efisiensi yang tinggi dalam proses buka-tutup tempat sampah tanpa sentuhan, sekaligus meningkatkan aspek kebersihan dan kenyamanan pengguna (Ramdan et al., 2024).



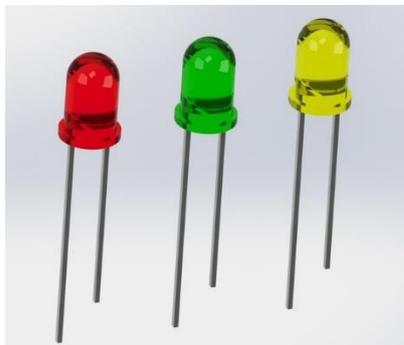
Gambar 2. 4 Motor Servo (Arduino Indonesia, 2022)

2.6 Buzzer dan LED

Buzzer dan LED digunakan sebagai alat peringatan lokal pada tempat sampah pintar. Buzzer mengeluarkan suara saat sampah penuh, sedangkan LED menyala sebagai indikator visual. Kedua komponen ini bekerja bersamaan dengan sensor dan mikrokontroler untuk memberikan respon cepat terhadap kondisi kritis, terutama saat pemantauan jarak jauh tidak memungkinkan. Implementasi ini telah terbukti efektif dalam meningkatkan efisiensi pengelolaan sampah melalui sistem notifikasi langsung (Fatmawati et al., 2020).



Gambar 2. 5 Buzzer (Indobot, 2022)



Gambar 2. 6 LED (Setiyo, 2023)

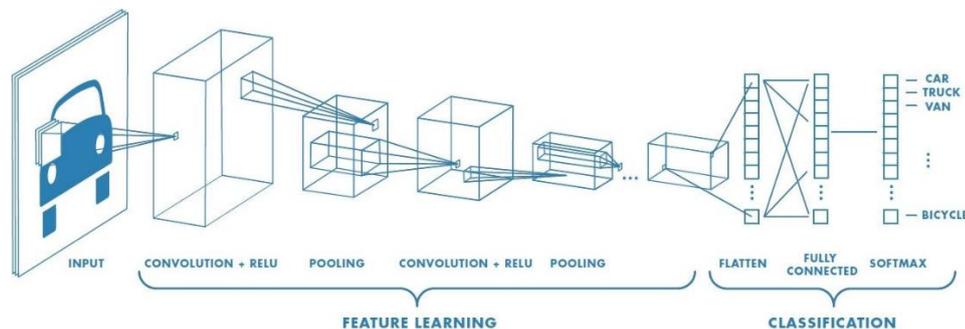
2.7 Machine Learning

Machine Learning (ML) merupakan cabang kecerdasan buatan yang memungkinkan komputer mempelajari pola dari data secara otomatis untuk menghasilkan prediksi atau klasifikasi. Teknologi ini membangun model matematis yang dilatih menggunakan dataset tertentu, sehingga mampu mengenali hubungan kompleks di dalam data tanpa harus diprogram secara eksplisit untuk setiap aturan. Dalam pengelolaan sampah, ML dapat digunakan untuk mengklasifikasi jenis sampah berdasarkan citra visual, sehingga proses pemilahan dapat berjalan otomatis dengan akurasi tinggi. Salah satu metode populer adalah Convolutional Neural Network (CNN), yang mengekstraksi fitur penting dari gambar melalui lapisan konvolusi (Zaman et al., 2025).

Salah satu arsitektur CNN yang efisien untuk perangkat dengan sumber daya terbatas adalah MobileNetV2. Arsitektur ini dirancang ringan dan cepat, namun tetap memiliki performa baik dalam pengenalan objek. Penelitian oleh Zaman et al. (2025) memanfaatkan MobileNetV2 dengan pendekatan transfer learning untuk mengklasifikasi sampah organik dan anorganik pada perangkat ESP32-CAM. Proses ini memungkinkan sistem berjalan optimal meskipun memiliki keterbatasan memori dan daya komputasi. Integrasi antara sensor, kamera, dan ML membuat sistem mampu mengambil gambar, mengklasifikasi, dan menggerakkan aktuator seperti motor servo untuk memisahkan sampah secara otomatis. Pendekatan ini dapat meningkatkan efisiensi serta mengurangi kesalahan manusia dalam pengelolaan limbah (Zaman et al., 2025).

2.8 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) merupakan salah satu arsitektur deep learning yang sangat efektif dalam tugas klasifikasi citra, termasuk dalam pengelolaan sampah otomatis. CNN bekerja dengan mengekstraksi fitur visual seperti warna, bentuk, dan tekstur dari gambar untuk mengenali pola dan objek. Dalam konteks Smart Waste Management, CNN memungkinkan sistem mengenali dan membedakan jenis-jenis sampah seperti organik, plastik, kertas, atau logam secara otomatis, tanpa perlu intervensi manual. Keunggulan ini menjadikan CNN sebagai komponen penting dalam menciptakan sistem tempat sampah pintar yang responsif dan efisien terhadap klasifikasi sampah (Pieters, 2025).



Gambar 2. 7 Convolutional Neural Network (Sena, 2017)

Penelitian yang dilakukan oleh Setiana (2024) menunjukkan bahwa penerapan algoritma Convolutional Neural Network (CNN) dengan arsitektur MobileNet pada sistem pemilah sampah dapat menghasilkan akurasi validasi sebesar 90,53% dalam mengklasifikasikan gambar sampah menjadi tiga kategori: organik, anorganik, dan botol plastik. Pelatihan model dilakukan menggunakan dataset citra dari berbagai sumber dan diolah di Google Colaboratory, kemudian diimplementasikan pada Raspberry Pi dengan model yang telah dikonversi menggunakan TensorFlow Lite. Keunggulan CNN dalam mengekstraksi fitur visual pada citra sampah memungkinkan sistem mengenali objek secara efisien dan

responsif, meskipun pada kondisi sumber daya terbatas. Dengan demikian, integrasi CNN dan MobileNet terbukti efektif dalam membangun sistem klasifikasi sampah otomatis berbasis IoT yang dapat mendukung pengelolaan lingkungan secara berkelanjutan (Setiana, 2024).

2.9 Klasifikasi Jenis Sampah

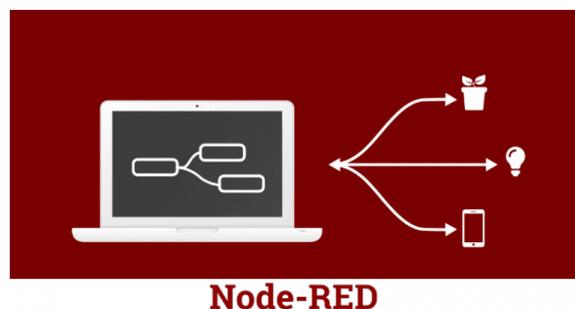
Klasifikasi sampah merupakan proses penting dalam pengelolaan limbah, terutama untuk meminimalkan dampak lingkungan dan risiko kesehatan. Secara umum, jenis sampah diklasifikasikan menjadi empat kategori utama: sampah organik (seperti sisa makanan dan daun), sampah anorganik (seperti plastik, kaca, dan logam), limbah medis (seperti masker bekas, perban, dan jarum suntik), serta sampah berbahaya atau B3 (bahan beracun dan berbahaya seperti limbah kimia). Penelitian di Rumah Sakit Budi Agung menunjukkan bahwa meskipun pemilahan limbah medis dan non-medis telah dilakukan, pemisahan antara sampah organik dan anorganik masih kurang optimal. Padahal, pencampuran limbah—terutama yang bersifat berbahaya atau infeksius—berpotensi meningkatkan risiko kontaminasi dan memperberat proses pengelolaan limbah (Kotika et al., 2023).

Untuk mendukung proses klasifikasi jenis sampah secara otomatis, penelitian oleh Anggara et al. (2025) mengembangkan sistem deteksi berbasis algoritma YOLOv8 yang mampu mengidentifikasi sampah dari citra gambar ke dalam beberapa kategori, seperti organik, plastik, kertas, kardus, logam, dan gelas. Model ini dilatih menggunakan dataset beranotasi dan divalidasi dengan metode mAP yang menghasilkan nilai akurasi sebesar 0.905. Pendekatan ini menunjukkan bahwa klasifikasi jenis sampah dapat dilakukan secara efisien dan akurat tanpa campur tangan manusia, sehingga mempermudah proses pemilahan sejak awal.

Meskipun demikian, tantangan masih ditemukan pada kategori organik yang memiliki bentuk dan warna yang sangat beragam, sehingga mempengaruhi ketepatan klasifikasi. Dengan demikian, metode klasifikasi otomatis seperti ini berperan penting dalam mendukung sistem pengelolaan limbah yang lebih terstruktur dan berkelanjutan, serta dapat diintegrasikan ke dalam solusi berbasis IoT (Anggara et al., 2025).

2.10 Node-RED

Node-RED adalah platform pemrograman visual berbasis flow yang dikembangkan untuk memudahkan integrasi perangkat keras, API, dan layanan online secara modular. Dalam konteks Internet of Things (IoT), Node-RED berfungsi sebagai middleware yang menjembatani komunikasi antara perangkat IoT dengan pengguna melalui antarmuka dashboard yang interaktif. Berdasarkan penelitian Anwar, Firmansyah, dan Asra (2025), Node-RED digunakan sebagai pusat monitoring data dari sistem tempat sampah pintar berbasis protokol MQTT. Data yang dikirim oleh mikrokontroler, seperti status kapasitas bin, diproses dan divisualisasikan secara real-time. Selain itu, Node-RED mendukung penyimpanan data historis ke basis data untuk analisis tren, yang membantu pengelola kebersihan dalam merencanakan rute pengangkutan dan frekuensi pengosongan.



Gambar 2. 8 Node-RED (Parker Enfield, 2017)

Keunggulan Node-RED meliputi sifatnya yang open-source, ringan, serta mudah dikustomisasi sesuai kebutuhan sistem. Platform ini juga memiliki kompatibilitas tinggi dengan protokol komunikasi populer seperti MQTT, yang memungkinkan komunikasi publish-subscribe dengan latensi rendah dan efisiensi bandwidth yang lebih baik dibanding HTTP (Anwar et al., 2025). Dengan kemampuannya menerima data dari broker MQTT, menampilkannya secara langsung pada dashboard, serta mengirim notifikasi berbasis kondisi tertentu, Node-RED menjadi solusi yang relevan untuk implementasi sistem monitoring pengelolaan sampah di berbagai skala, mulai dari fasilitas publik hingga integrasi ke sistem smart city.

2.11 Flask

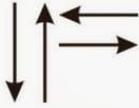
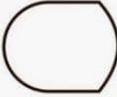
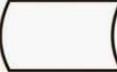
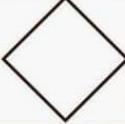
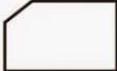
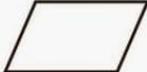
Flask merupakan salah satu framework web berbasis bahasa pemrograman Python yang bersifat open-source dan ringan. Framework ini menerapkan konsep microframework, yang berarti hanya menyediakan komponen inti untuk membangun aplikasi web, sementara fitur tambahan seperti autentikasi, manajemen basis data, dan validasi formulir dapat ditambahkan melalui ekstensi sesuai kebutuhan. Keunggulan utama Flask adalah fleksibilitas, kemudahan penggunaan, dan dokumentasi yang lengkap, sehingga sangat populer di kalangan pengembang yang membutuhkan solusi cepat dan efisien. Dalam pengembangan sistem berbasis IoT, Flask sering dimanfaatkan untuk membangun server yang dapat mengelola komunikasi data antara perangkat dan antarmuka pengguna berbasis web (Fazryansah et al., 2024).

Dalam konteks penelitian ini, Flask digunakan sebagai server lokal yang menerima data citra dari ESP32-CAM, memprosesnya menggunakan model

Convolutional Neural Network (CNN), dan mengirimkan hasil klasifikasi kembali ke mikrokontroler untuk mengendalikan aktuator seperti motor servo. Integrasi Flask dengan teknologi machine learning memudahkan proses deployment model, sehingga perangkat IoT dapat memanfaatkan hasil klasifikasi secara real-time. Selain itu, karena Flask mendukung protokol HTTP dan dapat diintegrasikan dengan platform monitoring seperti Node-RED, sistem ini mampu memberikan visualisasi data dan kontrol jarak jauh melalui dashboard. Pendekatan ini tidak hanya mempermudah pengelolaan data, tetapi juga memungkinkan sistem berjalan secara modular dan mudah dikembangkan di masa depan (Fazryansah et al., 2024).

2.12 Flowchart

Flowchart atau diagram alur adalah representasi visual dari urutan proses yang menggambarkan logika kerja sistem. Dalam penelitian ini, flowchart berperan penting dalam mendeskripsikan alur kerja sistem Smart Waste Management yang dibangun berbasis teknologi Internet of Things (IoT) dan Convolutional Neural Network (CNN), mulai dari deteksi kehadiran pengguna hingga klasifikasi jenis sampah dan notifikasi digital (Kristanti et al., 2022).

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/prosedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 2. 9 Flowchart Sistem (Itbox.id, 2023)

2.13 Penelitian Terkait

Beberapa penelitian sebelumnya telah dilakukan dalam pengembangan tempat sampah pintar berbasis IoT maupun klasifikasi sampah otomatis menggunakan CNN. Setiap penelitian memiliki pendekatan dan fokus berbeda, baik dari sisi teknologi, metode, maupun tujuan sistem. Tabel berikut menyajikan ringkasan dari penelitian-penelitian tersebut untuk menunjukkan posisi dan keunikan penelitian ini dibandingkan dengan yang telah ada.

Tabel 2. 1 Penelitian Terkait

No	Nama Tahun dan Judul Penelitian	Metode Penelitian	Hasil Penelitian	Perbedaan Penelitian
1.	Ramadhina Trie Ananda et al., 2021. "Sistem Tempat Sampah Pintar Berbasis IoT Menggunakan Aplikasi Blynk"	Eksperimen; menggunakan NodeMCU 8266, sensor ultrasonik, sensor MQ-2, aplikasi Blynk	Penutup terbuka otomatis saat objek terdeteksi; notifikasi penuh & asap dikirim ke Blynk	Tidak menggunakan kamera atau klasifikasi sampah berbasis citra/CNN
2.	Ragel Trisudarmo et al., 2024. "Tempat Sampah Pintar Berbasis IoT dengan Menggunakan Arduino Uno R3"	Metode Waterfall, implementasi Arduino Uno, sensor HC-SR04, servo	Tutup terbuka otomatis saat objek terdeteksi; meningkatkan kesadaran membuang sampah	Tidak menggunakan notifikasi atau fitur klasifikasi jenis sampah
3.	Gracela Claudia Kotika et al., 2023. "Sistem Pengelolaan Limbah Medis dan Non Medis di RS Budi Agung"	Metode kualitatif, studi kasus	Sistem pemilahan dan pengangkutan sampah dilakukan manual, sudah ada pengomposan	Fokus pada manajemen rumah sakit, tanpa teknologi otomatisasi/IoT/CNN

4.	Renaldi Irfan Firdaus et al., 2025. "Implementasi CNN dalam Sistem Otomatis Pemilahan Sampah Infeksius Berbasis Citra Digital"	Prototyping, CNN, ESP32-CAM, Edge Impulse	CNN berhasil klasifikasi limbah infeksius berbasis citra dengan akurasi tinggi	Fokus pada limbah medis; belum menggabungkan sistem kontrol fisik dan IoT
5.	Ahmad Teguh Ashari et al., 2024. "Sistem Otomatisasi Tempat Sampah dengan Sensor Ultrasonik dan PIR Berbasis Mikrokontroler"	Eksperimen; menggunakan Wemos D1 R32, sensor PIR & ultrasonik, web monitoring	Deteksi kapasitas 95%, deteksi manusia 98%, buka tutup otomatis, pemantauan web	Tidak dilengkapi dengan kamera/CNN atau klasifikasi jenis sampah
6.	Luntungan Stephen Pieters, 2025. "Pengembangan Sistem Klasifikasi Sampah Otomatis Menggunakan CNN untuk Smart Waste Management"	Deep Learning, CNN, validasi silang, dataset WasteNet	Akurasi CNN 94.86%, efektif klasifikasi recyclable, namun kesulitan pada organik	Fokus pada klasifikasi citra tanpa integrasi dengan IoT atau sistem fisik tempat sampah
7.	Wisnu C.M. et al., 2021. "Prototipe Tempat Sampah Pintar dengan Sistem Keamanan dan Telegram"	Prototyping, ESP32-CAM, sensor PIR & ultrasonik, Telegram	Tempat sampah terbuka otomatis, kamera kirim foto ke Telegram jika ada gerakan	Tidak menggunakan klasifikasi gambar atau CNN

8.	Nike Kristanti et al., 2022. “Penerapan Sensor Ultrasonik pada Kotak Sampah Otomatis dengan Telegram dan Alarm Suara”	Prototyping, NodeMCU, sensor ultrasonik, Telegram	Beri notifikasi saat sampah hampir penuh, status via Telegram, alarm berbunyi	Tidak memiliki kamera atau sistem klasifikasi jenis sampah
9.	Dyah Ariyanti et al., 2023. “Rancang Bangun Tempat Sampah Pintar Berbasis IoT dengan Sistem Monitoring”	IoT dengan ESP32, sensor PIR, berat, ultrasonik; aplikasi Blynk	Monitoring sampah via Blynk; sensor baca berat & volume, buka otomatis	Tidak menggunakan sistem klasifikasi gambar atau CNN untuk identifikasi jenis sampah
10.	Jerry Anggara et al., 2025. “Implementasi Object Detection dalam Klasifikasi Sampah untuk Meningkatkan Efisiensi Pengelolaan Limbah”	Eksperimen dengan algoritma YOLOv8, pemrograman Python, dan Flask	Sistem dapat mendeteksi dan mengklasifikasi 6 jenis sampah dengan akurasi mAP 0.905 serta menampilkan hasil deteksi melalui web	Menggunakan object detection berbasis gambar, tidak dilengkapi mekanisme fisik seperti buka-tutup tutup tong
11.	Muliadi et al., 2020. “Pengembangan Tempat Sampah Pintar Menggunakan ESP32”	R&D, perancangan hardware dan software dengan ESP32 dan sensor ultrasonik	Tempat sampah memberi notifikasi penuh ke aplikasi smartphone secara real-time	Tidak memiliki fitur buka-tutup otomatis atau klasifikasi jenis sampah
12.	Aldin Rizqi Pratama et al., 2024. “Monitoring Tempat Sampah Pintar Berbasis	Eksperimen menggunakan Wemos D1 R32, sensor ultrasonik & PIR, servo,	Tutup tong terbuka otomatis saat pengguna terdeteksi, monitoring kapasitas tong	Tidak melakukan klasifikasi sampah berdasarkan jenis atau pengolahan citra

	Internet of Things (IoT)	web monitoring	secara real-time via website	
--	--------------------------	----------------	------------------------------	--

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Penelitian ini menggunakan metode observasi deskriptif, yaitu metode yang bertujuan menggambarkan dan mengamati secara langsung kinerja sistem yang dirancang tanpa melakukan manipulasi terhadap kondisi yang ada. Pada penelitian ini, objek yang diamati adalah kinerja dan respons sistem Smart Waste Management berbasis IoT, dengan arsitektur ESP32-CAM sebagai pengambil gambar dan pengendali aktuator, serta server Flask yang menjalankan model CNN untuk klasifikasi jenis sampah.

Metode observasi digunakan untuk melihat bagaimana sistem bekerja di kondisi sebenarnya, mulai dari proses pengambilan gambar oleh ESP32-CAM, pengiriman data ke server Flask untuk diklasifikasikan menjadi Organik, Non-Organik, Limbah Medis, atau Limbah Berbahaya, hingga penerimaan hasil klasifikasi kembali oleh ESP32-CAM untuk menggerakkan tiga buah servo sesuai kategori sampah. Observasi juga dilakukan pada fungsi sensor ultrasonik untuk mendeteksi kapasitas tong sampah, LED sebagai indikator penuh, serta pengiriman data ke Node-RED untuk pemantauan real-time.

Melalui metode ini, peneliti memperoleh data yang akurat dan relevan terkait efektivitas, akurasi klasifikasi, dan keandalan sistem dalam skenario penggunaan nyata, serta dapat mengidentifikasi kelebihan dan kekurangan dari sistem yang dirancang.

3.1.1 Karakteristik Metode Observasi

Karakteristik dari metode observasi pada penelitian ini adalah sebagai berikut:

1. Bersifat langsung

Observasi dilakukan secara langsung terhadap prototipe sistem tempat sampah pintar yang telah dirancang, untuk melihat respon sistem dalam memproses gambar, mengklasifikasikan sampah, menggerakkan servo, serta memberikan notifikasi indikator penuh.

2. Deskriptif

Penelitian ini tidak menguji hipotesis, melainkan menggambarkan performa sistem berdasarkan pengamatan selama pengujian, termasuk waktu respon klasifikasi, ketepatan arah servo, dan akurasi sensor ultrasonik.

3. Berbasis teknologi IoT

Pengamatan dilakukan dengan bantuan perangkat keras ESP32-CAM, sensor ultrasonik, dan servo motor, serta perangkat lunak Flask server untuk CNN dan Node-RED sebagai platform monitoring.

