



TUGAS AKHIR (EE33410)

**SISTEM ALARM BERBASIS SENSOR JARAK DAN
SENSOR BERAT UNTUK PENGUKURAN VOLUME
SAMPAH PADA PROTOTIPE KAPAL**

**Ahmad Valentino
NRP. 0423030024**

**Dosen Pembimbing
Urip Mudjiono, S.T., M.T.**

**PROGRAM STUDI D3 TEKNIK KELISTRIKAN KAPAL
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
2026**

(Halaman ini sengaja dikosongkan)



PPNS POLITEKNIK
PERKAPALAN
NEGERI SURABAYA

TUGAS AKHIR (EE33410)

**SISTEM ALARM BERBASIS SENSOR JARAK DAN
SENSOR BERAT UNTUK PENGUKURAN VOLUME
SAMPAH PADA PROTOTIPE KAPAL**

**Ahmad Valentino
NRP. 0423030024**

**Dosen Pembimbing
Urip Mudjiono, S.T., M.T.**

**PROGRAM STUDI D3 TEKNIK KELISTRIKAN KAPAL
JURUSAN TEKNIK KELISTRIKAN KAPAL
POLITEKNIK PERKAPALAN NEGERI SURABAYA
2026**

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

TUGAS AKHIR

**SISTEM ALARM BERBASIS SENSOR JARAK DAN SENSOR BERAT
UNTUK PENGUKURAN VOLUME SAMPAH PADA PROTOTIPE
KAPAL**

Disusun Oleh:
Ahmad Valentino
0423030024

Diajukan Untuk Memenuhi Salah Satu Syarat Kelulusan
Program Studi D3 Teknik Kelistrikan Kapal
Jurusan Teknik Kelistrikan Kapal
POLITEKNIK PERKAPALAN NEGERI SURABAYA

Disetujui oleh Tim Penguji Tugas Akhir Tanggal Ujian: 15 Juli 2026
Periode Wisuda: Oktober 2026

Menyetujui,

Dosen Penguji

1. **li⁺ Munadhif, S.ST., M.T.**
2. **Dr. Catur Rakhmad H, S.T., M.T.**
3. **Riko Satrya Fajar J. P, S.T., M.T.**
4. **Urip Mudjiono, S.T., M.T.**

NIDN/NUPTK

(0010079102)
(0025027303)
(6859776677130162)
(0027056803)

Tanda Tangan

(.....)
(.....)
(.....)
(.....)

Dosen Pembimbing

1. **Urip Mudjiono, S.T., M.T.**

NIDN

(0027056803)

Tanda Tangan

(.....)

Menyetujui
Ketua Jurusan,

Isa Rachman, S.T., M.T.
NIP. 198008162008121001

Mengetahui
Koordinator Program Studi,

li⁺ Munadhif, S.ST., M.T.
NIP. 199107102018031001



(Halaman ini sengaja dikosongkan)



PERNYATAAN BEBAS PLAGIAT

No. : F.WD I. 021
Date : 22 Juli 2026
Rev. : 01
Page : 1 dari 1

Yang bertandatangan dibawah ini :

Nama : Ahmad Valentino
NRP : 0423030024
Jurusan/Prodi : Teknik Kelistrikan Kapal/ D3 Teknik Kelistrikan Kapal

Dengan ini menyatakan dengan sesungguhnya bahwa :

Tugas Akhir yang akan saya kerjakan dengan judul :

SISTEM ALARM BERBASIS SENSOR JARAK DAN SENSOR BERAT UNTUK PENGUKURAN VOLUME SAMPAH PADA PROTOTYPE KAPAL

Adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain.

Apabila dikemudian hari terbukti terdapat plagiat dalam karya ilmiah tersebut,
maka saya bersedia menerima sanksi sesuai ketentuan peraturan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Surabaya, 22 Juli 2026

Yang membuat pernyataan,

(Ahmad Valentino)

0423030024

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur atas kehadiran Allah SWT yang senantiasa melindungi, memberikan pertolongan, serta memberi rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “ **Sistem Alarm Berbasis Sensor Jarak Dan Sensor Berat Untuk Pengukuran Volume Sampah Pada Prototipe Kapal** ”.

Selama penyusunan Tugas Akhir ini penulis telah mendapatkan banyak bantuan dari berbagai pihak, baik secara langsung maupun tidak langsung. Sebagai apresiasi, ucapan terima kasih yang tulus dan ikhlas disampaikan kepada:

1. Allah SWT yang memberikan segala kemudahan, kelancaran, kekuatan, serta kenikmatan yang luar biasa besarnya dalam proses pengerjaan dan penyelesaian Tugas Akhir ini sehingga Tugas Akhir ini dapat terselesaikan dengan baik.
2. Bapak Edi Sopandi dan Ibu Iti Warniti selaku orang tua dari penulis yang sangat berpengaruh dalam pengerjaan Tugas Akhir ini baik dukungan secara materi, pemberian semangat serta kasih sayang dan doa tulus yang tiada hentinya dalam proses penyelesaian Tugas Akhir ini.
3. Penulis mengucapkan terima kasih kepada kakak tercinta atas doa, dukungan, motivasi, serta bantuan finansial yang telah diberikan sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
4. Bapak Rachmad Tri Soelistijono, S.T., M.T. Selaku Direktur Politeknik Perkapalan Negeri Surabaya.
5. Bapak Isa Rachman, S.T., M.T. Selaku Ketua Jurusan Teknik Kelistrikan Kapal di Politeknik Perkapalan Negeri Surabaya.
6. Bapak Dimas Pristovani Riananda, S.ST., M.T. Selaku koordinator Tugas Akhir Teknik Kelistrikan Kapal.
7. Bapak Urip Mudjiono, S.T.,M.T. Selaku dosen pembimbing , yang telah memberikan banyak masukan, mengarahkan dan membimbing kepada penulis selama pengerjaan Tugas Akhir ini.