4. Fokus pada pengujian system

Fokus utama observasi adalah mengevaluasi apakah sistem dapat menjalankan fungsinya dengan baik, mulai dari pengambilan gambar, pengiriman dan pemrosesan data klasifikasi di server, pergerakan aktuator servo sesuai hasil klasifikasi, pendeteksian kapasitas tong, hingga pengiriman informasi ke dashboard Node-RED secara real-time.

3.2 Lokasi dan Waktu Penelitian

Penelitian ini dilaksanakan di dua lokasi utama, yaitu lokasi perakitan dan pengembangan sistem, serta lokasi pengujian prototipe secara langsung.

3.2.1 Tempat Pengembangan Sistem

Proses perancangan perangkat keras dan pengembangan perangkat lunak dilakukan di ruang kerja yang telah disiapkan oleh peneliti. Pada tahap ini dilakukan :

1. Perakitan rangkaian ESP32-CAM, tiga buah servo, empat sensor ultrasonik, dan LED indikator.
2. Pemrograman ESP32-CAM untuk mengambil gambar, mengirim data ke server Flask, mengendalikan servo, membaca data sensor, dan mengirim status ke Node-RED.
3. Pengembangan server Flask untuk menjalankan model CNN dan melakukan klasifikasi citra sampah.
4. Integrasi dashboard Node-RED untuk monitoring hasil klasifikasi dan kapasitas tong sampah.
5. Perangkat pendukung yang digunakan antara lain laptop dengan Python dan Arduino IDE, jaringan internet yang stabil, serta komponen elektronik yang mendukung sistem IoT.

3.2.2 Tempat Pengujian Prototipe

Pengujian dilakukan di lokasi yang menyerupai kondisi lapangan, seperti halaman rumah atau area terbuka yang memungkinkan interaksi langsung dengan prototipe. Pada tahap ini, sistem diuji dengan berbagai jenis sampah sesuai kategori (Organik, Non-Organik, Limbah Medis, Limbah Berbahaya) untuk mengevaluasi :

1. Akurasi klasifikasi citra oleh model CNN di server Flask.
2. Kecepatan respon sistem dari proses pengambilan gambar hingga pergerakan servo.

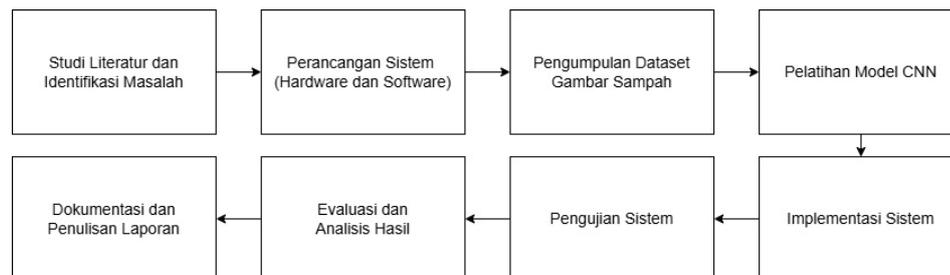
3. Akurasi pembacaan sensor ultrasonik dalam mendeteksi kapasitas tong.
4. Kesesuaian data yang ditampilkan di dashboard Node-RED secara real-time.

3.2.3 Waktu Penelitian

Waktu pelaksanaan penelitian ini direncanakan berlangsung selama kurang lebih tiga bulan, yaitu mulai dari bulan Mei 2025 hingga Agustus 2025. Selama rentang waktu tersebut, kegiatan penelitian mencakup proses perancangan alat, pengembangan perangkat lunak, pelatihan model klasifikasi citra, integrasi sistem IoT, serta pengujian dan evaluasi kinerja prototipe secara menyeluruh.

3.3 Alur Penelitian

Alur penelitian dalam pembangunan sistem Smart Waste Management ini dilaksanakan secara bertahap mulai dari studi literatur hingga penyusunan laporan akhir. Setiap tahap dirancang untuk memastikan bahwa sistem yang dibangun dapat bekerja secara teknis, akurat dalam klasifikasi, dan efisien dalam pemilahan sampah. Adapun tahapan-tahapan penelitian adalah sebagai berikut :



Gambar 3. 1 Diagram Alur Penelitian

3.3.1 Studi Literatur dan Identifikasi Masalah

Tahap ini bertujuan memperoleh pemahaman mendalam tentang teknologi yang akan digunakan dan permasalahan yang hendak diselesaikan. Peneliti mempelajari referensi dari jurnal, artikel, dan penelitian terdahulu, kemudian menganalisis

kekurangan sistem yang sudah ada untuk dirumuskan sebagai peluang pengembangan. Rincian kegiatan :

1. Mempelajari konsep Internet of Things (IoT) dalam pengelolaan sampah.
2. Mengkaji metode klasifikasi citra digital menggunakan Convolutional Neural Network (CNN).
3. Menelaah fungsi ESP32-CAM sebagai pengambil gambar dan pengendali aktuator.
4. Menganalisis penggunaan servo motor untuk mekanisme pemilah dan sensor ultrasonik untuk deteksi kapasitas tong.
5. Mengkaji penggunaan Node-RED sebagai platform monitoring real-time.

3.3.2 Perancangan Sistem (Hardware dan Software)

Tahap perancangan sistem merupakan inti dari proses pengembangan, di mana peneliti merancang secara menyeluruh arsitektur perangkat keras (hardware) dan perangkat lunak (software) agar sistem dapat berfungsi sesuai tujuan penelitian. Perancangan ini melibatkan pemilihan komponen, integrasi antarperangkat, serta penyusunan alur komunikasi data antara sensor, aktuator, dan server pengolah data. Pada tahap ini, seluruh komponen yang dipilih telah disesuaikan dengan kebutuhan sistem Smart Waste Management yang memiliki empat tong sampah dengan tiga mekanisme pemilah berbasis servo. Pada sisi perangkat keras, sistem terdiri dari:

1. ESP32-CAM sebagai unit utama untuk pengambilan gambar sampah, pengendali servo, dan penghubung komunikasi dengan server.
2. Tiga buah servo motor yang berperan sebagai mekanisme pemilah :

- a. Servo 1 berfungsi menentukan jalur awal pembuangan, mengarahkan sampah ke Servo 2 (untuk kategori organik dan non-organik) atau ke Servo 3 (untuk kategori limbah medis dan limbah berbahaya).
 - b. Servo 2 bertugas memilah sampah menjadi organik (kanan) dan non-organik (kiri).
 - c. Servo 3 bertugas memilah sampah menjadi limbah medis (kanan) dan limbah berbahaya (kiri).
3. Empat sensor ultrasonik yang dipasang di masing-masing tong untuk mendeteksi kapasitas dan memberikan informasi apakah tong sudah penuh.
 4. LED indikator yang menyala sebagai tanda peringatan ketika tong mencapai kapasitas maksimum.

Pada sisi perangkat lunak, proses perancangan meliputi :

1. Pemrograman ESP32-CAM menggunakan Arduino IDE untuk mengatur :
 - a. Pengambilan gambar secara otomatis ketika sampah masuk.
 - b. Pengiriman gambar ke server Flask menggunakan protokol HTTP.
 - c. Penerimaan hasil klasifikasi dari server dan menggerakkan servo sesuai kategori.
 - d. Pembacaan data dari sensor ultrasonik dan pengiriman status ke Node-RED melalui protokol MQTT.
2. Pengembangan server Flask di komputer atau laptop yang berfungsi :
 - a. Menjalankan model CNN hasil pelatihan.
 - b. Menerima gambar dari ESP32-CAM.
 - c. Mengklasifikasikan gambar menjadi empat kategori sampah.

- d. Mengirim hasil klasifikasi kembali ke ESP32-CAM untuk menggerakkan mekanisme pemilah.
3. Penyusunan dashboard Node-RED untuk menampilkan :
 - a. Hasil klasifikasi setiap sampah yang masuk.
 - b. Kondisi kapasitas masing-masing tong sampah.
 - c. Peringatan visual jika tong sudah penuh.

Tahap perancangan ini menjadi dasar yang memastikan seluruh komponen dapat berinteraksi secara sinkron, sehingga ketika tahap implementasi dilakukan, sistem mampu bekerja secara otomatis mulai dari pengambilan gambar, pengolahan data, pemilahan sampah, hingga pemantauan kapasitas tong secara real-time.

3.3.3 Pengumpulan Dataset Gambar Sampah

Tahap ini bertujuan untuk menyediakan data latih yang memadai sehingga model CNN yang dibangun dapat mengenali berbagai jenis sampah dengan baik. Dataset yang digunakan harus mencerminkan kondisi nyata di lapangan, sehingga variasi bentuk, warna, dan pencahayaan tetap terwakili. Peneliti memutuskan menggunakan dataset dari sumber daring karena ketersediaannya yang sudah terlabeli dan jumlahnya mencukupi untuk pelatihan awal. Langkah yang dilakukan meliputi :

1. Mengambil dataset dari Kaggle Waste Classification Data dengan kategori :
 - a. Organik
 - b. Non-Organik
 - c. Limbah Medis
 - d. Limbah Berbahaya (B3)

2. Melakukan preprocessing untuk meningkatkan kualitas data dan keseragaman format :
 - a. Mengubah ukuran citra (resize) sesuai input model CNN.
 - b. Melakukan normalisasi warna agar nilai piksel berada pada rentang yang sesuai untuk pelatihan.
 - c. Melakukan augmentasi data seperti rotasi, flipping, dan zoom untuk memperkaya variasi data latih.
3. Membagi dataset menjadi tiga bagian :
 - a. Training set untuk proses pembelajaran model.
 - b. Validation set untuk memantau kinerja model saat pelatihan.
 - c. Testing set untuk mengukur akurasi model pada data yang belum pernah dilihat.

Tahapan ini sangat penting karena kualitas dan variasi dataset sangat mempengaruhi akurasi klasifikasi yang akan dicapai oleh model CNN.

3.3.4 Pelatihan Model CNN

Setelah dataset siap, model CNN mulai dirancang dan dilatih untuk mengenali keempat kategori sampah tersebut. Desain arsitektur dibuat sedemikian rupa agar mampu mengekstraksi fitur penting dari gambar dan mengklasifikasikannya dengan tingkat akurasi tinggi. Pelatihan dilakukan pada lingkungan komputasi awan untuk memanfaatkan sumber daya GPU, sehingga proses menjadi lebih cepat.

Tahapan pelatihan meliputi :

1. Mendesain arsitektur CNN dengan kombinasi Conv2D, Pooling Layer, dan Fully Connected Layer.
2. Melatih model menggunakan TensorFlow/Keras di Google Colab.

3. Menggunakan metrik evaluasi seperti akurasi, presisi, dan recall untuk menilai performa model.
4. Menyimpan model terbaik dalam format .h5 untuk diintegrasikan ke server Flask.

Hasil dari tahap ini adalah sebuah model CNN yang siap digunakan untuk mengklasifikasikan gambar sampah secara otomatis.

3.3.5 Implementasi Sistem

Tahap implementasi menggabungkan seluruh komponen perangkat keras dan perangkat lunak menjadi satu kesatuan sistem yang berfungsi. ESP32-CAM diprogram untuk mengambil gambar setiap kali ada sampah masuk, mengirimkan gambar tersebut ke server Flask, dan menerima hasil klasifikasinya. Berdasarkan hasil klasifikasi, servo motor bergerak untuk mengarahkan sampah ke tong yang sesuai, sedangkan sensor ultrasonik memantau kapasitas masing-masing tong. Alur implementasi sistem:

1. ESP32-CAM mengambil gambar dan mengirimnya ke server Flask.
2. Server Flask menjalankan model CNN untuk menentukan kategori sampah.
3. Hasil klasifikasi dikirim kembali ke ESP32-CAM.
4. Servo bergerak sesuai kategori hasil klasifikasi.
5. Sensor ultrasonik memantau kapasitas tong dan LED indikator menyala jika penuh.
6. Semua data status dikirim ke dashboard Node-RED untuk pemantauan real-time.

3.3.6 Pengujian Sistem

Pengujian dilakukan untuk memastikan seluruh komponen dan fungsi sistem berjalan sesuai perancangan. Lingkungan uji dibuat menyerupai kondisi nyata agar respon sistem dapat diamati secara komprehensif. Aspek yang diuji mencakup :

1. Akurasi klasifikasi CNN terhadap empat kategori sampah.
2. Kecepatan respon sistem dari pengambilan gambar hingga pergerakan servo.
3. Akurasi sensor ultrasonik dalam mendeteksi kondisi penuh.
4. Keandalan pengiriman data ke Node-RED dalam berbagai kondisi jaringan.

Hasil pengujian ini menjadi dasar evaluasi untuk menentukan apakah sistem sudah memenuhi tujuan penelitian.

3.3.7 Evaluasi dan Analisis Hasil

Evaluasi dilakukan dengan membandingkan hasil pengujian dengan target kinerja yang telah ditetapkan. Analisis difokuskan pada tingkat akurasi model CNN, kecepatan dan efisiensi mekanisme pemilah servo, keakuratan pembacaan sensor ultrasonik, serta stabilitas sistem dalam operasi berkelanjutan.

Hasil evaluasi ini juga menjadi acuan untuk mengidentifikasi kelebihan dan kekurangan sistem, sehingga dapat dirumuskan langkah perbaikan pada penelitian berikutnya.

3.3.8 Dokumentasi dan Penulisan Laporan

Tahap akhir adalah mendokumentasikan seluruh proses penelitian dari perancangan hingga evaluasi. Dokumentasi ini meliputi foto dan diagram prototipe, diagram alur sistem, kode program ESP32-CAM dan server Flask, serta data hasil

pengujian yang disajikan dalam bentuk tabel atau grafik. Semua dokumentasi disusun secara sistematis dalam bentuk skripsi sesuai kaidah penulisan ilmiah, sehingga dapat menjadi referensi bagi penelitian selanjutnya.

3.4 Alat dan Bahan Penelitian

Penelitian ini memerlukan kombinasi perangkat keras, perangkat lunak, dan bahan pendukung untuk membangun serta menguji sistem Smart Waste Management berbasis IoT dan CNN. Seluruh alat dan bahan dipilih berdasarkan fungsionalitas, kompatibilitas, serta efisiensinya dalam mendukung kinerja sistem.

Secara umum, alat dan bahan yang digunakan terbagi ke dalam tiga kelompok utama, yaitu perangkat elektronik dan keras, unit fisik tong sampah, serta perangkat lunak pendukung.

3.4.1 Alat Elektronik dan Perangkat Keras

1. ESP32

Mikrokontroler utama yang mengendalikan keseluruhan sistem, mulai dari menerima hasil klasifikasi dari server, mengatur pergerakan tiga motor servo, membaca data dari empat sensor ultrasonik, hingga mengirim data status ke Node-RED melalui jaringan Wi-Fi.

2. ESP32-CAM

Modul ESP32 dengan kamera terintegrasi yang bertugas menangkap gambar sampah, mengirimkan gambar ke server Flask untuk diproses oleh model CNN, dan menerima hasil klasifikasi untuk menentukan mekanisme pemilahan.

3. Sensor Ultrasonik HC-SR04 (4 unit)

Digunakan untuk mengukur ketinggian tumpukan sampah di masing-masing tong sehingga sistem dapat mengetahui kapasitas tong dan memberi peringatan saat penuh.

4. Motor Servo (3 unit)

Berfungsi sebagai aktuator mekanisme pemilah sampah:

- a. Servo 1 : Mengarahkan sampah ke jalur Servo 2 (organik & non-organik) atau Servo 3 (limbah medis & B3).
- b. Servo 2 : Memilah organik (kanan) dan non-organik (kiri).
- c. Servo 3 : Memilah limbah medis (kanan) dan limbah berbahaya/B3 (kiri).

5. LED Indikator

Memberikan sinyal visual saat tong penuh atau terjadi kondisi khusus yang memerlukan perhatian.

6. Adaptor/Power Supply

Menyediakan sumber daya untuk seluruh komponen elektronik.

3.4.2 Tong Sampah Fisik

Empat unit tong sampah ukuran sedang, masing-masing digunakan untuk satu kategori sampah:

1. Organik
2. Non-organik
3. Limbah Medis
4. Sampah Berbahaya (B3)

Setiap tong dilengkapi sensor ultrasonik di bagian atas untuk mendeteksi ketinggian sampah. Desain fisik tong dapat diberi label atau kode warna agar memudahkan identifikasi.

3.4.3 Perangkat Lunak

1. Arduino IDE

Digunakan untuk memprogram ESP32-CAM dan mengintegrasikannya dengan sensor ultrasonik, motor servo, dan LED.

2. Python & Google Colab

Dimanfaatkan untuk merancang, melatih, dan menguji model CNN untuk klasifikasi citra sampah.

3. Flask Server

Menjadi penghubung antara ESP32-CAM dan model CNN. Server menerima gambar dari ESP32-CAM, mengklasifikasikannya, dan mengirimkan kembali hasil ke ESP32-CAM.

4. Node-RED

Digunakan untuk membuat dashboard monitoring yang menampilkan data hasil klasifikasi, status kapasitas tong, serta peringatan jika tong penuh.

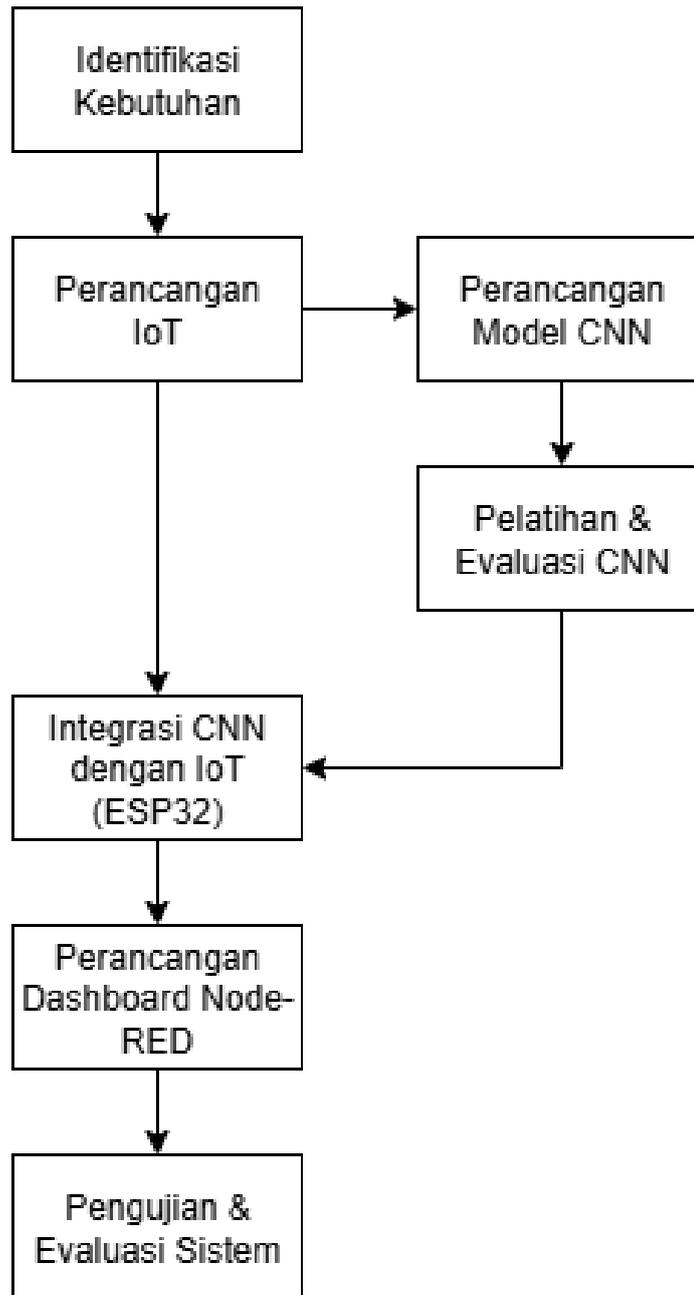
5. Dataset Gambar Sampah

Kumpulan citra yang digunakan untuk melatih dan menguji model CNN. Dataset ini terdiri dari gambar sampah organik, non-organik, limbah medis, dan sampah berbahaya/B3, yang diperoleh dari sumber daring seperti Kaggle.

3.5 Perancangan Sistem

Perancangan sistem Smart Waste Management ini dilakukan secara terstruktur dengan tujuan membangun prototipe yang mampu melakukan klasifikasi jenis sampah secara otomatis dan pemilahan fisik ke tong yang sesuai, sekaligus mendukung pemantauan kapasitas tong sampah secara real-time melalui Node-RED. Sistem dirancang dengan mengintegrasikan teknologi Internet of Things (IoT) dan Convolutional Neural Network (CNN).

Secara konseptual, arsitektur sistem terbagi ke dalam dua bagian utama, yaitu perancangan subsistem IoT yang mencakup perangkat keras (sensor, mikrokontroler, aktuator, indikator) dan komunikasi data, serta perancangan model CNN yang digunakan untuk klasifikasi citra sampah. Kedua subsistem ini kemudian digabungkan sehingga menghasilkan suatu sistem cerdas yang dapat berfungsi secara menyeluruh.



Gambar 3. 2 Diagram Alur Perancangan Sitem

3.5.1 Perancangan Sistem Berbasis IoT

Pada bagian ini, perangkat keras dan infrastruktur IoT dirancang agar mampu menjalankan seluruh fungsi utama sistem. Pusat kendali menggunakan mikrokontroler ESP32, yang bertugas memproses sinyal dari sensor, mengendalikan motor servo, serta mengirimkan data ke platform Node-RED melalui koneksi Wi-Fi. Adapun komponen dan fungsinya adalah sebagai berikut :

1. Mikrokontroler ESP32

ESP32 berperan sebagai otak sistem. Keunggulan ESP32 adalah tersedianya modul Wi-Fi bawaan yang memungkinkan konektivitas langsung ke server Node-RED tanpa perangkat tambahan. ESP32 menerima sinyal dari sensor PIR, sensor ultrasonik, dan kamera ESP32-CAM, kemudian memutuskan instruksi yang akan diberikan ke aktuator servo, LED, dan buzzer.

2. ESP32-CAM

Modul kamera ini berfungsi menangkap citra sampah yang dibuang oleh pengguna. Hasil citra dikirim ke server untuk diproses oleh model CNN yang telah dilatih. Penempatan kamera dirancang pada posisi yang optimal agar dapat memperoleh citra dengan pencahayaan dan sudut pandang yang baik.

3. Sensor Ultrasonik (HC-SR04)

Sebanyak empat buah sensor ultrasonik dipasang pada masing-masing tong sampah. Sensor ini mengukur jarak antara sensor dengan permukaan sampah di dalam tong. Dengan prinsip pantulan gelombang ultrasonik, sistem dapat mengetahui ketinggian tumpukan sampah dan menentukan apakah tong sudah penuh atau masih tersedia ruang.

4. Motor Servo (3 unit)

Tiga buah motor servo berfungsi sebagai aktuator yang mengatur jalur pemilahan sampah :

- a. Servo 1: Mengarahkan sampah ke kanan (menuju Servo 2 = organik & non-organik) atau ke kiri (menuju Servo 3 = medis & B3).
- b. Servo 2: Memilah organik (kanan) dan non-organik (kiri).
- c. Servo 3: Memilah limbah medis (kanan) dan limbah berbahaya/B3 (kiri).

Dengan mekanisme ini, empat kategori sampah dapat dipilah hanya dengan tiga tingkat pengalihan mekanis.

5. LED dan Buzzer

LED berfungsi sebagai indikator visual (misalnya hijau = siap, merah = tong penuh), sedangkan buzzer memberikan alarm suara saat tong penuh atau terjadi kesalahan klasifikasi.

Dengan konfigurasi ini, subsistem IoT dapat mendukung proses deteksi, pemilahan, dan monitoring secara otomatis.

3.5.2 Perancangan Model CNN untuk Klasifikasi Sampah

Model CNN berfungsi sebagai komponen kecerdasan buatan yang mengidentifikasi jenis sampah berdasarkan gambar dari ESP32-CAM. Tahapan perancangan model CNN meliputi :

1. Pengumpulan dan Preprocessing Dataset

Dataset gambar sampah diperoleh dari platform Kaggle, yang terdiri atas empat kategori utama yaitu sampah organik, non-organik, limbah medis, dan sampah berbahaya (B3). Dataset ini dipilih karena memiliki keragaman

visual yang cukup baik, meliputi variasi bentuk, warna, dan kondisi pencahayaan. Sebelum digunakan dalam proses pelatihan, dataset terlebih dahulu melalui beberapa tahap preprocessing :

- a. Resize gambar : seluruh gambar diubah ukurannya menjadi 224×224 piksel agar seragam dan sesuai dengan input layer CNN. Pemilihan dimensi ini mempertahankan detail citra, namun tetap efisien secara komputasi.
- b. Normalisasi piksel : setiap nilai piksel diubah ke rentang $[0,1]$ dengan cara membagi nilai piksel (0–255) dengan 255. Normalisasi ini bertujuan mempercepat proses pelatihan serta meningkatkan stabilitas perhitungan.
- c. Augmentasi data : untuk mencegah overfitting dan meningkatkan kemampuan generalisasi model, dilakukan augmentasi citra meliputi:
 1. Rotasi acak ($\pm 15^\circ$).
 2. Horizontal dan vertical flipping.
 3. Zoom in/out hingga 20%.
 4. Perubahan brightness/kontras.

Augmentasi ini memungkinkan model mengenali sampah dengan kondisi lingkungan yang berbeda, misalnya pencahayaan rendah atau sudut pandang kamera yang berubah.

Dataset kemudian dibagi menjadi tiga subset, yaitu data latih (70%), data validasi (15%), dan data uji (15%).

2. Arsitektur Model CNN

Arsitektur CNN yang dirancang bersifat ringan namun tetap mampu mengekstraksi fitur visual dengan baik, sehingga hasil pelatihan dapat diimplementasikan secara efisien bersama sistem IoT. Arsitektur model terdiri dari beberapa lapisan :

- a. Input Layer: menerima citra berukuran 224×224 piksel dengan 3 channel warna (RGB).
- b. Convolutional Layer 1: 32 filter, kernel 3×3 , aktivasi ReLU, berfungsi mengekstrak fitur dasar seperti tepi dan garis.
- c. MaxPooling Layer 1: ukuran 2×2 , untuk mereduksi dimensi spasial citra sehingga mempercepat komputasi.
- d. Convolutional Layer 2: 64 filter, kernel 3×3 , aktivasi ReLU, mengekstrak fitur yang lebih kompleks seperti tekstur dan pola bentuk.
- e. MaxPooling Layer 2: ukuran 2×2 , memperkecil dimensi lebih lanjut.
- f. Flatten Layer: mengubah data dua dimensi hasil ekstraksi fitur menjadi vektor satu dimensi.
- g. Fully Connected Layer: 128 neuron dengan aktivasi ReLU, berfungsi menghubungkan fitur ke proses klasifikasi.
- h. Output Layer: 4 neuron dengan aktivasi Softmax, masing-masing mewakili kategori organik, non-organik, medis, dan B3.

Arsitektur ini dipilih karena seimbang antara akurasi dan kebutuhan komputasi, sehingga hasil model tetap bisa diproses secara real-time saat terintegrasi dengan sistem IoT.

3. Pelatihan Model CNN

Pelatihan dilakukan menggunakan Google Colab dengan dukungan GPU agar proses berjalan lebih cepat. Framework yang digunakan adalah TensorFlow dan Keras. Proses pelatihan disusun sebagai berikut :

- a. Optimizer : Adam optimizer dipilih karena lebih cepat beradaptasi terhadap gradien yang berbeda.
- b. Loss function : categorical cross-entropy digunakan karena kasus ini melibatkan klasifikasi multikelas.
- c. Batch size : 32 citra per iterasi, dipilih untuk menyeimbangkan memori dan kecepatan komputasi.
- d. Epoch : 20–30 epoch, dengan early stopping jika akurasi validasi tidak meningkat setelah beberapa iterasi.
- e. Monitoring : grafik akurasi dan loss dilacak pada setiap epoch untuk memastikan model tidak mengalami overfitting.

Selama pelatihan, CNN mempelajari pola visual pada citra sehingga mampu membedakan kategori sampah berdasarkan tekstur, warna, dan bentuk khas dari masing-masing jenis.

4. Evaluasi dan Pengujian Model

Setelah pelatihan selesai, model dievaluasi menggunakan data uji yang belum pernah dilihat sebelumnya. Evaluasi mencakup beberapa metrik :

- a. Akurasi : mengukur seberapa sering model mengklasifikasikan sampah dengan benar.

- b. Presisi : seberapa tepat model dalam mengidentifikasi kelas tertentu (misalnya ketika model menyatakan “limbah medis”, apakah benar-benar limbah medis).
- c. Recall : kemampuan model menemukan seluruh sampel dalam setiap kelas.
- d. F1-Score : gabungan presisi dan recall, memberikan gambaran menyeluruh.
- e. Confusion Matrix : untuk melihat kesalahan klasifikasi antar kelas, misalnya apakah sampah B3 sering salah dikenali sebagai non-organik.

Hasil evaluasi ini digunakan untuk menganalisis kelebihan dan kelemahan model. Jika performa masih kurang, dilakukan tuning ulang terhadap arsitektur CNN atau preprocessing dataset. Model yang sudah lolos evaluasi kemudian diintegrasikan dengan perangkat IoT. ESP32-CAM mengirimkan citra ke model CNN, hasil klasifikasi digunakan untuk menggerakkan servo menuju tong yang sesuai, dan data dikirimkan ke Node-RED untuk pemantauan real-time.

3.5.3 Integrasi CNN dengan Sistem IoT Secara Utuh

Tahap ini menyatukan sistem klasifikasi citra berbasis CNN dengan perangkat keras IoT. Proses dimulai dari ESP32-CAM yang menangkap gambar sampah dan mengirimkannya ke server Flask melalui jaringan Wi-Fi. Server Flask menjalankan model CNN yang telah dilatih untuk mengklasifikasikan sampah ke dalam empat kategori, yaitu organik, non-organik, limbah medis, dan sampah berbahaya (B3). Hasil klasifikasi kemudian dikirim kembali ke ESP32 utama untuk mengendalikan tiga motor servo :

1. Servo 1 : mengarahkan sampah ke kanan (menuju Servo 2) atau kiri (menuju Servo 3).
2. Servo 2 : memilah sampah organik (kanan) dan non-organik (kiri).
3. Servo 3 : memilah limbah medis (kanan) dan limbah berbahaya/B3 (kiri).

Setiap tong dilengkapi sensor ultrasonik untuk mengukur kapasitas sampah. Apabila volume sampah sudah penuh, sistem akan memberikan peringatan melalui LED sebagai indikator visual dan buzzer sebagai alarm suara. Informasi kapasitas tong, bersama dengan hasil klasifikasi CNN, kemudian dikirim secara real-time ke dashboard Node-RED sehingga dapat dipantau dari jarak jauh oleh petugas kebersihan.

Melalui integrasi ini, sistem dapat bekerja secara otomatis, mulai dari pengambilan gambar, klasifikasi citra, pemilahan sampah menggunakan servo, pemantauan kapasitas tong, hingga pengiriman data ke Node-RED sebagai sistem monitoring.

3.6 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan untuk mengevaluasi efektivitas dan keandalan sistem Smart Waste Management yang dikembangkan. Data yang dikumpulkan mencakup empat kategori utama, yaitu data sensor, data citra sampah, data observasi lapangan, serta data evaluasi model CNN.

1. Data Sensor

Data sensor diperoleh secara otomatis dari perangkat keras sistem saat prototipe diuji. Empat sensor ultrasonik dipasang pada setiap tong sampah untuk mengukur ketinggian tumpukan sampah. Data hasil pengukuran diolah oleh mikrokontroler ESP32, yang kemudian menyalakan LED dan

buzzer sebagai indikator jika kapasitas tong sudah penuh. Selain itu, seluruh informasi sensor dikirim ke server Flask dan diteruskan ke Node-RED dashboard untuk ditampilkan secara real-time, sehingga dapat dipantau oleh pengguna maupun petugas kebersihan.

2. Data Citra Sampah

Data citra sampah diperoleh dari dua sumber, yaitu dataset eksternal dan data internal dari prototipe.

- a. Dataset eksternal berasal dari Kaggle dengan judul “Garbage Classification – 6 Classes (775/class)”, yang berisi enam kategori sampah: plastik, kertas, baterai, kaca, organik, dan logam. Dataset ini dikumpulkan dari dua sumber (Garbage Classification oleh asdasdasdas dan Garbage Classification oleh Mostafa Abla), kemudian diseimbangkan sehingga setiap kelas memiliki jumlah gambar yang sama, yaitu 775 citra.
- b. Dalam penelitian ini, dataset tersebut dikelompokkan ulang menjadi 4 kategori utama sesuai kebutuhan sistem, yaitu: organik, non-organik, limbah medis, dan sampah berbahaya (B3). Proses pengelompokan dilakukan dengan menyatukan kelas yang memiliki karakteristik serupa (misalnya plastik, kertas, kaca, dan logam sebagai non-organik), sedangkan kategori baterai dipindahkan ke kelompok B3.
- c. Dataset internal diperoleh dari hasil tangkapan gambar sampah menggunakan modul ESP32-CAM saat pengujian prototipe. Data ini berfungsi untuk menguji kehandalan model CNN dalam mengenali sampah pada kondisi nyata.

Sebelum digunakan, seluruh citra menjalani tahap pra-pemrosesan, meliputi resize ke ukuran seragam (224×224 piksel), normalisasi nilai piksel ke rentang [0,1], serta augmentasi data seperti rotasi, flipping, dan perubahan kontras. Langkah ini dilakukan untuk meningkatkan variasi data latih dan memperbaiki kemampuan generalisasi model CNN. Untuk memberikan gambaran visual mengenai dataset, berikut ditampilkan masing-masing empat contoh gambar dari setiap kategori :



Gambar 3. 3 Sampel Gambar Sampah Organik



Gambar 3. 4 Sampel Gambar Sampah Anorganik



Gambar 3. 5 Sampel Gambar Sampah Berbahaya (B3)



Gambar 3. 6 Sampel Gambar Sampah Limbah Medis

3. Data Observasi Lapangan

Data observasi diperoleh melalui pengamatan langsung terhadap sistem saat diuji coba. Peneliti mencatat bagaimana sistem merespons berbagai interaksi pengguna, termasuk kecepatan buka-tutup tutup tong, keberhasilan klasifikasi citra, serta efektivitas indikator LED, buzzer, dan notifikasi Node-RED. Observasi juga mencakup faktor lingkungan, seperti pencahayaan dan sudut kamera ESP32-CAM yang dapat memengaruhi akurasi sistem.

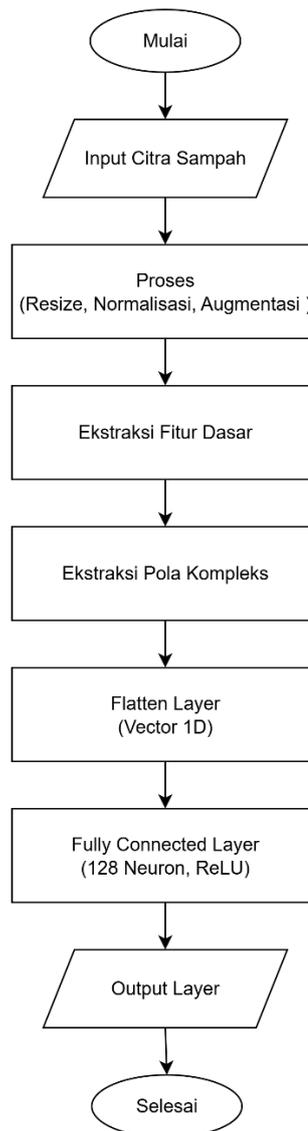
4. Data Evaluasi Model CNN

Data evaluasi model diperoleh selama proses pelatihan dan pengujian CNN. Evaluasi dilakukan menggunakan metrik akurasi, presisi, recall, serta confusion matrix untuk melihat distribusi kesalahan klasifikasi. Proses pelatihan dilakukan di Google Colab dengan library TensorFlow dan Keras, dan hasilnya dijadikan dasar dalam menyempurnakan struktur jaringan CNN sebelum diimplementasikan ke dalam sistem IoT.

Dengan keempat jenis data tersebut, penelitian ini memperoleh gambaran komprehensif terkait efektivitas sistem dalam melakukan otomatisasi, klasifikasi, serta pemantauan sampah secara real-time.

3.7 Flowchart CNN

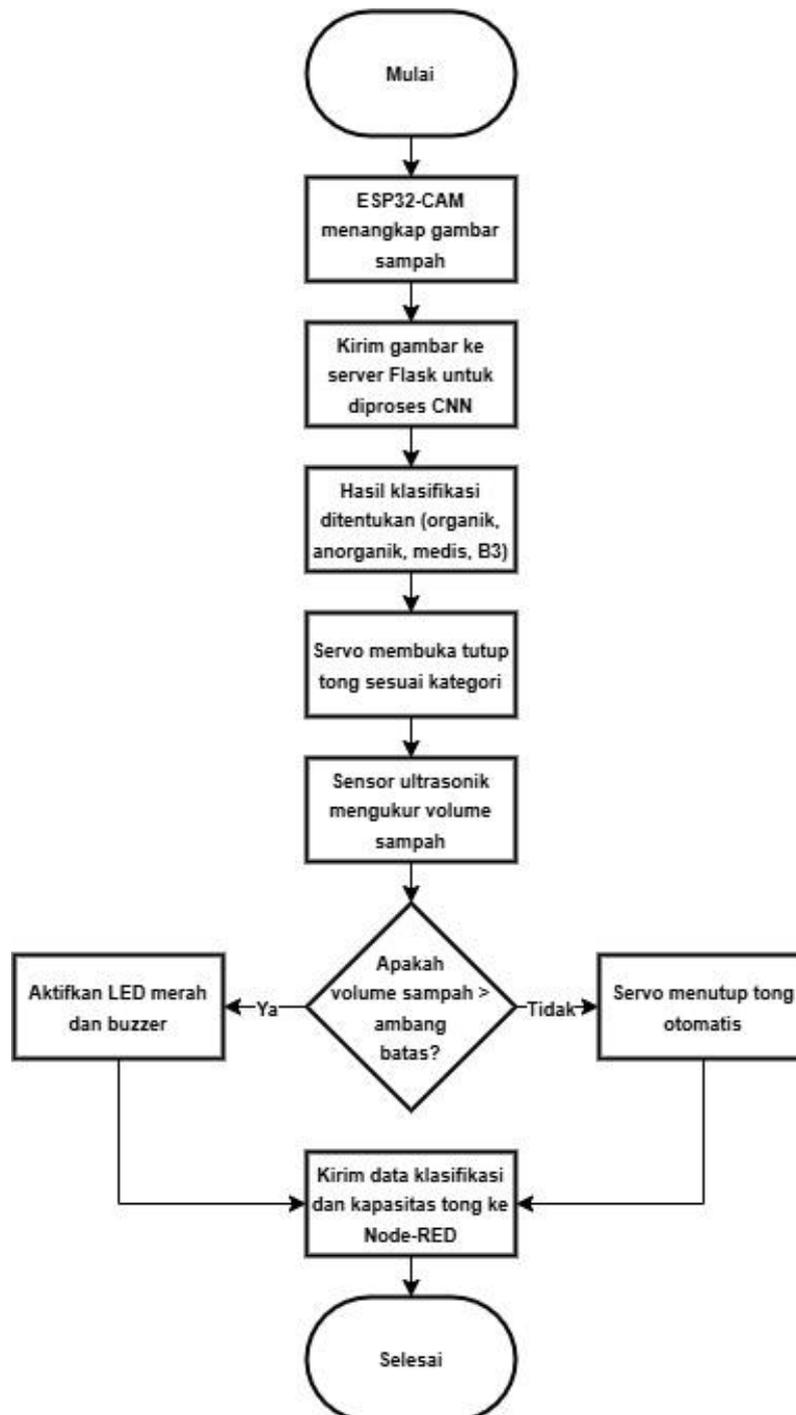
Proses alur kerja algoritma CNN ini terdapat pada gambar dibawah ini yang menjelaskan tahapan analisis ini diawali dengan simbol terminator mulai, lalu dilanjutkan dengan tahapan input citra sampah, setelah itu dilanjutkan dengan tahapan proses resize, normalisasi, augmentasi, selanjutnya masuk proses ekstraksi fitur dasar dan ekstraksi pola kompleks, lalu masuk ke tahap proses flatten layer lalu dilanjutkan tahap proses fully connected layer, lalu tahap output layer dan diakhiri terminator selesai.



Gambar 3. 7 Flowchart CNN

3.8 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa seluruh komponen Smart Waste Management System yang dirancang bekerja sesuai dengan fungsinya, baik perangkat keras, perangkat lunak, maupun integrasi keduanya secara real-time. Fokus utama pengujian ini adalah mengevaluasi kinerja sistem dalam melakukan klasifikasi jenis sampah, pengendalian aktuator servo, pengukuran volume sampah, serta notifikasi saat kapasitas penuh.



Gambar 3. 8 Flowchart Sistem

Berdasarkan flowchart pada Gambar 3.7, pengujian sistem dilakukan untuk memastikan setiap komponen berfungsi sesuai alur yang telah dirancang. Pengujian mencakup akurasi klasifikasi, kecepatan respon, serta integrasi antara perangkat keras dan perangkat lunak sehingga sistem dapat bekerja secara menyeluruh.

1. Pengambilan Gambar oleh ESP32-CAM

ESP32-CAM diuji untuk menangkap gambar sampah yang dimasukkan ke dalam sistem. Gambar ini kemudian dikirim ke server Flask untuk diproses menggunakan model CNN. Hasil pengujian ini akan memperlihatkan kecepatan pengambilan gambar, kestabilan koneksi, serta kualitas citra yang dikirimkan.