8. Bapak dan Ibu dosen pengajar yang telah menemani dari awal masuk perkuliahan dan yang telah mendidik penulis memiliki banyak ilmu yang dapat diimplementasikan untuk kehidupan setelah lulus.
9. Penulis mengucapkan terima kasih yang sebesar – sebesarnyanya kepada seluruh teman – teman angkatan PE 2023 atas kebersamaan, semangat, dan dukungan moral yang tak ternilai selama menjalani masa perkuliahan hingga penyusunan Tugas Akhir ini.
10. Diri saya sendiri (penulis) karena selalu berpikir positif dan selalu berusaha mempercayai diri sendiri, hingga akhirnya diri saya mampu membuktikan bahwa saya bisa bertahan sampai di titik ini dari awal hingga akhir.

Semoga dengan selesainya penyusunan dan pengerjaan Tugas Akhir ini dapat memberikan manfaat bagi civitas akademik Program Studi Teknik Kelistrikan Kapal, Politeknik Perkapalan Negeri Surabaya. Penulis menyadari dalam penyusunan Laporan Tugas Akhir ini masih banyak terdapat kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran bagi pembaca yang bersifat membangun sehingga Laporan Tugas Akhir ini bisa menjadi lebih baik lagi untuk proses pengembangan lebih lanjut.

Surabaya, 14, July, 2026

Ahmad Valentino

Sistem Alarm Berbasis Sensor Jarak dan Sensor Berat untuk Pengukuran Volume Sampah pada Prototipe Kapal

Ahmad Valentino

ABSTRAK

Permasalahan sampah di sungai dan laut memerlukan sistem pengambilan dan pengangkutan sampah yang mampu memantau kapasitas muatan secara otomatis agar tidak terjadi kelebihan volume maupun berat. Penelitian ini bertujuan merancang, mengimplementasikan, dan menguji sistem alarm berbasis sensor jarak dan sensor berat untuk pengukuran volume sampah pada prototipe kapal pengambil sampah berbasis *web server* menggunakan ESP32. Sistem menggunakan tiga sensor ultrasonik HC-SR04 untuk mengukur volume sampah dan sensor *load cell* HX711 untuk mengukur berat sampah. Seluruh data diproses oleh mikrokontroler ESP32 kemudian ditampilkan secara real-time melalui *web server*, serta digunakan untuk mengendalikan alarm dan *conveyor* secara otomatis ketika kapasitas muatan telah mencapai batas yang ditentukan. Berdasarkan hasil pengujian, sensor ultrasonik memiliki selisih pengukuran sebesar 0–0,2 cm, sedangkan sensor *load cell* memiliki selisih pembacaan 2–4 gram dengan nilai galat sebesar 0,34%–0,45% terhadap timbangan digital. Sistem alarm berhasil aktif secara otomatis ketika tinggi sampah mencapai batas 17 cm atau berat sampah mencapai 2 kg, kemudian *conveyor* berhenti secara otomatis untuk mencegah penambahan sampah. Selain itu, *web server* berhasil menampilkan informasi volume, berat, status alarm, dan status *conveyor* secara real-time melalui browser pada laptop maupun smartphone. Hasil penelitian menunjukkan bahwa sistem mampu bekerja dengan baik sebagai sistem monitoring kapasitas muatan sampah pada prototipe kapal pengambil sampah.

Kata kunci: sensor jarak, sensor berat, volume sampah, ESP32, *web server*

(Halaman ini sengaja dikosongkan)

Alarm System Based on Distance and Weight Sensors for Measuring Waste Volume on a Boat Prototype

Ahmad Valentino

ABSTRACT

Waste problems in rivers and marine environments require an automatic waste collection system capable of monitoring load capacity to prevent excessive volume and weight. This study designed, implemented, and evaluated an alarm system based on distance and weight sensors to measure waste volume on a web server-based waste collection boat prototype using an ESP32 microcontroller. The system employs three HC-SR04 ultrasonic sensors to measure waste volume and an HX711 load cell sensor to measure waste weight. All sensor data are processed by the ESP32 and displayed in real time via a web server, while also controlling the alarm and conveyor automatically when the load capacity reaches the predefined threshold. The experimental results showed that the ultrasonic sensors achieved a measurement deviation of 0–0.2 cm, while the load cell sensor produced a deviation of 2–4 g with an error ranging from 0.34% to 0.45% compared with a digital scale. The alarm system was successfully activated when the waste height reached 17 cm, or the waste weight reached 2 kg, and the conveyor stopped automatically to prevent additional waste from entering the storage compartment. Furthermore, the web server successfully displayed the waste volume, weight, alarm status, and conveyor status in real time through a laptop or smartphone web browser. These results indicate that the proposed system can effectively monitor the waste load capacity of a waste collection boat prototype.