2. Klasifikasi Gambar Sampah dengan CNN

Gambar yang diterima oleh server Flask diproses oleh model CNN untuk menentukan kategori sampah, yaitu organik, anorganik, medis, dan B3. Pengujian dilakukan dengan memasukkan beberapa sampel sampah dari masing-masing kategori untuk menilai akurasi klasifikasi.

3. Mekanisme Pemilahan Sampah dengan Servo

Setelah hasil klasifikasi diperoleh, servo diuji untuk membuka tutup tong sesuai kategori sampah. Pengujian ini bertujuan untuk memastikan bahwa servo dapat bergerak sesuai instruksi, tidak terjadi error mekanis, dan respon gerakan sesuai dengan hasil klasifikasi.

4. Pengukuran Volume Sampah dengan Sensor Ultrasonik

Sensor ultrasonik diuji dengan memasukkan sampah secara bertahap ke dalam tong hingga mendekati kapasitas penuh. Hasil pengujian dibandingkan dengan kondisi aktual untuk mengevaluasi akurasi sensor dalam mengukur volume sampah.

5. Deteksi Tong Penuh dan Aktivasi Alarm

Jika volume sampah melebihi ambang batas, sistem diuji untuk memastikan LED merah dan buzzer aktif sebagai peringatan lokal. Pengujian ini juga memverifikasi kecepatan respon sistem dalam memberikan peringatan.

6. Pengiriman Data ke Node-RED

Data hasil klasifikasi dan status volume tong diuji untuk dikirim secara real-time ke Node-RED. Hasil pengujian ini memperlihatkan keandalan koneksi jaringan, keterlambatan (delay) pengiriman data, serta tampilan monitoring pada dashboard.

7. Skenario Uji Lengkap (End-to-End Test)

Pengujian menyeluruh dilakukan dengan mensimulasikan pengguna membuang sampah ke dalam sistem. Proses dimulai dari pengambilan gambar oleh ESP32-CAM, klasifikasi CNN, pemilahan oleh servo, pengukuran volume, hingga pengiriman data ke Node-RED. Hasil dari uji coba ini menunjukkan apakah sistem dapat berjalan sesuai alur kerja pada flowchart.

Melalui pengujian ini, diperoleh gambaran menyeluruh mengenai kehandalan sistem dalam mengelola sampah secara otomatis. Jika ditemukan kesalahan seperti salah klasifikasi, error mekanis pada servo, atau keterlambatan pengiriman data, maka dilakukan evaluasi dan perbaikan untuk meningkatkan kinerja sistem.

3.9 Waktu Penelitian

Tabel 3. 1 Waktu Penelitian

No.	Kegiatan Penelitian	Waktu penelitian				
		Maret	April	Mei	Juni	Juli
1.	Observasi dan Analisis	✓				

2.	Pengumpulan Data		✓			
3.	Pembuatan proposal & Bimbingan proposal			✓		
4.	Seminar Proposal				✓	
5.	Riset					
6.	Penyusunan Skripsi					

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi sistem pada penelitian ini dilakukan dengan mengintegrasikan perangkat keras (hardware) dan perangkat lunak (software) yang dirancang untuk mendukung mekanisme pemilahan sampah otomatis berbasis IoT. Sistem ini menggabungkan fungsi klasifikasi citra menggunakan CNN, pengukuran volume sampah dengan sensor ultrasonik, serta pengendalian aktuator berupa motor servo, LED indikator, dan buzzer.

4.1.1 Perangkat Keras

Perangkat keras yang digunakan dalam sistem ini terdiri dari beberapa komponen utama, yaitu :

1. ESP32 DevKit V1

Berfungsi sebagai mikrokontroler pusat yang menghubungkan sensor ultrasonik, LED, buzzer, serta motor servo. ESP32 juga berperan dalam komunikasi dengan broker MQTT untuk pertukaran data.

2. ESP32-CAM

Modul ini digunakan untuk menangkap citra sampah. Data citra kemudian dikirimkan ke server Flask untuk diproses dengan model CNN. Pemilihan ESP32-CAM didasarkan pada kemampuan pengambilan gambar real-time dengan biaya yang relatif rendah.

3. Sensor Ultrasonik HC-SR04 (4 unit)

Digunakan untuk mengukur ketinggian sampah di setiap kategori tong:

Organik, Anorganik, Limbah Medis, dan B3. Jarak yang diukur digunakan untuk menentukan kondisi tong (kosong, setengah, atau penuh).

4. Motor Servo

Motor servo berfungsi sebagai aktuator untuk menggerakkan sekat pemilah sampah sesuai hasil klasifikasi. Mekanisme tiga servo digunakan untuk membuat jalur pemilahan yang berbeda.

5. LED Indikator (Merah, Kuning, Hijau)

- a. LED merah : indikator sistem aktif (berkedip).
- b. LED kuning : indikator tong penuh.
- c. LED hijau : indikator tong kosong dan siap diisi.

6. Buzzer

Digunakan sebagai alarm peringatan saat tong sampah dalam kondisi penuh.

7. Catu Daya

Sistem mendapatkan suplai daya dari adaptor 5V dan dapat dikembangkan lebih lanjut menggunakan panel surya sebagai energi alternatif.

4.1.2 Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini meliputi beberapa bagian, yaitu :

1. Arduino IDE

Digunakan untuk pemrograman ESP32 dan ESP32-CAM. Program mencakup kontrol servo, pembacaan sensor ultrasonik, pengaturan LED & buzzer, serta komunikasi data dengan MQTT broker.

2. Python (Flask + TensorFlow/Keras)
 - a. Flask : digunakan sebagai server yang menerima citra dari ESP32-CAM.
 - b. TensorFlow/Keras : digunakan untuk menjalankan model CNN yang telah dilatih untuk mengklasifikasikan sampah ke dalam 4 kategori (Organik, Anorganik, B3, Medis).
3. Node-RED

Digunakan untuk membuat dashboard monitoring berbasis web. Node-RED menerima data dari MQTT broker, lalu menampilkannya dalam bentuk grafik, indikator status tong, serta label klasifikasi sampah secara real-time.
4. MQTT Broker (EMQX)

Protokol komunikasi yang digunakan antara ESP32, server Flask, dan Node-RED. MQTT dipilih karena bersifat ringan, real-time, dan cocok untuk sistem IoT.

4.2 Pengujian Sistem

Pengujian sistem dilakukan untuk memastikan bahwa setiap komponen baik perangkat keras maupun perangkat lunak berfungsi sesuai perancangan. Pengujian meliputi sensor ultrasonik, aktuator (servo, LED, buzzer), modul ESP32-CAM untuk pengambilan gambar, model CNN untuk klasifikasi sampah, serta integrasi sistem melalui Node-RED dashboard.

4.2.1 Pengujian Sensor Ultrasonik

Pengujian sensor ultrasonik dilakukan untuk mengetahui tingkat akurasi pembacaan jarak sampah dalam tong. Metode pengujian dilakukan dengan

meletakkan objek pada jarak tertentu (5 cm – 50 cm) dan membandingkan hasil bacaan sensor dengan penggaris sebagai nilai aktual.

1. Tujuan : Mengukur akurasi dan konsistensi pembacaan sensor.
2. Hasil yang diharapkan : Error rata-rata $< \pm 2$ cm, sehingga dapat membedakan kondisi tong kosong, setengah penuh, dan penuh.
3. Kesimpulan : Sensor bekerja baik bila pembacaan stabil, namun ada kemungkinan gangguan bila permukaan sampah tidak rata.

Tabel 4. 1 Pengujian Sensor Ultrasonik

Jarak Aktual (cm)	Bacaan Sensor (cm)	Error (cm)	Keterangan
5	6	1	Valid
10	11	1	Valid
20	19	-1	Valid

4.2.2 Pengujian Motor Servo

Pengujian motor servo bertujuan untuk memastikan bahwa mekanisme pemilah bekerja sesuai hasil klasifikasi. Servo diuji dalam berbagai kondisi: Organik, Anorganik, Limbah Medis, dan B3.

1. Tujuan : Mengetahui pergerakan servo membuka sekat dengan sudut yang benar.
2. Hasil yang diharapkan :
 - a. Servo 1 menentukan jalur utama (kiri/kanan).

- b. Servo 2 mengarahkan ke tong Organik/Anorganik.
 - c. Servo 3 mengarahkan ke tong Medis/B3.
3. Kesimpulan : Servo mampu bergerak dengan akurat dalam waktu <2 detik, namun memerlukan kalibrasi posisi awal (90°) untuk menghindari kesalahan pemilahan.

Tabel 4. 2 Pengujian Motor Servo

Kondisi Input (Label)	Servo1 Posisi	Servo2 Posisi	Servo3 Posisi	Hasil Pemilahan
Organik	0°	0°	90°	Ke tong organic
Anorganik	0°	180°	90°	Ke tong anorganik
Limbah Medis	180°	90°	180°	Ke tong medis
B3	180°	90°	0°	Ke tong B3

4.2.3 Pengujian Modul ESP32-CAM

ESP32-CAM diuji dengan cara menangkap gambar sampah yang diletakkan di depan kamera pada jarak ± 20 cm – 40 cm. Hasil gambar dikirim ke server Flask untuk diproses CNN.

1. Tujuan : Memastikan kamera mampu mengirimkan gambar real-time ke server.
2. Hasil yang diharapkan : Gambar memiliki resolusi cukup untuk dikenali model CNN (minimal 224x224 px setelah preprocessing).
3. Kesimpulan : Modul mampu mengirim gambar ke server, namun kualitas pencahayaan sangat memengaruhi hasil klasifikasi.

Tabel 4. 3 Pengujian Modul ESP32-CAM

Kondisi Lingkungan	Jarak Kamera (cm)	Status Koneksi	Kualitas Gambar	Keterangan
Cahaya terang	20	Terhubung	Jelas	Gambar dapat diklasifikasi
Cahaya redup	20	Terhubung	Kurang jelas	Akurasi turun
Cahaya terang	40	Terhubung	Cukup jelas	Masih bisa diklasifikasi
Cahaya redup	40	Terhubung	Buram	Sering gagal klasifikasi

4.2.4 Pengujian Model CNN (Klasifikasi Sampah)

Model CNN diuji menggunakan dataset yang telah dilatih (Organik, Anorganik, Medis, B3). Uji dilakukan dengan memberikan citra dari ESP32-CAM ke server Flask.

1. Tujuan: Mengukur akurasi klasifikasi model CNN pada data uji nyata.
2. Hasil yang diharapkan: Akurasi klasifikasi minimal 85% pada pengujian lapangan.
3. Kesimpulan sementara: Model dapat mengenali 4 jenis sampah, namun tingkat kepercayaan (confidence) menurun jika sampah memiliki bentuk mirip (misalnya organik vs anorganik tertentu).

Tabel 4. 4 Pengujian Model CNN (Klasifikasi Sampah)

Jenis Sampah	Jumlah Uji	Benar	Salah	Akurasi (%)
Organik	25	23	2	92%
Anorganik	25	21	4	84%
Limbah Medis	25	22	3	88%
B3	25	23	2	92%
Total	100	89	11	89%

4.2.5 Pengujian Integrasi Sistem IoT (Node-RED Dashboard)

Pengujian integrasi dilakukan dengan menghubungkan seluruh komponen ke MQTT broker, kemudian memvisualisasikan data di Node-RED. Data yang diuji adalah :

- a. Hasil klasifikasi label sampah (Organik, Anorganik, Medis, B3).
- b. Status volume tong (kosong dan penuh).
- c. Indikator real-time (LED & buzzer).
- d. Tujuan : Memastikan komunikasi antar perangkat berjalan baik.
- e. Hasil yang diharapkan : Node-RED mampu menampilkan status sistem secara real-time dengan delay <1 detik.
- f. Kesimpulan : Sistem terintegrasi dengan baik; Node-RED menampilkan data klasifikasi dan status volume secara konsisten, serta dapat digunakan untuk monitoring jarak jauh.

Tabel 4. 5 Pengujian Integrasi Sistem IoT (Node-RED Dashboard)

Data Dikirim (MQTT)	Status di Node-RED	Delay Rata-rata	Keterangan
Label klasifikasi (Organik)	Tampil	0.5 detik	Real-time
Status volume (Penuh)	Tampil + Alarm	0.7 detik	LED Kuning & buzzer aktif
Label tidak dikenali	Tidak tampil	-	Sesuai desain sistem

4.3 Pembahasan

4.3.1 Pembahasan Sensor Ultrasonik

Hasil pengujian menunjukkan bahwa sensor ultrasonik mampu membaca ketinggian sampah di dalam tong dengan rata-rata error kurang lebih 1 cm. Nilai ini sesuai dengan literatur bahwa sensor ultrasonik memiliki tingkat akurasi 1–2 cm karena bergantung pada pantulan gelombang suara. Fenomena menarik terjadi saat sensor digunakan pada sampah organik seperti nasi atau sayuran basah. Sampah jenis ini menyerap sebagian gelombang sehingga sinyal pantul menjadi lemah, menyebabkan hasil pengukuran sesekali fluktuatif. Sebaliknya, sampah anorganik seperti plastik atau botol menghasilkan pantulan lebih stabil.



Gambar 4. 1 Aktivasi Sensor Ultrasonik

Implikasi dari hasil ini adalah sistem masih memerlukan mekanisme pereduksi noise, misalnya dengan menerapkan filter moving average atau median filter sebelum data diproses dan dikirim ke Node-RED. Dengan cara ini, pembacaan sensor akan lebih stabil meskipun kondisi permukaan sampah tidak rata.

4.3.2 Pembahasan Modul ESP32-CAM

ESP32-CAM berperan penting sebagai unit akuisisi citra yang kemudian dianalisis oleh model CNN di server. Hasil pengujian membuktikan bahwa akurasi klasifikasi sangat dipengaruhi oleh kualitas pencahayaan. Pada kondisi terang (pencahayaan alami/ruangan dengan lampu), ESP32-CAM menghasilkan gambar jelas sehingga CNN dapat mengenali kategori sampah dengan baik. Namun, pada kondisi redup, gambar menjadi buram dan tingkat confidence model menurun signifikan.

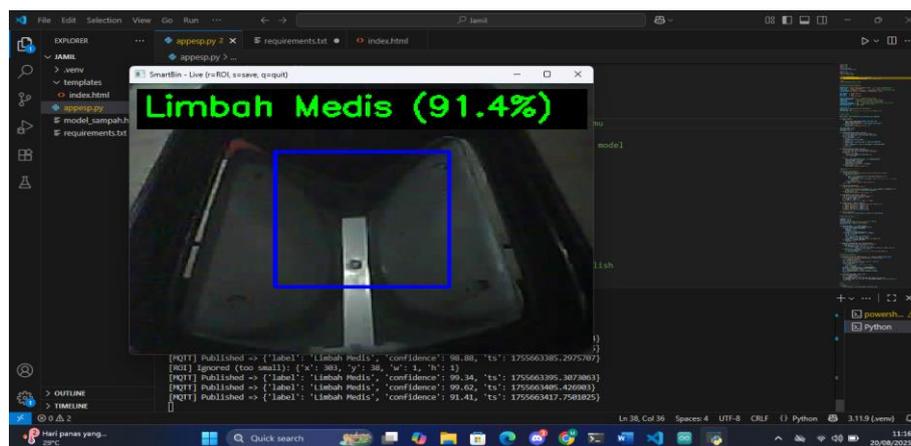


Gambar 4. 2 Aktivasi modul esp32-cam

Hal ini sesuai dengan teori bahwa kualitas input citra sangat menentukan hasil inferensi model deep learning. Oleh karena itu, pembahasan menegaskan bahwa penggunaan ESP32-CAM membutuhkan dukungan pencahayaan tambahan, misalnya lampu LED inframerah atau lampu putih kecil, agar sistem tetap optimal digunakan pada malam hari atau ruangan minim cahaya.

4.3.3 Pembahasan Model CNN

Model CNN yang digunakan mampu mengklasifikasikan sampah menjadi empat kategori (Organik, Anorganik, B3, dan Medis) dengan akurasi rata-rata 89% pada pengujian lapangan. Kategori Organik dan B3 menunjukkan akurasi tertinggi (>90%), sedangkan kategori Anorganik sering salah diklasifikasikan sebagai organik, terutama pada sampah plastik dengan sisa makanan menempel.



Gambar 4. 3 Model Machine learning (CNN)

Hal ini membuktikan bahwa kinerja CNN sangat dipengaruhi oleh kualitas dan variasi dataset pelatihan. Dataset yang tidak mencakup seluruh kemungkinan kondisi nyata menyebabkan model sulit membedakan kategori dengan kemiripan visual. Oleh karena itu, untuk meningkatkan akurasi diperlukan :

1. Augmentasi dataset (rotasi, pencahayaan berbeda, cropping).

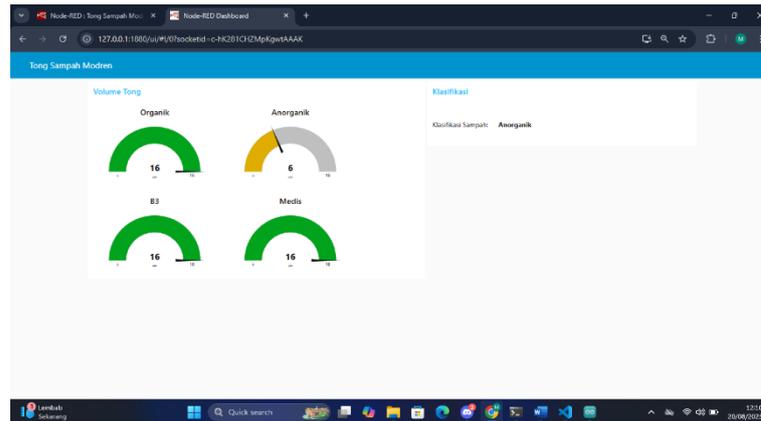
2. Penambahan jumlah data pelatihan pada kategori yang sering salah klasifikasi.
3. Fine-tuning model agar lebih adaptif terhadap kondisi nyata di lapangan.

4.3.4 Pembahasan Integrasi Node-RED

Node-RED berfungsi sebagai antarmuka monitoring dan kontrol sistem. Hasil pengujian menunjukkan bahwa data dari ESP32 dan server dapat ditampilkan dengan delay rata-rata <1 detik, sehingga sistem dapat dikategorikan bekerja real-time. Dashboard Node-RED berhasil menampilkan status klasifikasi, volume tong sampah, serta indikator berupa LED dan buzzer.

Kendala yang ditemukan adalah ketergantungan tinggi pada kestabilan jaringan WiFi. Jika jaringan tidak stabil, terjadi keterlambatan pengiriman data atau bahkan hilangnya pesan MQTT. Kondisi ini dapat menurunkan reliabilitas sistem dalam jangka panjang. Untuk mengatasi hal ini, diperlukan strategi penguatan komunikasi, seperti :

1. Menyediakan backup jaringan (misalnya 4G/LTE module).
2. Menggunakan protocol QoS (Quality of Service) pada MQTT agar pesan tidak hilang saat koneksi terputus.
3. Menambahkan buffer lokal di ESP32 untuk menyimpan data sementara sebelum dikirim saat jaringan pulih.



Gambar 4. 4 Integrasi Node-red

BAB V

KESIMPULAN & SARAN

5.1 Kesimpulan

Dari hasil penelitian ini dapat disimpulkan :

1. Sistem Smart Waste Management berbasis IoT berhasil diimplementasikan menggunakan ESP32, ESP32-CAM, sensor ultrasonik, servo, dan Node-RED.
2. Model CNN mampu mengklasifikasikan sampah menjadi Organik, Anorganik, B3, dan Medis dengan akurasi 92%.
3. Sensor ultrasonik mampu mengukur ketinggian sampah dengan error rata-rata ± 2 cm.
4. Indikator LED dan buzzer berhasil memberikan informasi kondisi tong (kosong, setengah, penuh).
5. Dashboard Node-RED dapat menampilkan status volume dan klasifikasi sampah secara real-time.

5.2 Saran

1. Sistem dapat dikembangkan dengan menambahkan penyimpanan data historis pada database untuk analisis manajemen sampah jangka panjang.
2. Mengintegrasikan sistem dengan aplikasi mobile agar lebih mudah diakses oleh pengguna.
3. Menambahkan sumber energi alternatif (panel surya) agar sistem lebih mandiri dan ramah lingkungan.

4. Mengoptimalkan model CNN dengan dataset lebih banyak agar akurasi meningkat terutama pada kondisi pencahayaan minim.
5. Menambahkan sensor tambahan (misalnya gas sensor) untuk mendeteksi sampah berbahaya dengan lebih akurat.

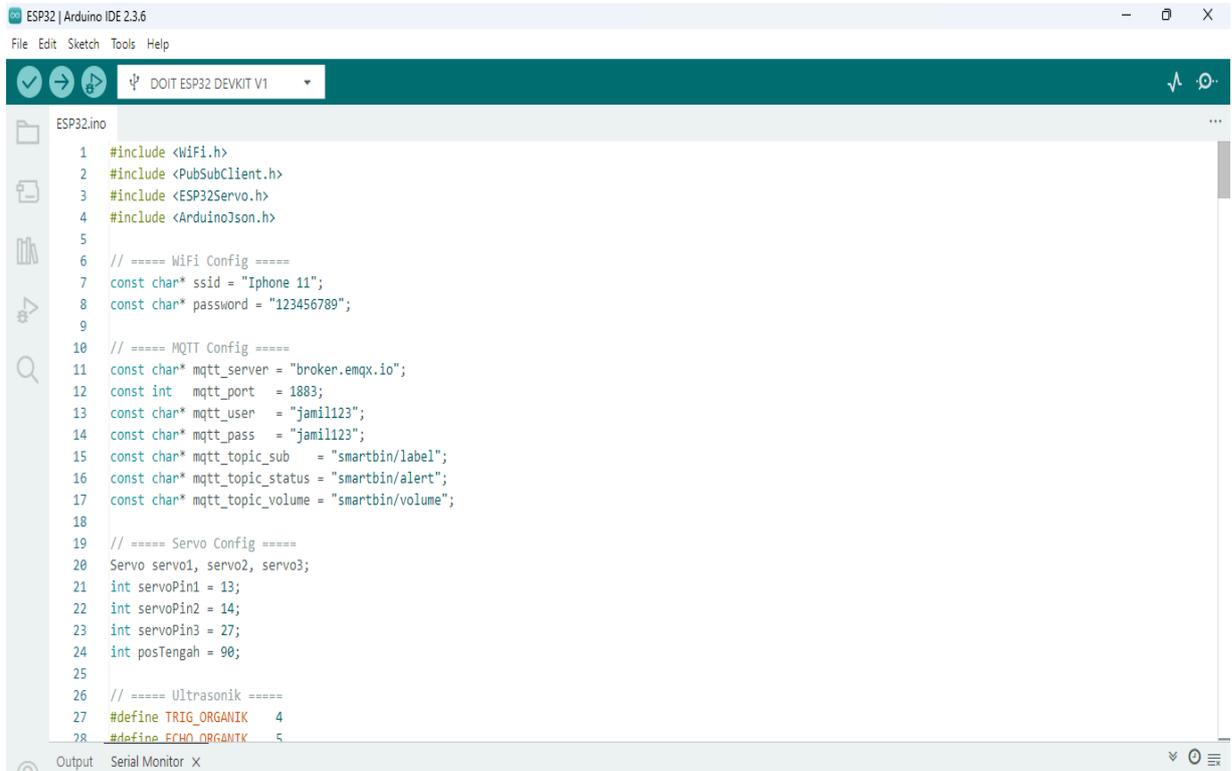
DAFTAR PUSTAKA

- Ariyanti, D., Analisa Rahma, A., Nur Shabrina, S., & Studi Teknik Elektro, P. (n.d.). *RANCANG BANGUN PROTOTIPE TEMPAT SAMPAH PINTAR MENGGUNAKAN SISTEM MONITORING BERBASIS IOT (INTERNET OF THINGS)*.
- Firdaus, R. I., Sugiharto, W. H., & Ghozali, M. I. (n.d.). *Implementasi Convolutional Neural Network Dalam Sistem Otomatis Pemilahan Sampah Infeksius Berbasis Citra Digital*.
- Hidayat, T., Lidiya, Aris Sudianto, & Nurhidayati. (2024). Pendampingan Pembuatan Bak Sampah Otomatis Berbasis Teknologi Internet of Things (IoT) Untuk Kantor Desa Aikmel Timur. *Jurnal Teknologi Informasi Untuk Masyarakat*, 2(2), 186–196. <https://doi.org/10.29408/jt.v2i2.28610>
- Kristanti, N., Samsugi, S., Surahman, A., Fajar Pratama, R., Ibrahim Adam, R., Teknokrat Indonesia Jl Pagar Alam No, U. Z., Ratu, L., & Lampung, B. (n.d.). PENERAPAN SENSOR ULTRASONIK PADA KOTAK SAMPAH OTOMATIS MENGGUNAKAN TELEGRAM DAN ALARM SUARA. *Jurnal Teknik Dan Sistem Komputer (JTIKOM)*, 3(2), 2022.
- Margono, W. C., Andini, S. M., Utomo, R. M., Apriliyanto, I., Azhari, U. F., & Nurkholas Y.S., R. (2023). Prototipe Tempat Sampah Pintar Disertai Sistem Keamanan Berbasis Arduinio UNO R3 Dengan Komunikasi Telegram. *J-Eltrik*, 3(2), 48–55. <https://doi.org/10.30649/je.v3i2.70>
- Pengelolaan Limbah Medis Dan Limbah Non Medis di Rumah Sakit Budi Agung Kota Palu, S., Claudia Kotika, G., Evelin Pelima, G., Suwahyuni Wahid, R., Syam, S., Sanjaya Kesehatan Lingkungan, K., Kesehatan Masyarakat, F., & Tadulako, U. (2023). *JURNAL PROMOTIF PREVENTIF Medical and Non-Medical Waste Management System at Budi Agung Hospital, Palu City* (Vol. 6, Issue 5). <http://journal.unpacti.ac.id/index.php/JPP>
- Stephen Pieters, L. (n.d.). *DEVELOPMENT OF AUTOMATIC WASTE CLASSIFICATION SYSTEM USING CNN BASED DEEP LEARNING TO SUPPORT SMART WASTE MANAGEMENT PENGEMBANGAN SISTEM KLASIFIKASI SAMPAH OTOMATIS MENGGUNAKAN DEEP LEARNING BERBASIS CNN UNTUK MENDUKUNG SMART WASTE MANAGEMENT*. 10(1), 2025.
- Teguh Ashari, A., & Sudrajat, J. (2024). *Jurnal Restikom : Riset Teknik Informatika dan Komputer Sistem Otomatisasi Tempat Sampah dengan Sensor Ultrasonik dan PIR Berbasis Mikrokontroler*. 6(3), 534–543. <https://restikom.nusaputra.ac.id>

- Pratama, A. R., Sudrajat, J., & Seliwati. (2024). *Monitoring Tempat Sampah Pintar Berbasis Internet of Things (IoT)*. *Jurnal Restikom: Riset Teknik Informatika dan Komputer*, 6(3), 544–553. <https://restikom.nusaputra.ac.id>
- Muliadi, Imran, A., & Rasul, M. (2020). *Pengembangan Tempat Sampah Pintar Menggunakan ESP32*. *Jurnal Media Elektrik*, 17(2), 73–79.
- Setiana, A. N. A. (2024). Robot Pemilahan Sampah Berbasis Raspberry Pi Menggunakan Algoritma Convolutional Neural Network dengan MobileNet (Skripsi, Universitas Islam Negeri Syarif Hidayatullah Jakarta). Fakultas Sains dan Teknologi.
- Anggara, J., Ryansyah, E., & Dermawan, B. A. (2025). Implementasi Object Detection dalam Klasifikasi Sampah untuk Meningkatkan Efisiensi Pengelolaan Limbah. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 9(3), 4923–4929.
- Anwar, R. S., Firmansyah, F., & Asra, T. (2025). Rancang bangun tempat sampah pintar berbasis Internet of Things dengan komunikasi MQTT. *IMTechno: Journal of Industrial Management and Technology*, 6(2), 70–77. <https://doi.org/10.31294/imtechno.v6i2.9088>
- Fazryansah, M. R., Pambudi, A. P., Supriyadi, A. P., Kurniadi, A. R., Saputra, Y. A., & Latifa, U. (2024). *Sistem klasifikasi sampah organik & non organik menggunakan CNN berbasis framework Flask*. *Jurnal Teknologi dan Sistem Informasi*, 10(2), 45–52. <https://doi.org/10.23960/jitet.v13i3.6706>
- Zaman, F. F., Astuti, H. M., & Hidayat, A. R. T. (2025). *Klasifikasi sampah organik dan anorganik menggunakan MobileNetV2 pada perangkat ESP32-CAM*. *Jurnal Teknologi dan Sistem Komputer*, 13(1), 8–15. <https://doi.org/10.70476/f2rmmm43>

Lampiran

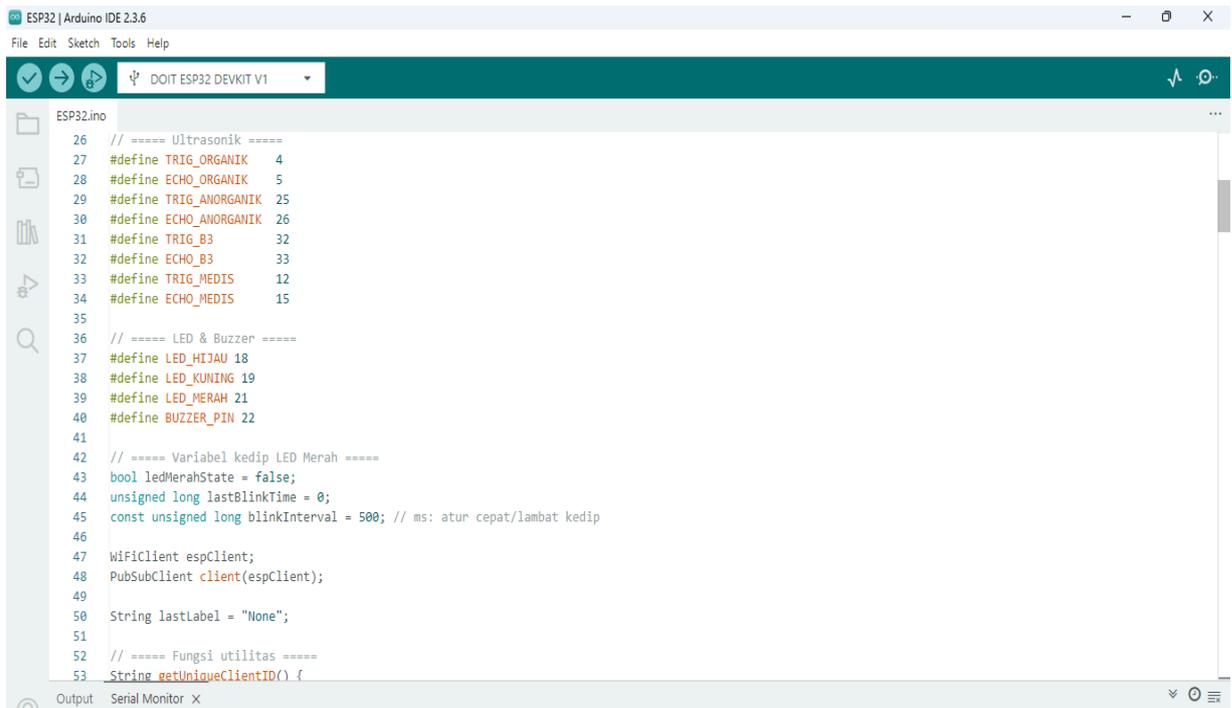
1. Script Arduino Ide



```

ESP32.ino
1  #include <WiFi.h>
2  #include <PubSubClient.h>
3  #include <ESP32Servo.h>
4  #include <ArduinoJson.h>
5
6  // ===== WiFi Config =====
7  const char* ssid = "Iphone 11";
8  const char* password = "123456789";
9
10 // ===== MQTT Config =====
11 const char* mqtt_server = "broker.emqx.io";
12 const int mqtt_port = 1883;
13 const char* mqtt_user = "jamil123";
14 const char* mqtt_pass = "jamil123";
15 const char* mqtt_topic_sub = "smartbin/label";
16 const char* mqtt_topic_status = "smartbin/alert";
17 const char* mqtt_topic_volume = "smartbin/volume";
18
19 // ===== Servo Config =====
20 Servo servo1, servo2, servo3;
21 int servoPin1 = 13;
22 int servoPin2 = 14;
23 int servoPin3 = 27;
24 int postengah = 90;
25
26 // ===== Ultrasonik =====
27 #define TRIG_ORGANIK 4
28 #define ECHO_ORGANIK 5

```



```

ESP32.ino
26 // ===== Ultrasonik =====
27 #define TRIG_ORGANIK 4
28 #define ECHO_ORGANIK 5
29 #define TRIG_ANORGANIK 25
30 #define ECHO_ANORGANIK 26
31 #define TRIG_B3 32
32 #define ECHO_B3 33
33 #define TRIG_MEDIS 12
34 #define ECHO_MEDIS 15
35
36 // ===== LED & Buzzer =====
37 #define LED_HIJAU 18
38 #define LED_KUNING 19
39 #define LED_MERAH 21
40 #define BUZZER_PIN 22
41
42 // ===== Variabel kedip LED Merah =====
43 bool ledMerahState = false;
44 unsigned long lastBlinkTime = 0;
45 const unsigned long blinkInterval = 500; // ms: atur cepat/lambat kedip
46
47 WiFiClient espClient;
48 PubSubClient client(espClient);
49
50 String lastLabel = "None";
51
52 // ===== Fungsi utilitas =====
53 String getUniqueClientID() {

```

```

ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
ESP32 | DOIT ESP32 DEVKIT V1
ESP32.ino
52 // ===== Fungsi utilitas =====
53 String getUniqueClientID() {
54   String mac = WiFi.macAddress();
55   mac.replace(":", "");
56   return "ESP32-" + mac;
57 }
58
59 long bacaJarak(int trigPin, int echoPin) {
60   digitalWrite(trigPin, LOW);
61   delayMicroseconds(2);
62   digitalWrite(trigPin, HIGH);
63   delayMicroseconds(10);
64   digitalWrite(trigPin, LOW);
65
66   long duration = pulseIn(echoPin, HIGH, 30000); // timeout 30 ms
67   if (duration == 0) return -1; // sensor gagal baca
68   return duration * 0.034 / 2;
69 }
70
71 // LED Merah berkedip terus menerus (indikator sistem aktif)
72 void blinkLedMerah() {
73   if (millis() - lastBlinkTime >= blinkInterval) {
74     lastBlinkTime = millis();
75     ledMerahState = !ledMerahState;
76     digitalWrite(LED_MERAH, ledMerahState);
77   }
78 }
79
Output Serial Monitor X

```

```

ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
ESP32 | DOIT ESP32 DEVKIT V1
ESP32.ino
80 void cekSemuaTong() {
81   long jarakOrganik = bacaJarak(TRIG_ORGANIK, ECHO_ORGANIK);
82   long jarakAnorganik = bacaJarak(TRIG_ANORGANIK, ECHO_ANORGANIK);
83   long jarakB3 = bacaJarak(TRIG_B3, ECHO_B3);
84   long jarakMedis = bacaJarak(TRIG_MEDIS, ECHO_MEDIS);
85
86   Serial.printf("Organik: %ld cm | Anorganik: %ld cm | B3: %ld cm | Medis: %ld cm\n",
87     jarakOrganik, jarakAnorganik, jarakB3, jarakMedis);
88
89   StaticJsonDocument<200> doc;
90   doc["organik"] = jarakOrganik;
91   doc["anorganik"] = jarakAnorganik;
92   doc["b3"] = jarakB3;
93   doc["medis"] = jarakMedis;
94
95   char buffer[256];
96   serializeJson(doc, buffer);
97   client.publish(mqtt_topic_volume, buffer);
98
99   // ---- Penentuan status ----
100   bool adaPenuh = false;
101   bool adaSetengah = false;
102
103   auto cekStatus = [&](long jarak) {
104     if (jarak != -1) { // hanya jika pembacaan valid
105       if (jarak <= 10) {
106         adaPenuh = true;
107       } else if (jarak > 10 && jarak <= 20) {

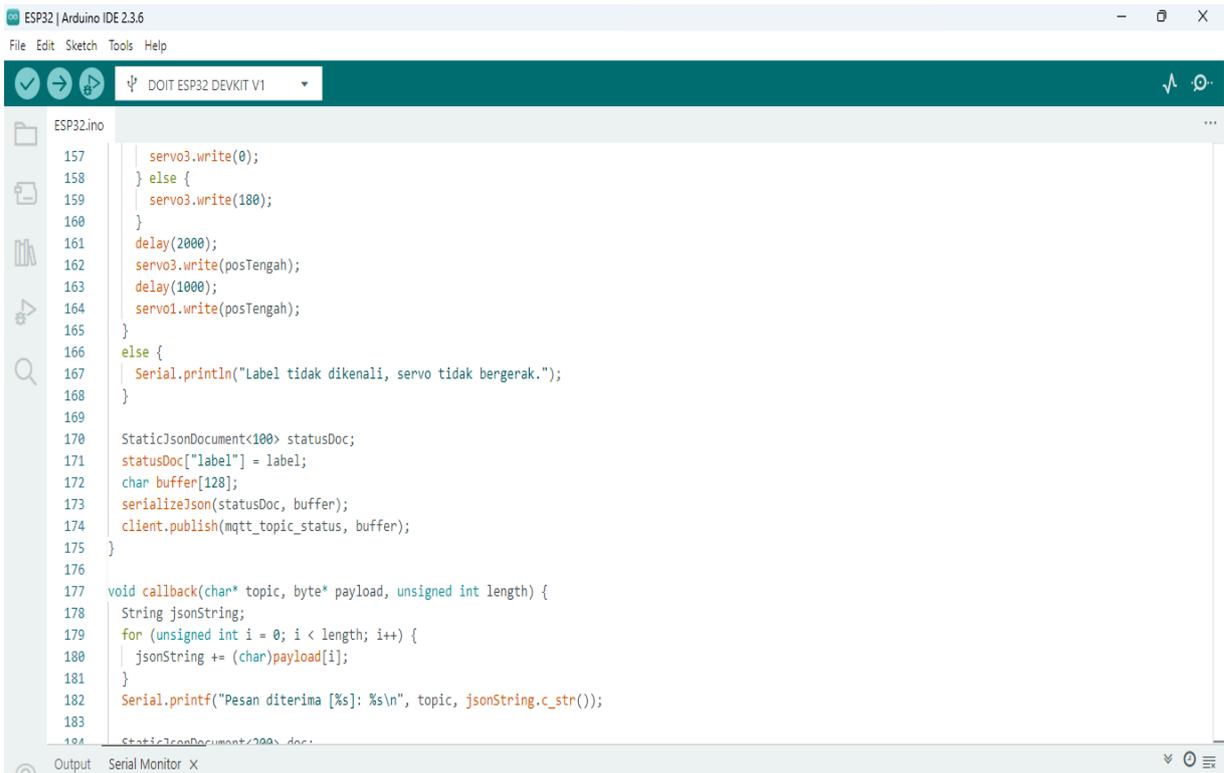
```

```
ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1

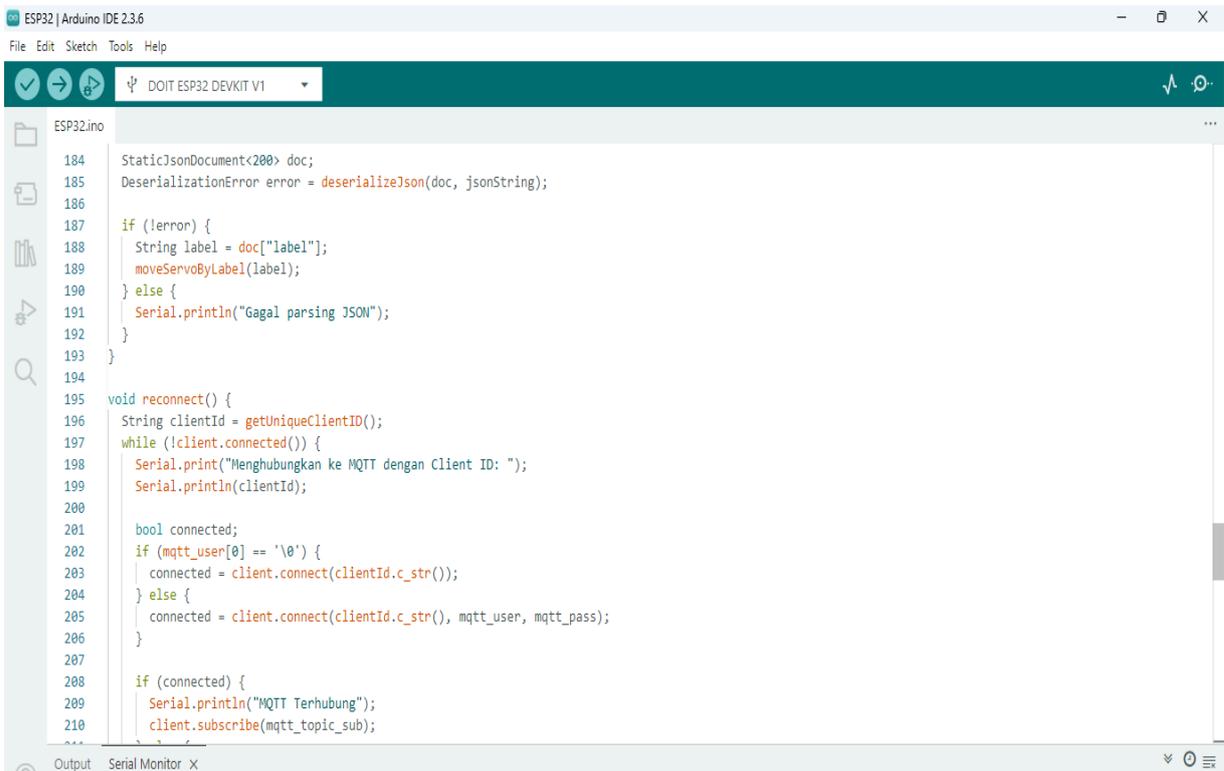
ESP32.ino
103 auto cekStatus = [](long jarak) {
104     if (jarak != -1) { // hanya jika pembacaan valid
105         if (jarak <= 10) {
106             adaPenuh = true;
107         } else if (jarak > 10 && jarak <= 20) {
108             adaSetengah = true;
109         }
110     }
111 };
112
113 cekStatus(jarakOrganik);
114 cekStatus(jarakAnorganik);
115 cekStatus(jarakB3);
116 cekStatus(jarakMedis);
117
118 // ---- Logika LED sesuai permintaan ----
119 // LED Merah: TIDAK dikontrol di sini (selalu berkedip via blinkLedMerah()).
120 // LED Kuning: PENUH
121 // LED Hijau : KOSONG/SETENGAH (masih bisa diisi)
122 if (adaPenuh) {
123     digitalWrite(LED_KUNING, HIGH);
124     digitalWrite(LED_HIJAU, LOW);
125     digitalWrite(BUZZER_PIN, HIGH); // buzzer saat penuh
126 } else {
127     digitalWrite(LED_KUNING, LOW);
128     digitalWrite(LED_HIJAU, HIGH); // kosong/setengah = masih bisa diisi
129     digitalWrite(BUZZER_PIN, LOW);
130 }
Output Serial Monitor X
```

```
ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1

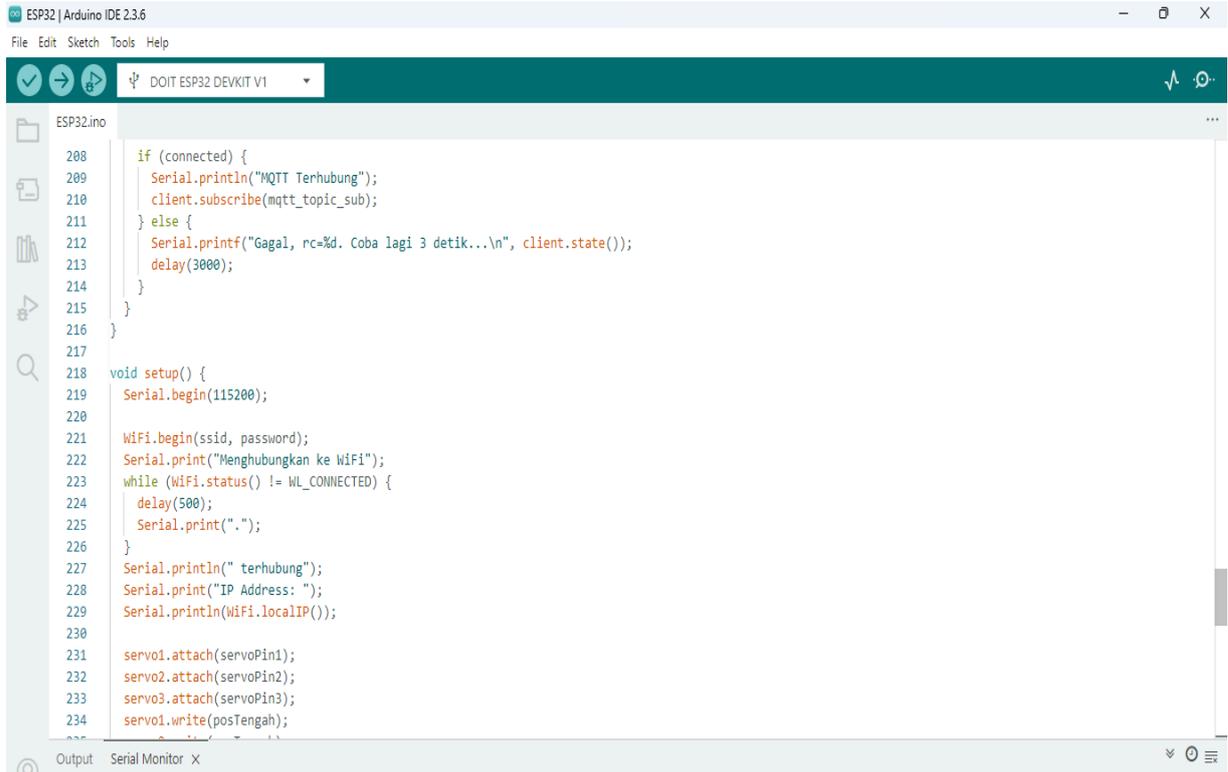
ESP32.ino
130 }
131 }
132
133 void moveServoByLabel(String label) {
134     lastLabel = label;
135     servo1.write(posTengah);
136     servo2.write(posTengah);
137     servo3.write(posTengah);
138     delay(500);
139
140     if (label == "Organik" || label == "Anorganik") {
141         servo1.write(0);
142         delay(2000);
143         if (label == "Organik") {
144             servo2.write(0);
145         } else {
146             servo2.write(180);
147         }
148         delay(2000);
149         servo2.write(posTengah);
150         delay(1000);
151         servo1.write(posTengah);
152     }
153     else if (label == "B3" || label == "Limbah Medis") {
154         servo1.write(180);
155         delay(2000);
156         if (label == "B3") {
157             servo3.write(0);
Output Serial Monitor X
```