Keywords: *ultrasonic sensor, load cell, waste volume, ESP32, web server..*

(Halaman ini sengaja dikosongkan)

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN.....	iii
PERNYATAAN BEBAS PLAGIAT.....	v
KATA PENGANTAR.....	vii
ABSTRAK	ix
ABSTRACT	xi
DAFTAR ISI	xiii
DAFTAR TABEL.....	xvii
DAFTAR GAMBAR.....	xix
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
BAB 2 TINJAUAN PUSTAKA	5
2.1 Kajian Penelitian Terdahulu	5
2.2 Kapal Pengangkut Sampah.....	6
2.3 ESP32	7
2.4 Modul <i>Load cell</i> HX 711.....	9
2.5 Sensor Ultrasonic HC SR-04	10
2.6 Relay.....	11
2.7. Motor Servo	12

2.8	Brushless DC Motor 12V	13
2.7	Server IoT	15
2.8	Arduino IDE	15
BAB 3 METODE PENELITIAN		17
3.1	Konsep Penelitian.....	17
3.1.1	Blok Diagram.....	19
3.1.2	Flow Chart Sistem.....	21
3.2	Diagram Alir Penelitian.....	23
3.3	Perencanaan dan Desain	25
3.3.1	Perancangan Hardware	25
3.3.2	Desain hardware.....	32
3.3.3	Perancangan Software	35
3.4	Rencana Pengujian	37
BAB 4 HASIL DAN PEMBAHASAN.....		40
4.1	Pengujian Parsial.....	44
4.1.1	Pengujian ESP32	44
4.1.2	Pengujian Regulator Tegangan (Step Down)	45
4.1.3	Pengujian Sensor Ultrasonikc	47
4.1.4	Pengujian Modul <i>Load cell</i>	49
4.1.5	Pengujian Motor Penggerak <i>Conveyor</i>	53
4.1.6	Pengujian Relay	54
4.1.7	Pengujian <i>Buzzer</i> (Alarm)	55
4.2	Pengujian Terintegrasi.....	56
4.2.1	Pengujian Integrasi Sensor Ultrasonik dan Sensor Berat	56
4.2.2	Pengujian Sistem Alarm	59
4.2.3	Pengujian Sistem <i>Conveyor</i>	61

4.2.4	Pengujian <i>Web server</i> dan Kendali	64
4.3	Pembahasan	65
BAB 5 KESIMPULAN DAN SARAN.....		68
5.1	Kesimpulan	68
5.2	Saran.....	69
DAFTAR PUSTAKA.....		70
LAMPIRAN.....		72
BIODATA PENULIS.....		102

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

Tabel 2. 1 Spesifikasi ESP32	8
Tabel 3. 1 Koneksi PIN IO ESP32	30
Tabel 3. 2 Rencana data uji	36
Tabel 3. 3 Jadwal penelitian	37
Tabel 3. 4 Rencana Anggaran Penelitian Tugas Akhir	37
Tabel 4. 1 Pengujian Tegangan ESP32	43
Tabel 4. 2 Pengujian Regulator Tegangan	45
Tabel 4. 3 Pengujian Tegangan Sensor Ultrasonik	47
Tabel 4. 4 Pengujian Beban <i>Load cell</i>	50
Tabel 4. 5 Pengujian Tegangan Motor DC	52
Tabel 4. 6 Pengujian Tegangan Relay	53
Tabel 4. 7 Pengujian Buzzer	54

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2. 1 Kapal Pengangkut Sampah.....	6
Gambar 2. 2 ESP32.....	7
Gambar 2. 3 Load cell HX 711	10
Gambar 2. 4 Sensor HC SR-04.....	11
Gambar 2. 5 Relay 5V.....	12
Gambar 2. 6 Brushless Motor DC 12V.....	14
Gambar 2. 7 ESP32 Web server	15
Gambar 2. 8 Arduino IDE.	16
Gambar 3. 1 Ilustrasi Sistem yang sudah ada.....	17
Gambar 3. 2 Ilustrasi sistem yang akan dibuat	18
Gambar 3. 3 Diagram Blok	19
Gambar 3. 4 Flow Chart Sistem	22
Gambar 3. 5 Flow Chart Penelirian	24
Gambar 3. 6 Perancangan Hardware	26
Gambar 3. 7 Diagram Pengawatan Hardware	29
Gambar 3. 8 Desain Prototipe Alat.....	32
Gambar 4. 1 Hasil Perancangan Miniatur	40
Gambar 4. 2 Rangkaian Controller	41
Gambar 4. 3 Conveyor	41
Gambar 4. 4 Motor Penggerak Conveyor	42
Gambar 4. 5 Sensor Load cell	43
Gambar 4. 6 Sensor Ultrasonik	43
Gambar 4. 7 Pengujian Tegangan ESP32	44
Gambar 4. 8 Pengujian Regulator Tegangan.....	45
Gambar 4. 9 Pengujian Tegangan Sensor Ultrasonik	47
Gambar 4. 10 Pengujian Sensor Ultrasonic Terisi Sedang	47
Gambar 4. 11 Pengujian Ultrasonic Hampir Penuh.....	48
Gambar 4. 12 Pengujian Tegangan Load cell	50
Gambar 4. 13 Pengujian Beban 895 gram	50
Gambar 4. 14 Pengujian Beban 560 gram	51
Gambar 4. 15 Pengujian Beban 295 Gram.....	51
Gambar 4. 16 Pengujian Motor DC	53
Gambar 4. 17 Pengujian Tegangan Relay Saat Aktif	54
Gambar 4. 18 Pengujian Alarm	55
Gambar 4. 19 Tampilan Serial Monitor (a) dan (b) Web server Kondisi Kosong	57
Gambar 4. 20 Tampilan Serial Monitor (a) dan (b) Web server Kondisi Hampir Penuh.....	57
Gambar 4. 21 Serial Monitor Kapasitas Penuh (a) dan (b) Data Web server	58
Gambar 4. 22 Alarm Aktif pada ESP32.....	59
Gambar 4. 23 Tampilan Serial Monitor dan Web server Saat Alarm Aktif	59
Gambar 4. 24 Alarm Pada Kapasitas penuh.....	60
Gambar 4. 25 Tampilan saat alarm aktif.....	60
Gambar 4. 26 Conveyor Berjalan Normal	61
Gambar 4. 27 Tampilan Web server (a) dan (b) Serial Monitor Conveyor Normal	