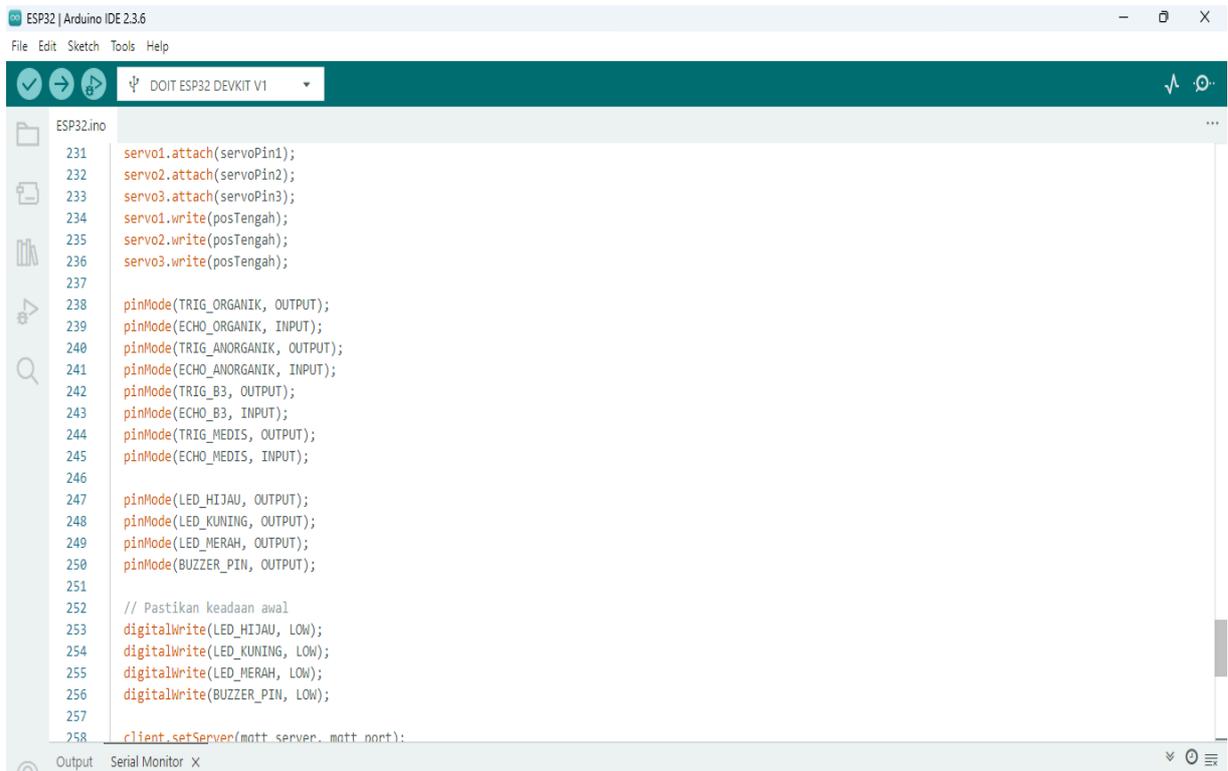
```
ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
ESP32.ino
157     | servo3.write(0);
158     } else {
159     | servo3.write(180);
160     }
161     delay(2000);
162     servo3.write(posTengah);
163     delay(1000);
164     servo1.write(posTengah);
165     }
166     else {
167     | Serial.println("Label tidak dikenali, servo tidak bergerak.");
168     }
169
170     StaticJsonDocument<100> statusDoc;
171     statusDoc["label"] = label;
172     char buffer[128];
173     serializeJson(statusDoc, buffer);
174     client.publish(mqtt_topic_status, buffer);
175     }
176
177     void callback(char* topic, byte* payload, unsigned int length) {
178     String jsonString;
179     for (unsigned int i = 0; i < length; i++) {
180     | jsonString += (char)payload[i];
181     }
182     Serial.printf("Pesan diterima [%s]: %s\n", topic, jsonString.c_str());
183
184     StaticJsonDocument<200> doc;
```



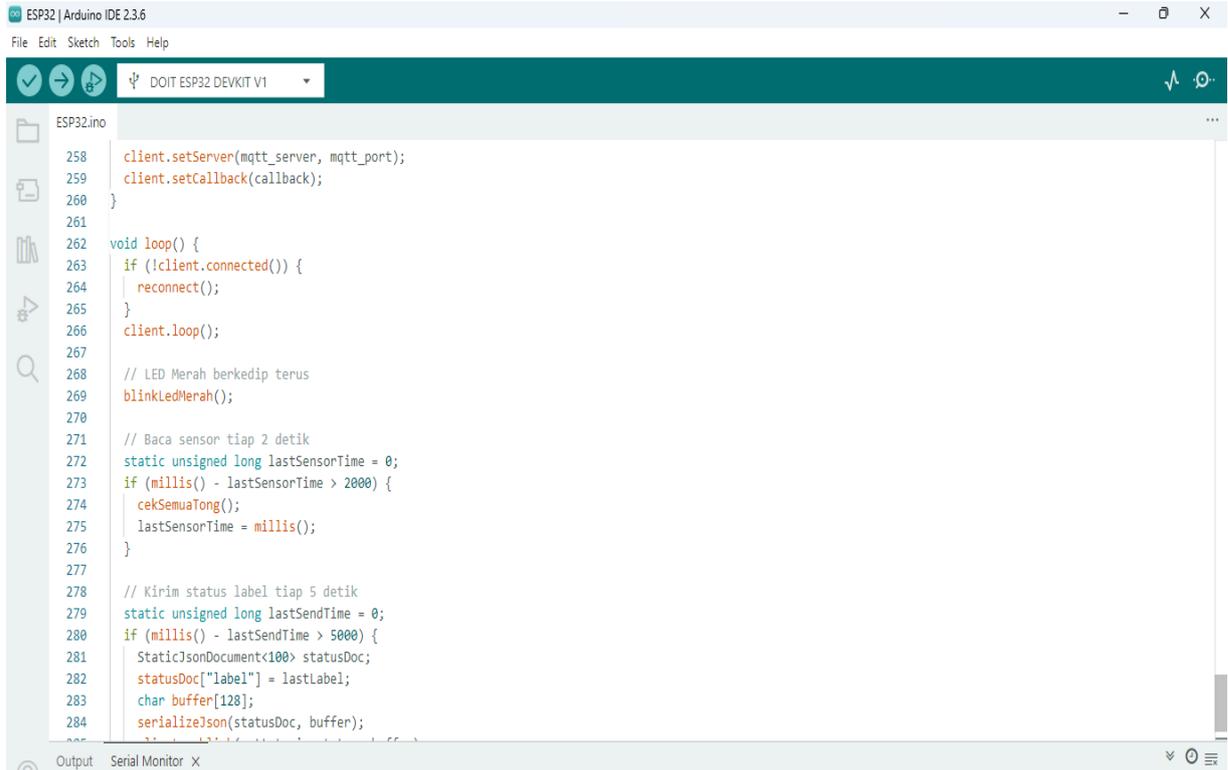
```
ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
ESP32.ino
184     StaticJsonDocument<200> doc;
185     DeserializationError error = deserializeJson(doc, jsonString);
186
187     if (!error) {
188     | String label = doc["label"];
189     | moveServoByLabel(label);
190     } else {
191     | Serial.println("Gagal parsing JSON");
192     }
193     }
194
195     void reconnect() {
196     String clientId = getUniqueId();
197     while (!client.connected()) {
198     | Serial.print("Menghubungkan ke MQTT dengan Client ID: ");
199     | Serial.println(clientId);
200
201     bool connected;
202     if (mqtt_user[0] == '\0') {
203     | connected = client.connect(clientId.c_str());
204     } else {
205     | connected = client.connect(clientId.c_str(), mqtt_user, mqtt_pass);
206     }
207
208     if (connected) {
209     | Serial.println("MQTT Terhubung");
210     | client.subscribe(mqtt_topic_sub);
211     }
```



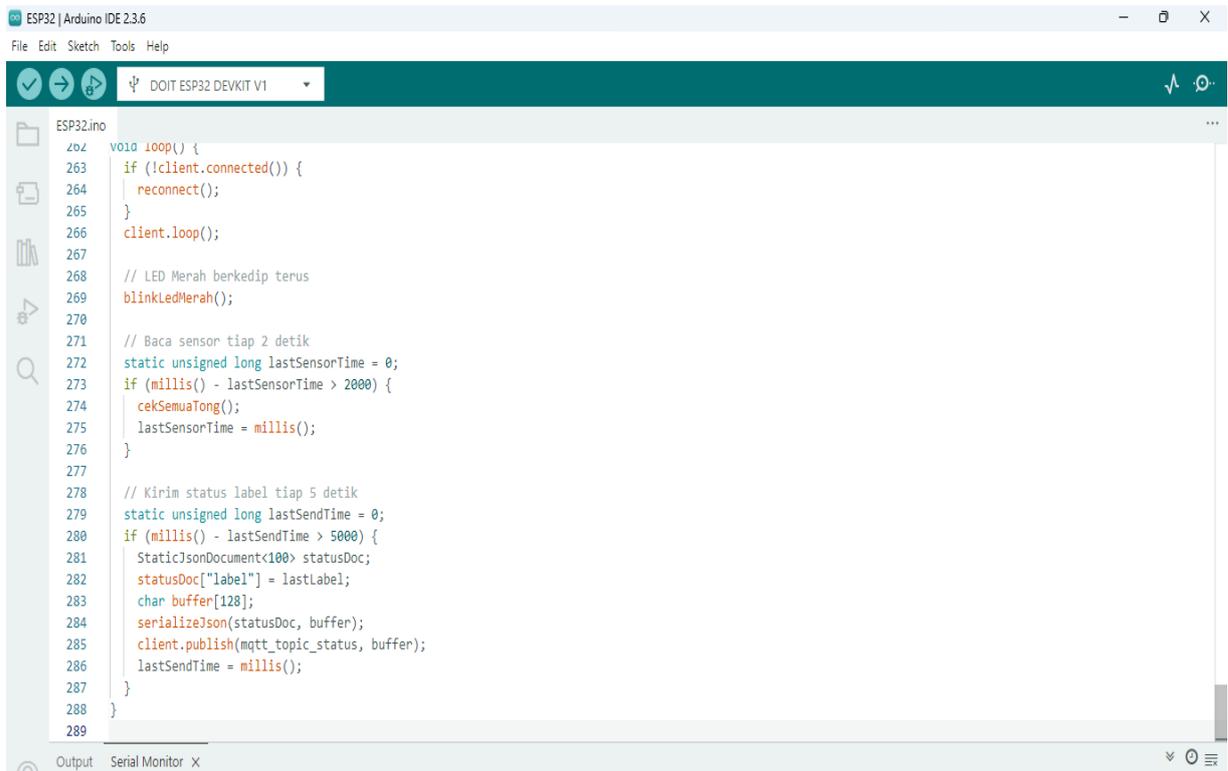
```
ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
ESP32.ino
208   if (connected) {
209     Serial.println("MQTT Terhubung");
210     client.subscribe(mqtt_topic_sub);
211   } else {
212     Serial.printf("Gagal, rc=%d. Coba lagi 3 detik...\n", client.state());
213     delay(3000);
214   }
215 }
216 }
217
218 void setup() {
219   Serial.begin(115200);
220
221   WiFi.begin(ssid, password);
222   Serial.print("Menghubungkan ke WiFi");
223   while (WiFi.status() != WL_CONNECTED) {
224     delay(500);
225     Serial.print(".");
226   }
227   Serial.println(" terhubung");
228   Serial.print("IP Address: ");
229   Serial.println(WiFi.localIP());
230
231   servo1.attach(servoPin1);
232   servo2.attach(servoPin2);
233   servo3.attach(servoPin3);
234   servo1.write(posTengah);
235
236   Output Serial Monitor X
```



```
ESP32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
DOIT ESP32 DEVKIT V1
ESP32.ino
231   servo1.attach(servoPin1);
232   servo2.attach(servoPin2);
233   servo3.attach(servoPin3);
234   servo1.write(posTengah);
235   servo2.write(posTengah);
236   servo3.write(posTengah);
237
238   pinMode(TRIG_ORGANIK, OUTPUT);
239   pinMode(ECHO_ORGANIK, INPUT);
240   pinMode(TRIG_ANORGANIK, OUTPUT);
241   pinMode(ECHO_ANORGANIK, INPUT);
242   pinMode(TRIG_B3, OUTPUT);
243   pinMode(ECHO_B3, INPUT);
244   pinMode(TRIG_MEDIS, OUTPUT);
245   pinMode(ECHO_MEDIS, INPUT);
246
247   pinMode(LED_HIJAU, OUTPUT);
248   pinMode(LED_KUNING, OUTPUT);
249   pinMode(LED_MERAH, OUTPUT);
250   pinMode(BUZZER_PIN, OUTPUT);
251
252   // Pastikan keadaan awal
253   digitalWrite(LED_HIJAU, LOW);
254   digitalWrite(LED_KUNING, LOW);
255   digitalWrite(LED_MERAH, LOW);
256   digitalWrite(BUZZER_PIN, LOW);
257
258   client.setServer(mqtt_server, mqtt_port);
259
260   Output Serial Monitor X
```

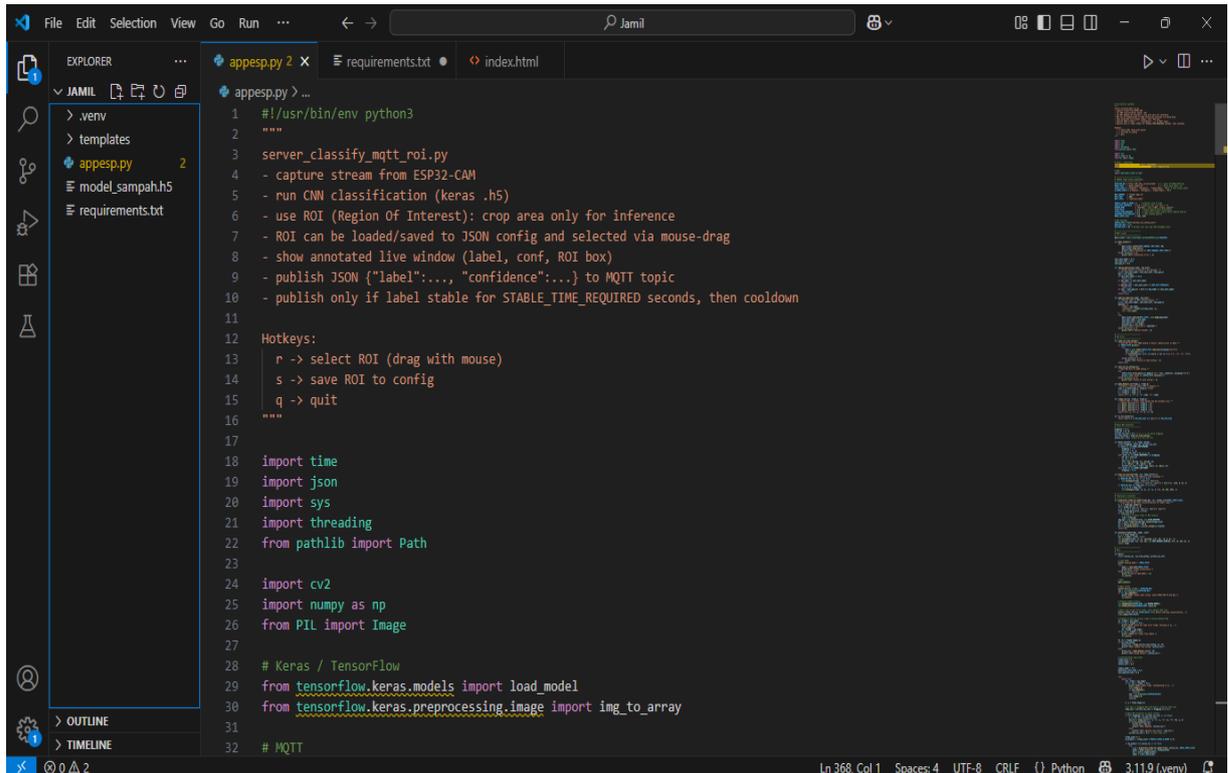


```
ESP32.ino
258 client.setServer(mqtt_server, mqtt_port);
259 client.setCallback(callback);
260 }
261
262 void loop() {
263   if (!client.connected()) {
264     reconnect();
265   }
266   client.loop();
267
268   // LED Merah berkedip terus
269   blinkLedMerah();
270
271   // Baca sensor tiap 2 detik
272   static unsigned long lastSensorTime = 0;
273   if (millis() - lastSensorTime > 2000) {
274     cekSemuaTong();
275     lastSensorTime = millis();
276   }
277
278   // Kirim status label tiap 5 detik
279   static unsigned long lastSendTime = 0;
280   if (millis() - lastSendTime > 5000) {
281     StaticJsonDocument<100> statusDoc;
282     statusDoc["label"] = lastLabel;
283     char buffer[128];
284     serializeJson(statusDoc, buffer);
285     client.publish(mqtt_topic_status, buffer);
286     lastSendTime = millis();
287   }
288 }
289
```



```
ESP32.ino
262 void loop() {
263   if (!client.connected()) {
264     reconnect();
265   }
266   client.loop();
267
268   // LED Merah berkedip terus
269   blinkLedMerah();
270
271   // Baca sensor tiap 2 detik
272   static unsigned long lastSensorTime = 0;
273   if (millis() - lastSensorTime > 2000) {
274     cekSemuaTong();
275     lastSensorTime = millis();
276   }
277
278   // Kirim status label tiap 5 detik
279   static unsigned long lastSendTime = 0;
280   if (millis() - lastSendTime > 5000) {
281     StaticJsonDocument<100> statusDoc;
282     statusDoc["label"] = lastLabel;
283     char buffer[128];
284     serializeJson(statusDoc, buffer);
285     client.publish(mqtt_topic_status, buffer);
286     lastSendTime = millis();
287   }
288 }
289
```

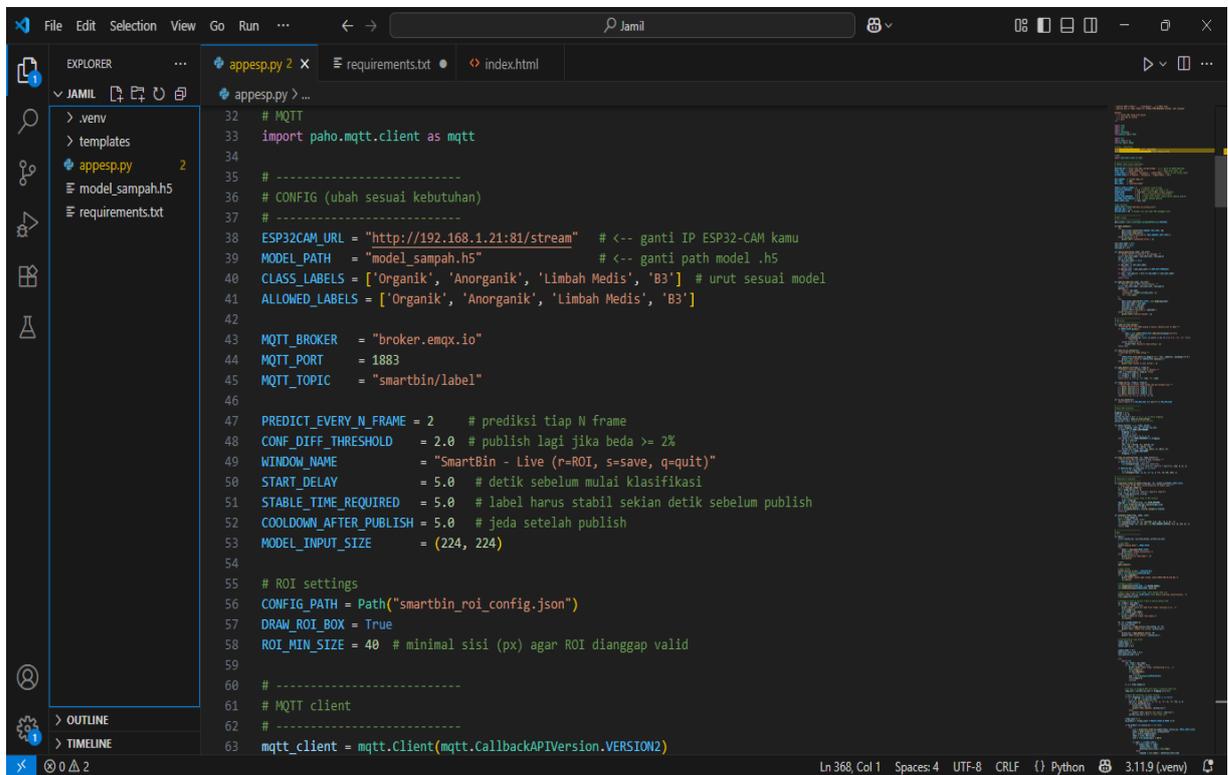
2. Script Visual Studio Code



```

1  #!/usr/bin/env python3
2  """
3  server_classify_mqtt_roi.py
4  - capture stream from ESP32-CAM
5  - run CNN classification (keras .h5)
6  - use ROI (Region Of Interest): crop area only for inference
7  - ROI can be loaded/saved to JSON config and selected via mouse-drag
8  - show annotated live window (label, conf, ROI box)
9  - publish JSON {"label":..., "confidence":...} to MQTT topic
10 - publish only if label stable for STABLE_TIME_REQUIRED seconds, then cooldown
11
12 Hotkeys:
13 r -> select ROI (drag with mouse)
14 s -> save ROI to config
15 q -> quit
16 """
17
18 import time
19 import json
20 import sys
21 import threading
22 from pathlib import Path
23
24 import cv2
25 import numpy as np
26 from PIL import Image
27
28 # Keras / TensorFlow
29 from tensorflow.keras.models import load_model
30 from tensorflow.keras.preprocessing.image import img_to_array
31
32 # MQTT

```



```

32 # MQTT
33 import paho.mqtt.client as mqtt
34
35 # -----
36 # CONFIG (ubah sesuai kebutuhan)
37 # -----
38 ESP32CAM_URL = "http://192.168.1.21:81/stream" # <- ganti IP ESP32-CAM kamu
39 MODEL_PATH = "model_sampah.h5" # <- ganti path model .h5
40 CLASS_LABELS = ['Organik', 'Anorganik', 'Limbah Medis', 'B3'] #urut sesuai model
41 ALLOWED_LABELS = ['Organik', 'Anorganik', 'Limbah Medis', 'B3']
42
43 MQTT_BROKER = "broker.emqx.io"
44 MQTT_PORT = 1883
45 MQTT_TOPIC = "smartbin/label"
46
47 PREDICT_EVERY_N_FRAME = 2 # prediksi tiap N frame
48 CONF_DIFF_THRESHOLD = 2.0 # publish lagi jika beda >= 2%
49 WINDOW_NAME = "SmartBin - Live (r=ROI, s=save, q=quit)"
50 START_DELAY = 5.0 # detik sebelum mulai klasifikasi
51 STABLE_TIME_REQUIRED = 5.0 # label harus stabil sekian detik sebelum publish
52 COOLDOWN_AFTER_PUBLISH = 5.0 # jeda setelah publish
53 MODEL_INPUT_SIZE = (224, 224)
54
55 # ROI settings
56 CONFIG_PATH = Path("smartbin_roi_config.json")
57 DRAW_ROI_BOX = True
58 ROI_MIN_SIZE = 40 # minimal sisi (px) agar ROI dianggap valid
59
60 # -----
61 # MQTT client
62 # -----
63 mqtt_client = mqtt.Client(mqtt.CallbackAPIVersion.VERSION2)

```

The screenshot shows a Visual Studio Code editor window with the following code in `appesp.py`:

```

64
65 def mqtt_connect():
66     try:
67         mqtt_client.connect(MQTT_BROKER, MQTT_PORT, 60)
68         mqtt_client.loop_start()
69         print(f"[MQTT] Connected to {MQTT_BROKER}:{MQTT_PORT}")
70     except Exception as e:
71         print("[MQTT] Connection error:", e)
72
73 last_sent_label = None
74 last_sent_conf = None
75 last_pub_ts = 0.0
76
77 def should_publish(new_label, new_conf):
78     """ Decide whether to publish a new message. """
79     global last_sent_label, last_sent_conf, last_pub_ts
80     now = time.time()
81     if last_sent_label is None:
82         return True
83     if new_label != last_sent_label:
84         return True
85     if abs(new_conf - last_sent_conf) >= CONF_DIFF_THRESHOLD:
86         return True
87     if (now - last_pub_ts) > 10.0 and new_label == last_sent_label:
88         return True
89     return False
90
91 def publish_label(new_label, new_conf):
92     """ Publish label to MQTT asynchronously. """
93     global last_sent_label, last_sent_conf, last_pub_ts
94     payload = {
95         "label": new_label,

```

The status bar at the bottom indicates: Ln 368, Col 1, Spaces: 4, UTF-8, CRLF, Python, 3.11.9 (venv).

The screenshot shows a Visual Studio Code editor window with the following code in `appesp.py`:

```

91 def publish_label(new_label, new_conf):
96     "confidence": round(float(new_conf), 2),
97     "ts": time.time()
98 }
99 try:
100     mqtt_client.publish(MQTT_TOPIC, json.dumps(payload))
101     last_sent_label = new_label
102     last_sent_conf = new_conf
103     last_pub_ts = time.time()
104     print(f"[MQTT] Published => {payload}")
105 except Exception as e:
106     print("[MQTT] Publish failed:", e)
107
108 # -----
109 # ROI utils
110 # -----
111 def load_roi_from_config():
112     """Load ROI dict from JSON config if exists. Returns dict or None."""
113     if CONFIG_PATH.exists():
114         try:
115             data = json.loads(CONFIG_PATH.read_text(encoding="utf-8"))
116             roi = data.get("roi")
117             if isinstance(roi, dict) and all(k in roi for k in ("x", "y", "w", "h")):
118                 return roi
119         except Exception as e:
120             print("[ROI] Failed to load config:", e)
121     return None
122
123 def save_roi_to_config(roi):
124     """Save ROI dict to JSON config."""
125     try:
126         CONFIG_PATH.write_text(json.dumps({"roi": roi}, indent=2), encoding="utf-8")

```

The status bar at the bottom indicates: Ln 368, Col 1, Spaces: 4, UTF-8, CRLF, Python, 3.11.9 (venv).

```

File Edit Selection View Go Run ... ← → Jamil
EXPLORER
  JAMIL
    .venv
    templates
    appesp.py 2
    model_sampah.hs
    requirements.txt
  OUTLINE
  TIMELINE
  0 2

appesp.py 2 X requirements.txt index.html
appesp.py > ...
123 def save_roi_to_config(roi):
127     print(f"ROI Saved to {CONFIG_PATH.resolve()}")
128     except Exception as e:
129         print("ROI Failed to save config:", e)
130
131 def make_default_roi(frame_w, frame_h):
132     """Create a centered square ROI as default."""
133     side = int(min(frame_w, frame_h) * 0.5)
134     x = (frame_w - side) // 2
135     y = (frame_h - side) // 2
136     return {"x": x, "y": y, "w": side, "h": side}
137
138 def clamp_roi(roi, frame_w, frame_h):
139     """Ensure ROI is within frame bounds and has minimum size."""
140     x = max(0, min(roi["x"], frame_w - 1))
141     y = max(0, min(roi["y"], frame_h - 1))
142     w = max(1, min(roi["w"], frame_w - x))
143     h = max(1, min(roi["h"], frame_h - y))
144     return {"x": x, "y": y, "w": w, "h": h}
145
146 def is_roi_valid(roi):
147     return roi["w"] >= ROI_MIN_SIZE and roi["h"] >= ROI_MIN_SIZE
148
149 # -----
150 # Mouse ROI selection
151 # -----
152 dragging = False
153 start_pt = (0, 0)
154 current_roi_rect = None # (x, y, w, h) while dragging
155 roi_from_config = load_roi_from_config()
156 active_roi = None # dict {"x", "y", "w", "h"}
157
Ln 368, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.9 (.venv)

```

```

File Edit Selection View Go Run ... ← → Jamil
EXPLORER
  JAMIL
    .venv
    templates
    appesp.py 2
    model_sampah.hs
    requirements.txt
  OUTLINE
  TIMELINE
  0 2

appesp.py 2 X requirements.txt index.html
appesp.py > ...
158 def mouse_cb(event, x, y, flags, param):
159     global dragging, start_pt, current_roi_rect
160     if event == cv2.EVENT_LBUTTONDOWN:
161         dragging = True
162         start_pt = (x, y)
163         current_roi_rect = (x, y, 1, 1)
164     elif event == cv2.EVENT_MOUSEMOVE and dragging:
165         x0, y0 = start_pt
166         x1, y1 = x, y
167         left, top = min(x0, x1), min(y0, y1)
168         w, h = abs(x1 - x0), abs(y1 - y0)
169         current_roi_rect = (left, top, max(1, w), max(1, h))
170     elif event == cv2.EVENT_LBUTTONUP:
171         dragging = False
172
173 def draw_roi_overlay(frame, roi, temp_rect=None):
174     """Draw active ROI and temporary drag rectangle."""
175     if DRAW_ROI_BOX and roi is not None:
176         cv2.rectangle(frame, (roi["x"], roi["y"]),
177                       (roi["x"] + roi["w"], roi["y"] + roi["h"]), (255, 0, 0), 2)
178     if DRAW_ROI_BOX and temp_rect is not None:
179         x, y, w, h = temp_rect
180         cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 255), 1)
181
182 # -----
183 # Preprocess & annotate
184 # -----
185 def preprocess_frame_for_model(frame_bgr, roi, target_size=MODEL_INPUT_SIZE):
186     """Crop frame to ROI then resize/normalize for model input."""
187     h, w = frame_bgr.shape[:2]
188     roi = clamp_roi(roi, w, h)
189     x, y, w, h = roi["x"], roi["y"], roi["w"], roi["h"]

```

```

File Edit Selection View Go Run ... Jamil
EXPLORER
  JAMIL
    .venv
    templates
    appesp.py 2
    model_sampah.h5
    requirements.txt
  OUTLINE
  TIMELINE
  0.2

appesp.py 2
requirements.txt
index.html
appesp.py > ...
185 def preprocess_frame_for_model(frame_bgr, roi, target_size=MODEL_INPUT_SIZE):
186     roi = clamp_roi(roi, w, h)
187     x, y, rw, rh = roi["x"], roi["y"], roi["w"], roi["h"]
188     crop = frame_bgr[y:y+rh, x:x+rw]
189     if crop.size == 0:
190         # fallback to whole frame if ROI invalid
191         crop = frame_bgr
192     img_rgb = cv2.cvtColor(crop, cv2.COLOR_BGR2RGB)
193     pil = Image.fromarray(img_rgb).resize(target_size)
194     arr = img_to_array(pil) / 255.0
195     arr = np.expand_dims(arr, axis=0).astype(np.float32)
196     return arr
197
198
199
200 def annotate_frame(frame, label, conf):
201     h, w = frame.shape[:2]
202     txt = f"{label} ({conf:.1f}%)"
203     cv2.rectangle(frame, (5, 5), (min(360, w-5), 40), (0, 0, 0), -1)
204     cv2.putText(frame, txt, (10, 28), cv2.FONT_HERSHEY_SIMPLEX, 0.8, (0, 255, 0), 2)
205     return frame
206
207 # -----
208 # Main
209 # -----
210 def main():
211     global active_roi, roi_from_config, current_roi_rect
212
213     # Load model
214     print("Loading model:", MODEL_PATH)
215     try:
216         model = load_model(MODEL_PATH)
217         print("Model loaded successfully.")
218     except Exception as e:

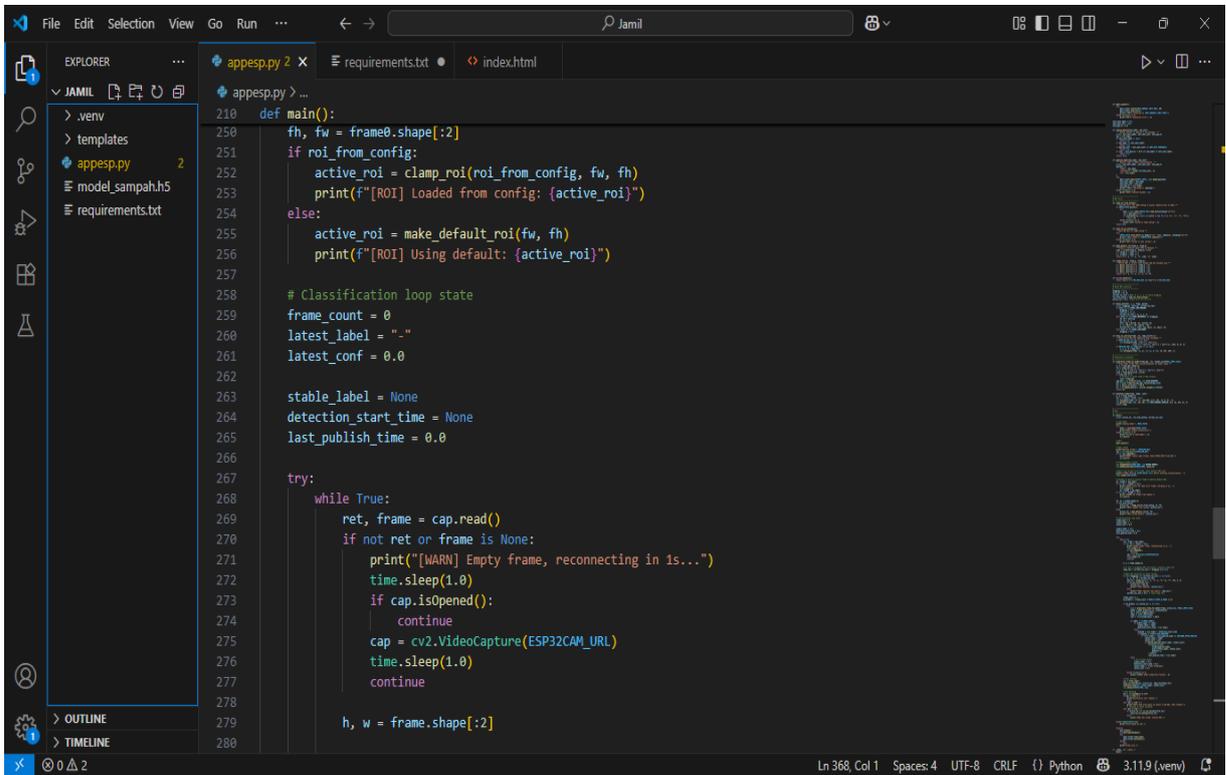
```

```

File Edit Selection View Go Run ... Jamil
EXPLORER
  JAMIL
    .venv
    templates
    appesp.py 2
    model_sampah.h5
    requirements.txt
  OUTLINE
  TIMELINE
  0.2

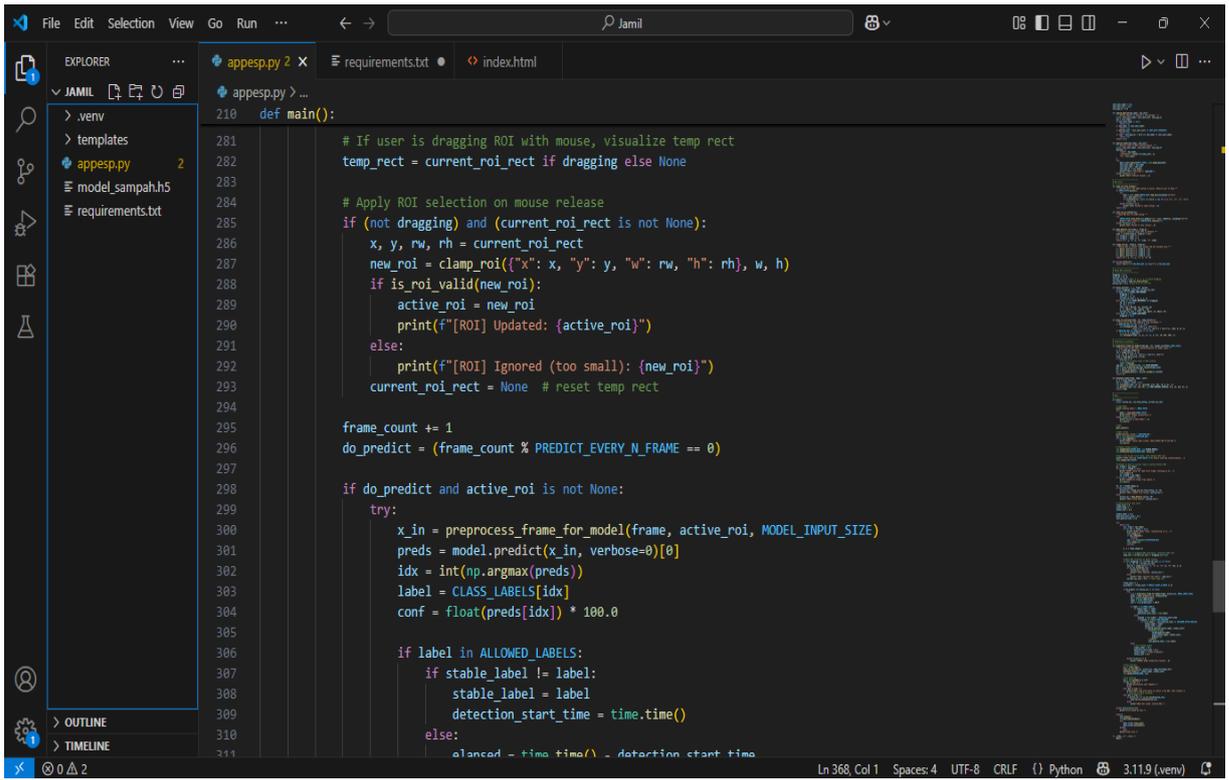
appesp.py 2
requirements.txt
index.html
appesp.py > ...
210 def main():
211     print("Failed to load model:", e)
212     sys.exit(1)
213
214     # MQTT
215     mqtt_connect()
216
217     # Open stream
218     print("Opening stream:", ESP32CAM_URL)
219     cap = cv2.VideoCapture(ESP32CAM_URL)
220     if not cap.isOpened():
221         print("ERROR: Cannot open stream. Check ESP32-CAM IP and URL.")
222         sys.exit(1)
223
224     # Prepare window & mouse
225     cv2.namedWindow(WINDOW_NAME, cv2.WINDOW_NORMAL)
226     cv2.setMouseCallback(WINDOW_NAME, mouse_cb)
227
228     # Wait a bit to get first frame, infer default ROI size
229     print(f"[INFO] Waiting {START_DELAY:.0f}s before starting classification..")
230     time.sleep(START_DELAY)
231
232     # Attempt to grab an initial frame to define default ROI
233     ok, frame0 = cap.read()
234     if not ok or frame0 is None:
235         print("[WARN] Could not read first frame; retrying in 1s...")
236         time.sleep(1.0)
237         ok, frame0 = cap.read()
238     if not ok or frame0 is None:
239         print("[ERROR] No frames from camera.")
240         sys.exit(1)
241