.....	62
Gambar 4. 28 Conveyor Berhenti Saat Sampah Penuh	63
Gambar 4. 29 Tampilan Serial Monitor (a) dan (b) Web server Saat Sampah Penuh.....	63
Gambar 4. 30 Tampilan Web server dan Kendali	64

(Halaman ini sengaja dikosongkan)

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Permasalahan sampah di sungai dan laut hingga saat ini masih menjadi isu lingkungan yang serius. Penumpukan sampah di perairan dapat mengganggu ekosistem, menghambat aliran air, menurunkan kualitas lingkungan, serta berdampak pada aktivitas transportasi dan pariwisata perairan. Salah satu upaya yang dilakukan untuk mengatasi permasalahan tersebut adalah penggunaan kapal pengambil sampah yang berfungsi untuk mengangkut sampah dari permukaan sungai atau laut.

Dalam proses pengambilan dan pengangkutan sampah, diperlukan sistem pemantauan yang mampu memberikan informasi kondisi muatan sampah secara *real-time*. Informasi mengenai *volume* atau kapasitas sampah yang telah terkumpul sangat penting untuk mencegah terjadinya kelebihan muatan, meningkatkan efisiensi operasional, serta menjamin keselamatan dan keandalan sistem pengangkutan sampah. Namun, pada umumnya proses pemantauan volume sampah masih dilakukan secara manual sehingga kurang efektif dan berpotensi menimbulkan kesalahan.

Perkembangan teknologi mikrokontroler dan sistem berbasis *Internet of Things (IoT)* memungkinkan penerapan sistem monitoring otomatis yang dapat diakses secara jarak jauh. Penggunaan sensor jarak dan sensor berat dapat dimanfaatkan untuk memperkirakan *volume* sampah yang terkumpul pada wadah penampungan. Sensor jarak digunakan untuk mendeteksi ketinggian sampah, sedangkan sensor berat digunakan untuk mengetahui beban sampah yang terangkut. Data dari kedua sensor tersebut kemudian dapat diolah untuk menentukan kondisi kapasitas muatan.

Sebagai pusat pengendali sistem, *ESP32* dipilih karena memiliki kemampuan pemrosesan yang baik serta fitur komunikasi nirkabel yang mendukung implementasi *web server*. Dengan adanya *web server*, informasi kondisi muatan sampah dapat dipantau secara *real-time* melalui perangkat seperti komputer atau *smartphone*. Selain itu, sistem alarm diperlukan sebagai indikator

peringatan ketika *volume* sampah telah mencapai batas tertentu sehingga operator dapat segera melakukan tindakan yang diperlukan.

Berdasarkan permasalahan tersebut, diperlukan suatu perancangan sistem yang mampu melakukan pengukuran, pemantauan, dan pemberian peringatan terhadap volume sampah secara otomatis. Oleh karena itu, pada tugas akhir ini dilakukan penelitian dengan judul **“Sistem Alarm Berbasis Sensor Jarak dan Sensor Berat untuk Pengukuran Volume Sampah pada Prototipe Kapal”** yang diharapkan dapat menjadi solusi pendukung dalam meningkatkan efektivitas sistem pengambilan dan pengangkutan sampah di perairan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan di atas, maka rumusan masalah dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana merancang sistem pengukuran volume sampah berbasis sensor jarak dan sensor berat pada prototipe kapal pengambil sampah?
2. Bagaimana merancang sistem alarm sebagai indikator kapasitas muatan sampah?
3. Bagaimana merancang sistem *monitoring volume* sampah berbasis *web server* menggunakan *ESP32*.

1.3 Batasan Masalah

Batasan-batasan dalam pelaksanaan Tugas Akhir ini adalah sebagai berikut:

1. Sistem dirancang menggunakan *ESP32* sebagai pengendali utama.
2. Penelitian difokuskan pada sistem pengukuran, *monitoring*, dan alarm *volume* sampah.
3. Perancangan termasuk sistem pengendalian atau navigasi pergerakan kapal namun tidak masuk dalam pembahasan pengujian.
4. Pengukuran *volume* sampah dilakukan menggunakan kombinasi data sensor jarak dan sensor berat.

5. Sistem monitoring ditampilkan melalui *web server*.

1.4 Tujuan Penelitian

Tujuan dari penulisan Tugas Akhir ini adalah sebagai berikut:

1. Merancang sistem pengukuran *volume* sampah pada prototipe kapal pengambil sampah.
2. Merancang sistem alarm untuk memberikan peringatan ketika *volume* sampah mencapai batas tertentu.
3. Merancang sistem *monitoring volume* sampah berbasis *web server* menggunakan *ESP32*.

1.5 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian Tugas Akhir ini adalah sebagai berikut:

1. **Bagi Penulis** Sebagai syarat untuk menyelesaikan Tugas Akhir pada program studi yang ditempuh serta sebagai sarana penerapan ilmu yang telah dipelajari.
2. **Bagi Institusi Pendidikan** Penelitian ini dapat dijadikan sebagai referensi atau bahan pembelajaran terkait penerapan sistem monitoring berbasis sensor dan *web server*.
3. **Bagi Pengembangan Teknologi Lingkungan** Memberikan gambaran penerapan sistem monitoring dan alarm otomatis pada proses pengambilan dan pengangkutan sampah di sungai atau laut guna meningkatkan efektivitas dan keselamatan operasional.

(Halaman ini sengaja dikosongkan)

BAB 2

TINJAUAN PUSTAKA

2.1 Kajian Penelitian Terdahulu

Bagian ini menjelaskan tentang penelitian sebelumnya dan referensi-referensi yang berhubungan dengan penelitian yang akan dilakukan. Di bawah ini adalah penjabaran dari penelitian-penelitian yang telah dilakukan sebelumnya :

- 1) Penelitian oleh Gunawan dkk. mengembangkan sistem monitoring sampah berbasis IoT yang memadukan sensor ultrasonik dan sensor beban (*load cell*) untuk memantau ketinggian dan berat sampah secara real-time. Sistem yang dirancang mampu menampilkan indikator penuh pada web ketika tinggi atau berat sampah mencapai batas maksimum, serta hasil pengujian menunjukkan bahwa pengukuran tinggi sampah memiliki kesalahan rata-rata hanya 0,08 cm dan pengukuran berat memiliki selisih rata-rata 0,03 kg dibandingkan timbangan konvensional, sehingga sistem ini memberikan akurasi yang sangat dekat dengan kondisi nyata. Hasil tersebut menunjukkan bahwa penerapan sensor ganda ini efektif untuk monitoring volume dan berat sampah dalam konteks IoT. (Gunawan, Hernawati & Aditya, 2021).
- 2) Penelitian oleh Salamah dkk. merancang sistem monitoring berat dan volume sampah pada alat pemilah sampah berbasis IoT menggunakan aplikasi Blynk. Sistem ini memanfaatkan sensor *load cell* dan sensor ultrasonik yang dipasang pada kotak sampah organik dan anorganik serta terhubung melalui NodeMCU ESP8266. Hasil penelitian menunjukkan bahwa sistem mampu mengukur berat dan volume sampah secara real-time dan mengirim notifikasi saat tempat sampah sudah penuh dari jarak uji hingga 1 km, sehingga mempermudah petugas dalam mengetahui kondisi sampah tanpa perlu pengecekan langsung. (Salamah, Kusumanto & Evelina, 2023)
- 3) Penelitian oleh Yama mengembangkan tempat sampah cerdas berbasis NodeMCU ESP8266 yang mampu mendeteksi volume sampah menggunakan sensor ultrasonik dan mengirim notifikasi otomatis melalui aplikasi Telegram ketika tempat sampah penuh. Hasil penelitian menunjukkan bahwa sensor ultrasonik HC-SR04 efektif mendeteksi ketinggian sampah hingga 5 cm dan

sistem mampu mengirimkan notifikasi otomatis, sehingga dapat meningkatkan efisiensi pengelolaan sampah serta mempermudah petugas dalam menjaga kondisi tempat sampah secara efektif. (Yama, 2023)

2.2 Kapal Pengangkut Sampah

Kapal pengangkut sampah adalah kapal kerja yang dirancang secara khusus untuk melakukan aktivitas pengambilan serta pemindahan sampah yang berada di permukaan perairan, baik di sungai, kanal, maupun wilayah laut pesisir. Kapal ini umumnya dilengkapi dengan mekanisme pengangkut mekanis, seperti *conveyor belt*, yang berfungsi memindahkan sampah dari permukaan air ke dalam bak penampungan di atas kapal. Penerapan kapal pengangkut sampah bertujuan untuk mengurangi tingkat pencemaran perairan, menjaga kebersihan lingkungan air, serta meminimalkan dampak negatif sampah terhadap ekosistem dan aktivitas manusia di sekitar perairan tersebut (Mu'afifah et al., 2023).



Gambar 2. 1 Kapal Pengangkut Sampah.

Sumber : <https://mavideniz.com.tr/news-events/trash-skimmer-boat/>

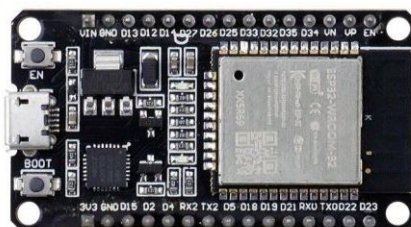
Perancangan kapal pengangkut sampah perlu mempertimbangkan berbagai aspek teknis, antara lain stabilitas kapal, kemampuan bermanuver di perairan dangkal, serta efisiensi proses pengumpulan sampah. Desain lambung tipe katamaran banyak digunakan karena mampu memberikan kestabilan yang lebih baik ketika kapal membawa muatan sampah dengan distribusi beban yang

tidak seragam. Selain itu, penggunaan sistem *conveyor* memungkinkan proses pengangkutan sampah dilakukan secara berkesinambungan dengan intervensi operator yang minimal, sehingga meningkatkan efektivitas dan keselamatan selama operasi pembersihan perairan berlangsung (Mu'afifah et al., 2023).

Keberadaan kapal pengangkut sampah memiliki peran strategis dalam mendukung sistem pengelolaan lingkungan perairan yang berkelanjutan. Kapal ini memungkinkan kegiatan pembersihan perairan dilakukan secara rutin dan terkontrol, khususnya pada wilayah yang berpotensi mengalami penumpukan sampah dalam jumlah besar. Dengan perancangan yang tepat serta dukungan sistem pemantauan muatan yang memadai, kapal pengangkut sampah dapat meningkatkan efisiensi operasional sekaligus berkontribusi dalam upaya pelestarian lingkungan perairan secara menyeluruh (Mu'afifah et al., 2023).