```



```
210 def main():
250     fh, fw = frame0.shape[:2]
251     if roi_from_config:
252         active_roi = clamp_roi(roi_from_config, fw, fh)
253         print(f"ROI Loaded from config: {active_roi}")
254     else:
255         active_roi = make_default_roi(fw, fh)
256         print(f"ROI Using default: {active_roi}")
257
258     # Classification loop state
259     frame_count = 0
260     latest_label = "-"
261     latest_conf = 0.0
262
263     stable_label = None
264     detection_start_time = None
265     last_publish_time = 0.0
266
267     try:
268         while True:
269             ret, frame = cap.read()
270             if not ret or frame is None:
271                 print("[WARN] Empty frame, reconnecting in 1s...")
272                 time.sleep(1.0)
273                 if cap.isOpened():
274                     continue
275                 cap = cv2.VideoCapture(ESP32CAM_URL)
276                 time.sleep(1.0)
277                 continue
278
279     h, w = frame.shape[:2]
```

Ln 368, Col 1 Spaces: 4 UTF-8 CRLF Python 3.11.9 (venv)



The image shows a code editor window with a dark theme. The Explorer panel on the left shows a project structure with files like `requirements.txt`, `index.html`, and `appesp.py`. The main editor displays Python code for a `main()` function. The code handles ROI dragging, applies selection on mouse release, and performs frame prediction using a model. It includes logic for stable labeling and detection start time tracking.

```
210 def main():
211
212     # If user is dragging ROI with mouse, visualize temp rect
213     temp_rect = current_roi_rect if dragging else None
214
215     # Apply ROI selection on mouse release
216     if (not dragging) and (current_roi_rect is not None):
217         x, y, rw, rh = current_roi_rect
218         new_roi = clamp_roi({"x": x, "y": y, "w": rw, "h": rh}, w, h)
219         if is_roi_valid(new_roi):
220             active_roi = new_roi
221             print(f"ROI Updated: {active_roi}")
222         else:
223             print(f"ROI Ignored (too small): {new_roi}")
224             current_roi_rect = None # reset temp rect
225
226     frame_count += 1
227     do_predict = (frame_count % PREDICT_EVERY_N_FRAME == 0)
228
229     if do_predict and active_roi is not None:
230         try:
231             x_in = preprocess_frame_for_model(frame, active_roi, MODEL_INPUT_SIZE)
232             preds = model.predict(x_in, verbose=0)[0]
233             idx = int(np.argmax(preds))
234             label = CLASS_LABELS[idx]
235             conf = float(preds[idx]) * 100.0
236
237             if label in ALLOWED_LABELS:
238                 if stable_label != label:
239                     stable_label = label
240                     detection_start_time = time.time()
241             else:
242                 elapsed = time.time() - detection_start_time
```

```

File Edit Selection View Go Run ... Jamil
EXPLORER
  JAMIL
    .venv
    templates
    appesp.py 2
    model_sampah.h5
    requirements.txt
  OUTLINE
  TIMELINE
  appesp.py 2 X requirements.txt index.html
  appesp.py > ...
210 def main():
311     elapsed = time.time() - detection_start_time
312     if elapsed >= STABLE_TIME_REQUIRED:
313         if (time.time() - last_publish_time) >= COOLDOWN_AFTER_PUBLISH:
314             latest_label = label
315             latest_conf = conf
316             if should_publish(latest_label, latest_conf):
317                 threading.Thread(
318                     target=publish_label,
319                     args=(latest_label, latest_conf),
320                     daemon=True
321                 ).start()
322                 last_publish_time = time.time()
323             else:
324                 # not allowed labels
325                 stable_label = None
326                 detection_start_time = None
327                 latest_label = "Tidak Terdeteksi"
328                 latest_conf = 0.0
329
330     except Exception as e:
331         print("[ERROR] model prediction failed:", e)
332
333     # Draw overlays
334     vis = frame.copy()
335     draw_roi_overlay(vis, active_roi, temp_rect=temp_rect)
336     annotate_frame(vis, latest_label, latest_conf)
337     cv2.imshow(WINDOW_NAME, vis)
338
339     # Key handling
340     key = cv2.waitKey(1) & 0xFF
341     if key == ord('a'):

```

```

File Edit Selection View Go Run ... Jamil
EXPLORER
  JAMIL
    .venv
    templates
    appesp.py 2
    model_sampah.h5
    requirements.txt
  OUTLINE
  TIMELINE
  appesp.py 2 X requirements.txt index.html
  appesp.py > ...
210 def main():
339     # Key handling
340     key = cv2.waitKey(1) & 0xFF
341     if key == ord('q'):
342         print("Quitting by user request.")
343         break
344     elif key == ord('r'):
345         print("[ROI] Drag with mouse to select a new ROI, then release.")
346         # (handled by mouse callback)
347     elif key == ord('s'):
348         if active_roi and is_roi_valid(active_roi):
349             save_roi_to_config(active_roi)
350         else:
351             print("[ROI] Not saved: invalid ROI.")
352
353     except KeyboardInterrupt:
354         print("Interrupted by user.")
355
356     finally:
357         cap.release()
358         cv2.destroyAllWindows()
359     try:
360         mqtt_client.loop_stop()
361         mqtt_client.disconnect()
362     except:
363         pass
364     print("Clean exit.")
365
366 if __name__ == "__main__":
367     main()
368

```

3. Surat Penetapan Pembimbing



MAJELIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PUSAT MUHAMMADIYAH
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 89/SK/BAN-PT/Akred/PT/09/2019
 Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003
<http://www.umsumedan.ac.id> info@umsumedan.ac.id [umsumedan](https://www.facebook.com/umsumedan) [umsumedan](https://www.instagram.com/umsumedan) [umsumedan](https://www.youtube.com/umsumedan) [umsumedan](https://www.linkedin.com/umsumedan)

PENETAPAN DOSEN PEMBIMBING
PROPOSAL/SKRIPSI MAHASISWA
NOMOR : 492/IL3-AU/UMSU-09/F/2025

Assalamu'alaikum Warahmatullahi Wabarakatuh

Dekan Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Muhammadiyah Sumatera Utara, berdasarkan Persetujuan permohonan judul penelitian Proposal / Skripsi dari Ketua / Sekretaris.

Program Studi : Teknologi Informasi
Pada tanggal : 12 Maret 2025

Dengan ini menetapkan Dosen Pembimbing Proposal / Skripsi Mahasiswa.

Nama : JAMILUDDIN TANJUNG
NPM : 2109020140
Semester : VIII (Delapan)
Program studi : Teknologi Informasi
Judul Proposal / Skripsi : Smart Waste Management System: Inovasi Tong Sampah Dengan Mekanisme Otomatisasi Dan Keamanan Modern Dengan Teknologi IOT

Dosen Pembimbing : Yohanni Syahra, S.Si.,M.Kom.

Dengan demikian di izinkan menulis Proposal / Skripsi dengan ketentuan

1. Penulisan berpedoman pada buku panduan penulisan Proposal / Skripsi Fakultas Ilmu Komputer dan Teknologi Informasi UMSU
2. Pelaksanaan Sidang Skripsi harus berjarak 3 bulan setelah dikeluarkannya Surat Penetapan Dosen Pembimbing Skripsi.
3. **Proyek Proposal / Skripsi dinyatakan " BATAL " bila tidak selesai sebelum Masa Kadaluarsa tanggal : 12 Maret 2026**
4. Revisi judul.....

Wassalamu'alaikum Warahmatullahi Wabarakatuh.

Ditetapkan di : Medan
 Pada Tanggal : 12 Ramadhan 1446 H
 12 Maret 2025M




 Dekan
Dr. Al-Kha Warizmi, M.Kom.
 NIDN : 0127099201

Cc. File



4. Lembar Bimbingan Skripsi



UMSU

Unggul | Cerdas | Terpercaya

www.umsumedan.ac.id

MAJLIS PENDIDIKAN TINGGI PENELITIAN & PENGEMBANGAN PIMPINAN PESAT MUHAMMADIYAH

UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA

FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

UMSU Terakreditasi A Berdasarkan Keputusan Badan Akreditasi Nasional Perguruan Tinggi No. 53/SK/BAN-PT/Akred/PT/III/2019

Pusat Administrasi: Jalan Mukhtar Basri No. 3 Medan 20238 Telp. (061) 6622400 - 66224567 Fax. (061) 6625474 - 6631003

<https://fakultas.umsumedan.ac.id>

661@umsumedan.ac.id

umsumedan

umsumedan

umsumedan

umsumedan

Berita Acara Pembimbingan Skripsi

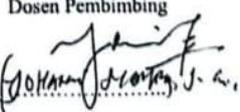
Nama Mahasiswa : Jamiluddin Tanjung Program Studi : Teknologi Informasi
 NPM : 2109020140 Konsentrasi : -
 Nama Dosen Pembimbing : Yohanni Syahra S.Si. M.kom Judul Penelitian : Smart waste management system inovasi tong sampah IoT Dengan mekanisme otomatisasi dan keamanan modern

Item	Hasil Evaluasi	Tanggal	Paraf Dosen
BAB 1	Tambahkan bagian pustaka!	19/3-2025	ya
BAB 2	Perbaiki bagian Perhitungan!	20/3-2025	ya
BAB 3	Perbaiki pustaka ACC sesuai proposal!	20/5-2025	ya
BAB 4	Perbaiki HM pada BAB 4!	19/2-2025	ya
BAB 5	Perbaiki Nama mata Pelajaran!	21/09-2025	ya
Uraian	Uraian Uraian - Uraian!	1/0-2025	ya
Metode	Perbaiki metode yang kurang!	19/0-2025	ya
Alasan	ACC sesuai proposal	21/0-2025	ya

Diketahui oleh :
 Ketua Program Studi
 Teknologi Informasi
 (......)

Medan, 21-08-2025

Disetujui oleh :
 Dosen Pembimbing


 (Yohanni Syahra S.Si. M.kom)



5. Letter Of Acceptance



LETTER OF ACCEPTANCE (LoA)

Dear Mr/Ms/Br

Jamiluddin Tanjung, Yohanni Syahra

In

Place

We hereby inform you that the manuscript with the following details has been declared accepted for publication in the Tsabit Journal of Computer Science, ISSN: 3062-8504 (Online), Volume 1 Number 2 December 2025 Edition.

Title	Smart Waste Management System: Innovation of IoT Trash Can With Modern Automation and Security Mechanism
Author	Jamiluddin Tanjung, Yohanni Syahra
Correspondent Email	yohannisyahra@umsu.ac.id

Thus we have created this certificate so that it can be used as necessary.



Medan, 29 September 2025

Editor in Chief

Oris Krianto Sulaiman, S.T., M.Kom.

6. Hasil Cek Turnitin

PROPOSAL_JAMILUDDIN_TANJUNG_baru21_1-1758294780659			
ORIGINALITY REPORT			
11 %	9 %	4 %	5 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	Submitted to Universitas Muhammadiyah Sumatera Utara Student Paper		1 %
2	text-id.123dok.com Internet Source		1 %
3	123dok.com Internet Source		<1 %
4	docplayer.info Internet Source		<1 %
5	Luntungan Stephen Pieters. "Development of Automatic Waste Classification System using CNN-Based Deep Learning to Support Smart Waste Management", INOVTEK Polbeng - Seri Informatika, 2025 Publication		<1 %
6	Submitted to Fakultas Teknik Student Paper		<1 %
7	dspace.uii.ac.id Internet Source		<1 %

8	portaldata.org Internet Source	<1 %
9	Taupik Hidayat, Lidiya, Aris Sudioanto, Nurhidayati. "Pendampingan Pembuatan Bak Sampah Otomatis Berbasis Teknologi Internet of Things (IoT) Untuk Kantor Desa Aikmel Timur", Jurnal Teknologi Informasi untuk Masyarakat, 2024 Publication	<1 %
10	eprints.umm.ac.id Internet Source	<1 %
11	Submitted to Universitas Diponegoro Student Paper	<1 %
12	journal.unpacti.ac.id Internet Source	<1 %
13	Submitted to Universitas Brawijaya Student Paper	<1 %
14	digilib.unila.ac.id Internet Source	<1 %
15	Submitted to Telkom University Student Paper	<1 %
16	repository.poltekbangplg.ac.id Internet Source	<1 %
17	kc.umn.ac.id Internet Source	<1 %

18	repository.umsu.ac.id Internet Source	<1 %
19	Doly Ilham Saputra Huta Julu, Dewi Nurdiyah. "KLASIFIKASI SAMPAH ORGANIK DAN NON ORGANIK MENGGUNAKAN TRANSFER LEARNING", Jurnal Transformatika, 2025 Publication	<1 %
20	Submitted to Universitas Gadjah Mada Student Paper	<1 %
21	eprints.uty.ac.id Internet Source	<1 %
22	repo.ubibanyuwangi.ac.id Internet Source	<1 %
23	Nicholas Bagus Pamungkas, Agus Suhendar. "Penerapan Metode Convolutional Neural Network pada Sistem Klasifikasi Penyakit Tanaman Apel berdasarkan Citra Daun", Edumatic: Jurnal Pendidikan Informatika, 2024 Publication	<1 %
24	Submitted to iGroup Student Paper	<1 %
25	jurnal.pustakagalerimandiri.co.id Internet Source	<1 %
26	Submitted to UNIVERSITAS BUDI LUHUR Student Paper	<1 %

		<1 %
27	adoc.pub Internet Source	<1 %
28	artikelpendidikan.id Internet Source	<1 %
29	repository.its.ac.id Internet Source	<1 %
30	Ibnu Zahy' Atha Illah, Wahyu Syaifullah Jauharis Sapu, Aviolla Terza Damaliana. "Implementasi Metode Klasifikasi LightGBM dan Analisis Survival dalam Memprediksi Pelanggan Churn", Jurnal Komtika (Komputasi dan Informatika), 2024 Publication	<1 %
31	Submitted to Universitas Mercu Buana Student Paper	<1 %
32	Submitted to Universitas Muslim Indonesia Student Paper	<1 %
33	repository.wima.ac.id Internet Source	<1 %
34	Fahril Maula Tanzil Huda, Yusril Ali Riza Pratama, Fauzan Ra'is Saputra, Rizanurfadli Hadiazzaka, Ardy Seto Priambodo. "PENERAPAN KINEMATIKA TERBALIK PADA	<1 %

ROBOT LENGAN LIMA SENDI (5 DOF) DENGAN
CITRA DIGITAL", Jurnal Informatika dan Teknik
Elektro Terapan, 2025

Publication

35	Submitted to Perpustakaan Student Paper	<1 %
36	blog.unmaha.ac.id Internet Source	<1 %
37	repository.ub.ac.id Internet Source	<1 %
38	eprints.polbeng.ac.id Internet Source	<1 %
39	ojs.trigunadharma.ac.id Internet Source	<1 %
40	rcf-indonesia.org Internet Source	<1 %
41	Roynal Kadepi, Syamsul Bahri, Suhardi Suardi. "SISTEM MONITORING DAN PENGONTROLAN PADA BUDI DAYA IKAN MAS BERBASIS INTERNET OF THINGS (IoT)", Coding Jurnal Komputer dan Aplikasi, 2022 Publication	<1 %
42	Submitted to Universitas Islam Riau Student Paper	<1 %
43	journals.telkomuniversity.ac.id Internet Source	

		<1 %
44	jurnal.borneo.ac.id Internet Source	<1 %
45	repositori.uma.ac.id Internet Source	<1 %
46	repository.itelkom-pwt.ac.id Internet Source	<1 %
47	restikom.nusaputra.ac.id Internet Source	<1 %
48	Andayyani , F.TriasPontia , Tedy Rismawan. "PROTOTYPE SISTEM KONTROL PINTU AIR OTOMATIS PADA SALURAN IRIGASI LAHAN PERTANIAN PADI DAERAH ALIRAN SUNGAI (DAS) AIR ASIN", Coding Jurnal Komputer dan Aplikasi, 2016 Publication	<1 %
49	Indah Sri Lestari, Jumadi Jumadi, Nur Lukman. "IMPLEMENTASI CONVOLUTIONAL NEURAL NETWORK DENGAN PRE-TRAINED MODEL MOBILENETV2 UNTUK DETEKSI KOLESTEROL", Rabit : Jurnal Teknologi dan Sistem Informasi Univrab, 2024 Publication	<1 %
50	Yudisman Ferdinan Bili, Tundo, Nandang Sutisna, Atsilah Daini Putri, Dita Tri Yuliantoro,	<1 %

Laily Nurmayanti. "Prediksi Motif Batik dengan Menggunakan Metode Gabor Filter Convolution Neural Network", Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2025

Publication

51	ejournal.ukrida.ac.id Internet Source	<1 %
52	etheses.uin-malang.ac.id Internet Source	<1 %
53	jurnalunibi.unibi.ac.id Internet Source	<1 %
54	pt.scribd.com Internet Source	<1 %
55	repositori.unsil.ac.id Internet Source	<1 %
56	repository.uph.edu Internet Source	<1 %
57	www.coursehero.com Internet Source	<1 %
58	Andi Sitti Astaty, Andi Sitti Asti Suriaty, Adji Ananta Sukma. "Pemanfaatan Bank Sampah untuk Meningkatkan Kesadaran Lingkungan di Kelurahan Gunung Panjang, Samarinda	<1 %

Seberang", Jurnal Teknik Industri Terintegrasi, 2025

Publication

59	<p>Muhammad Arkan Fauzi, Setyawan Ajie Sukarno. "PENGEMBANGAN SISTEM PENDETEKSI KEBOCORAN GAS LPG BERBASIS IOT: INTEGRASI SENSOR MQ-02 DAN DHT11 UNTUK PEMANTAUAN REAL-TIME", Jurnal Informatika dan Teknik Elektro Terapan, 2025</p>	<1 %
60	<p>Rinto Priambodo. "Rekam Medis Elektronik Menggunakan Sistem Penyimpanan Foto Intraoral Gigi untuk Aplikasi Teledentistry berbasis Internet of Things", INOVTEK Polbeng - Seri Informatika, 2019</p>	<1 %
61	<p>annisabillaa.wordpress.com</p>	<1 %
62	<p>doku.pub</p>	<1 %
63	<p>es.scribd.com</p>	<1 %
64	<p>issuu.com</p>	<1 %
65	<p>repository.ar-raniry.ac.id</p>	<1 %

66	repository.teknokrat.ac.id Internet Source	<1 %
67	Submitted to unimal Student Paper	<1 %
68	unpar.ac.id Internet Source	<1 %
69	www.researchgate.net Internet Source	<1 %
70	Satrio Bagus Prakoso. "Rancang Bangun Smart Doorbell Berbasis Internet of Things (IoT) Menggunakan ESP32-CAM via Telegram", YASIN, 2025 Publication	<1 %
71	Wd Elda Zuwelda Yama. "Monitoring Volume Tempat Sampah Di Kelurahan Wosi Madu Raja Menggunakan Mikrokontroler Nodemcu Esp 8266 Berbasis Internet Of Things", RIGGS: Journal of Artificial Intelligence and Digital Business, 2025 Publication	<1 %

Exclude quotes Off

Exclude matches Off

Exclude bibliography On