2.3 ESP32

ESP32 merupakan sebuah *system-on-chip* (SoC) yang mengintegrasikan mikroprosesor *dual-core berbasis Xtensa 32-bit LX6* dengan modul komunikasi *Wi-Fi 2,4 GHz* dan *Bluetooth versi 4.2* dalam satu *chip*. *ESP32* dirancang menggunakan teknologi fabrikasi 40 nm yang memungkinkan konsumsi daya rendah namun tetap memiliki performa tinggi. *Chip* ini mendukung mode *station*, *soft access point*, dan *promiscuous mode* secara bersamaan, serta memiliki kecepatan transmisi data hingga 150 Mbps pada standar 802.11 b/g/n. Selain itu, *ESP32* juga dilengkapi pemrosesan kriptografi perangkat keras seperti *AES*, *SHA*, dan *RSA* yang memperkuat keamanan data pada aplikasi berbasis *Internet of Things (IoT)* (Espressif Systems, 2022).



Gambar 2. 2 ESP32.

Sumber : <https://www.cytron.io/p-nodemcu-esp32?srsId=AfmBOOp0Jeeef6n2fKkKozVrOJTj8RfCDA4HclbazWRYDXFLt6wHe00>

Dari sisi perangkat keras, *ESP 32* memiliki memori internal berupa 520 KB *SRAM* dan 448 KB *ROM*, serta mendukung memori eksternal berupa *flash* dan *PSRAM* melalui antarmuka *QSPI*. *ESP32* bekerja pada tegangan operasi 2,3–3,6 V dengan rekomendasi tegangan 3,3 V. Konsumsi daya pada mode aktif berada pada kisaran 20–68 mA tergantung frekuensi kerja dan konfigurasi inti prosesor, sedangkan pada mode *deep-sleep* dapat mencapai serendah 10 μ A. Selain itu, *ESP 32* juga dilengkapi berbagai periferal seperti *ADC 12-bit*, *DAC*, *PWM*, *UART*, *SPI*, *I2C*, *I2S*, serta *touch sensor* yang membuatnya sangat fleksibel untuk berbagai aplikasi sistem tertanam (Espressif Systems, 2022). Spesifikasi secara lebih detil dari *ESP32* dapat dilihat pada table di bawah ini :

Tabel 2. 1 Spesifikasi *ESP32*

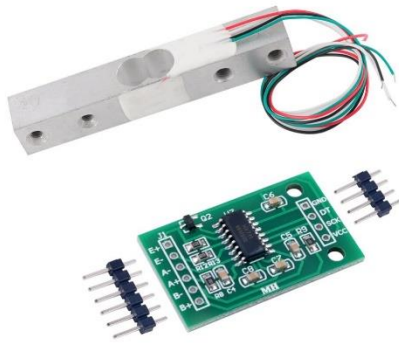
No	Parameter	Spesifikasi
1	Arsitektur <i>CPU</i>	<i>Xtensa 32-bit LX6 (single atau dual-core)</i>
2	Frekuensi CPU Maksimum	Hingga 240 MHz
3	Memori <i>ROM</i>	448 KB
4	Memori <i>SRAM</i>	520 KB
5	Memori Eksternal	<i>Flash dan PSRAM via QSPI</i>
6	Tegangan Operasi	2,3 – 3,6 V
7	Tegangan Rekomendasi	3,3 V
8	<i>Wi-Fi</i>	802.11 b/g/n, 2,4 GHz, hingga 150 Mbps
9	<i>Bluetooth</i>	<i>Bluetooth 4.2 BR/EDR dan BLE</i>
10	Mode Daya	Active, Modem-sleep, Light-sleep, Deep-sleep,

No	Parameter	Spesifikasi
		Hibernation
11	Konsumsi Arus Mode Aktif	20 – 68 mA
12	Konsumsi Arus <i>Deep-sleep</i>	$\pm 10 \mu\text{A}$
13	ADC	2 buah ADC 12-bit, 18 channel
14	DAC	2 channel
15	Antarmuka Digital	<i>GPIO, UART, SPI, I2C, I2S, PWM, Ethernet MAC</i>
16	Fitur Keamanan	<i>AES, SHA, RSA, RNG</i>
17	Suhu	$-40\text{ }^{\circ}\text{C}$ hingga $125\text{ }^{\circ}\text{C}$

Sumber : Espressif Systems, 2022

2.4 Modul *Load cell* HX 711

Load cell merupakan perangkat sensor yang berfungsi untuk mendeteksi besarnya gaya atau beban dengan memanfaatkan perubahan karakteristik listrik pada *strain gauge* akibat adanya deformasi mekanik. Saat beban diberikan, elemen logam pada *load cell* mengalami perubahan bentuk dalam skala sangat kecil yang menyebabkan nilai resistansi *strain gauge* ikut berubah. Perubahan resistansi tersebut menghasilkan sinyal tegangan yang proporsional terhadap beban yang diterima, namun dengan amplitudo yang sangat rendah sehingga memerlukan pengolahan lanjutan sebelum dapat dibaca oleh sistem digital (Avia Semiconductor, 2018).



Gambar 2. 3 *Load cell* HX 711. Sumber : <https://bt-electronic.com/product/arduino-load-cell-hx711-10kg/>

HX711 adalah modul penguat sinyal dan konverter analog ke digital yang dirancang khusus untuk aplikasi pengukuran berat. Modul ini memiliki resolusi 24-bit dan dilengkapi penguat internal dengan tingkat noise rendah, sehingga mampu memperkuat sinyal kecil dari *load cell* secara akurat. Setelah proses penguatan dan konversi, data digital dikirimkan ke mikrokontroler melalui antarmuka komunikasi dua jalur, yang memudahkan integrasi dengan sistem kendali seperti ESP32 dalam aplikasi penimbangan dan monitoring beban secara real-time (Avia Semiconductor, 2018).

2.5 Sensor Ultrasonic HC SR-04

Sensor ultrasonik HC-SR04 adalah sensor jarak yang bekerja dengan memanfaatkan gelombang ultrasonik untuk mendeteksi posisi suatu objek. Sensor ini menghasilkan gelombang ultrasonik melalui pin pemicu (trigger), kemudian menerima kembali gelombang pantulan melalui pin penerima (echo) setelah mengenai objek di depannya. Selang waktu antara sinyal pemancar dan sinyal pantulan digunakan sebagai dasar perhitungan jarak dengan mengacu pada kecepatan rambat suara di udara (Electrofreaks, 2019).



Gambar 2. 4 Sensor HC SR-04.

Sumber : <https://www.nn-digital.com/blog/2019/07/31/cara-kerja-sensor-hc-sr04-dan-contoh-program-dengan-arduino/>

HC-SR04 mampu melakukan pengukuran jarak dalam rentang beberapa sentimeter hingga beberapa meter dengan tingkat ketelitian yang memadai untuk aplikasi monitoring. Sensor ini banyak digunakan dalam sistem pengukuran ketinggian, otomasi, dan pemantauan volume karena pengoperasiannya sederhana dan mudah diintegrasikan dengan mikrokontroler. Dalam aplikasi pengukuran volume, data jarak yang diperoleh dari sensor ultrasonik dapat digunakan untuk memperkirakan tinggi material atau objek, seperti sampah di dalam wadah penampungan, sehingga tingkat kepenuhan dapat diketahui secara efektif (*Electrofreaks*, 2019).

2.6 Relay

Relay merupakan komponen elektromekanis yang berfungsi sebagai sakelar otomatis untuk menghubungkan dan memutuskan arus listrik berdaya besar menggunakan sinyal listrik berdaya kecil. Relay bekerja berdasarkan prinsip elektromagnetik, yaitu ketika arus mengalir melalui kumparan (*coil*), maka akan terbentuk medan magnet yang menarik armatur sehingga mengubah posisi kontak dari kondisi *normally open* (NO) ke *normally closed* (NC) atau sebaliknya. Mekanisme ini memungkinkan pengendalian beban bertegangan tinggi secara aman menggunakan sistem kontrol bertegangan rendah (*Boylestad & Nashelsky*, 2013).



Gambar 2. 5 Relay 5V

Sumber : <https://digiwarestore.com/en/io-module/relay-module-1-channel-5v-with-led-indicator-263068.html>

Relay banyak diaplikasikan pada sistem berbasis mikrokontroler karena mampu memberikan isolasi listrik antara rangkaian kendali dan rangkaian daya. Isolasi ini berfungsi untuk melindungi rangkaian pengendali dari lonjakan arus dan tegangan tinggi pada sisi beban. Dalam sistem otomasi dan *Internet of Things (IoT)*, relay digunakan sebagai aktuator untuk mengendalikan perangkat seperti lampu, motor listrik, pompa, dan peralatan lainnya secara otomatis maupun jarak jauh (Boylestad & Nashelsky, 2013).

2.7. Motor Servo

Motor servo merupakan aktuator elektromekanis yang dirancang untuk mengendalikan posisi sudut, kecepatan, atau torsi secara presisi melalui sistem kendali tertutup (*closed-loop control*). Motor servo terdiri dari beberapa komponen utama, yaitu motor listrik (umumnya motor DC), rangkaian pengendali, sistem roda gigi reduksi, dan sensor posisi berupa potensiometer atau encoder. Sensor posisi berfungsi memberikan umpan balik terhadap posisi poros motor, sehingga sistem dapat membandingkan posisi aktual dengan posisi yang diinginkan. Apabila terjadi perbedaan, rangkaian pengendali akan menyesuaikan sinyal kendali hingga posisi poros sesuai dengan perintah yang diberikan (Bolton, 2015).

Prinsip kerja motor servo didasarkan pada sinyal kendali berupa pulsa modulasi lebar pulsa (*Pulse Width Modulation / PWM*). Lebar pulsa menentukan sudut putaran poros servo, di mana pulsa dengan lebar tertentu akan merepresentasikan posisi sudut tertentu. Rangkaian pengendali internal servo akan memproses sinyal PWM tersebut dan menggerakkan motor hingga sensor posisi mendeteksi bahwa sudut poros telah sesuai dengan sudut yang diperintahkan. Dengan adanya sistem umpan balik ini, motor servo mampu mempertahankan posisi secara stabil meskipun terjadi perubahan beban eksternal, sehingga sangat cocok digunakan pada sistem yang membutuhkan ketelitian dan kestabilan gerak (Bolton, 2015).

Motor servo banyak digunakan dalam berbagai aplikasi sistem kendali otomatis karena kemampuannya dalam mengatur posisi secara akurat dan responsif. Dalam aplikasi pengendalian mekanik, seperti sistem kemudi, lengan robot, dan aktuator presisi lainnya, motor servo berperan penting dalam mengubah sinyal listrik menjadi gerakan sudut yang terkontrol. Keunggulan motor servo terletak pada kemudahan pengendalian, presisi tinggi, serta kemampuannya untuk bekerja dalam sistem kendali tertutup, sehingga performa sistem secara keseluruhan menjadi lebih andal dan efisien (Bolton, 2015).

2.8 Brushless DC Motor 12V

Brushless DC Motor (BLDC) adalah motor listrik yang menggunakan magnet permanen pada bagian rotor dan kumparan pada stator, serta tidak menggunakan sikat (*brush*) dan komutator mekanik dalam proses kerjanya. Proses komutasi pada motor BLDC dilakukan secara elektronik dengan mengatur aliran arus ke kumparan stator secara bergantian sehingga membentuk medan magnet berputar. Medan magnet ini akan berinteraksi dengan magnet permanen pada rotor dan menghasilkan momen putar yang menyebabkan rotor berputar. Tidak digunakannya sikat pada motor BLDC mengurangi gesekan mekanis dan percikan listrik, sehingga meningkatkan efisiensi, keandalan, dan umur pakai motor dibandingkan motor DC konvensional (Şen & Mutluer, 2025).



Gambar 2. 6 Brushless Motor DC 12V

Sumber : Motor DC Brushless Mikro 12V 40 untuk Pompa Urea Otomotif - Cina Motor Micro DC, Motor Roda Gigi DC

Prinsip kerja motor BLDC sangat bergantung pada sistem komutasi elektronik yang mengatur urutan pemberian arus pada kumparan stator sesuai dengan posisi rotor. Informasi posisi rotor dapat diperoleh menggunakan sensor posisi, seperti sensor Hall Effect, atau melalui metode tanpa sensor (*sensorless*) dengan memanfaatkan tegangan balik (*back electromotive force*). Pengaturan arus dan urutan komutasi ini memungkinkan medan magnet stator berputar secara sinkron dengan rotor, sehingga menghasilkan putaran yang halus dan torsi yang relatif stabil. Karakteristik ini menjadikan motor BLDC mampu beroperasi pada kecepatan tinggi dengan tingkat efisiensi yang baik serta respon dinamis yang cepat (Şen & Mutluer, 2025).

Motor BLDC banyak digunakan sebagai penggerak pada berbagai aplikasi karena memiliki rasio daya terhadap bobot yang tinggi, efisiensi energi yang baik, dan kebutuhan perawatan yang rendah. Dalam sistem penggerak, kecepatan motor BLDC berbanding lurus dengan tegangan yang diberikan, sedangkan torsi berbanding lurus dengan arus yang mengalir pada kumparan stator. Ketika beban meningkat, arus motor akan bertambah untuk mempertahankan kecepatan putaran. Karakteristik tersebut menjadikan motor BLDC sangat sesuai digunakan sebagai penggerak pada sistem propulsi listrik, termasuk pada aplikasi kendaraan dan kapal listrik, yang membutuhkan performa tinggi dengan konsumsi energi yang efisien (Şen & Mutluer, 2025).

2.7 Server IoT

Web server dalam sistem *Internet of Things (IoT)* berfungsi sebagai pusat layanan yang memungkinkan perangkat tertanam diakses melalui jaringan menggunakan protokol *HTTP*. Berdasarkan jurnal *IoT-Based Smart Room Using Web server-Based ESP32 Microcontroller*, *web server* digunakan sebagai media antarmuka antara pengguna dan perangkat keras untuk melakukan *monitoring* serta pengendalian peralatan listrik secara jarak jauh. Sistem ini memungkinkan pengguna mengakses data dan mengontrol perangkat hanya melalui *browser* tanpa perlu aplikasi tambahan, sehingga meningkatkan fleksibilitas dan kemudahan penggunaan (Akbar et al., 2022).



Gambar 2. 7 ESP32 Web server

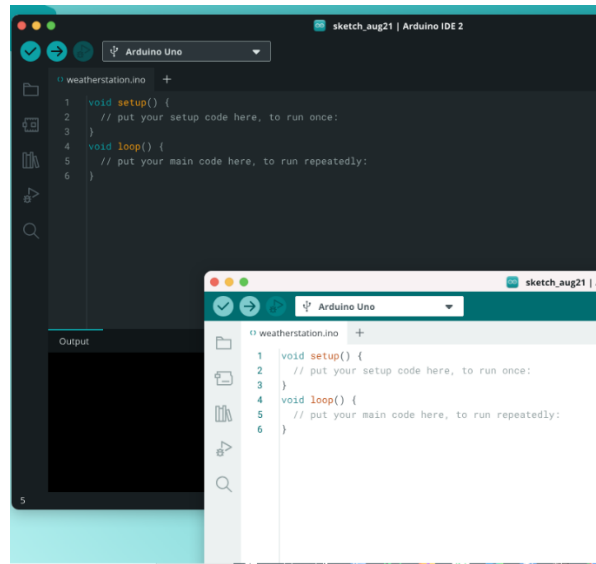
Sumber : <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>

Penerapan *web server* langsung pada mikrokontroler memungkinkan proses pengiriman dan penerimaan data berlangsung secara *real-time* melalui jaringan *Wi-Fi*. *Web server* berperan sebagai *server* lokal yang mengelola permintaan (*request*) dan respons (*response*) dari pengguna, baik dalam bentuk tampilan kondisi perangkat maupun perintah kendali. Dengan konsep ini, sistem *IoT* dapat bekerja secara mandiri tanpa ketergantungan pada *server* eksternal, sehingga lebih efisien dari segi biaya dan lebih stabil untuk aplikasi monitoring ruangan pintar (Akbar et al., 2022).

2.8 Arduino IDE

Arduino IDE atau *Arduino Integrated Development Environment* adalah perangkat lunak lintas *platform* yang dirancang untuk membantu pengguna dalam menulis, menyusun (*compile*), dan memuat (*upload*) program ke papan

mikrokontroler seperti *Arduino* dan papan kompatibel lainnya. Perangkat lunak ini tersedia secara bebas dan bersifat *open-source*, serta dapat diunduh langsung melalui situs resmi *Arduino*, mendukung berbagai sistem operasi seperti *Windows*, *macOS*, dan *Linux* tanpa biaya lisensi tambahan (*Arduino.cc*, 2025).



Gambar 2. 8 Arduino IDE. Sumber : <https://www.arduino.cc/en/software/>

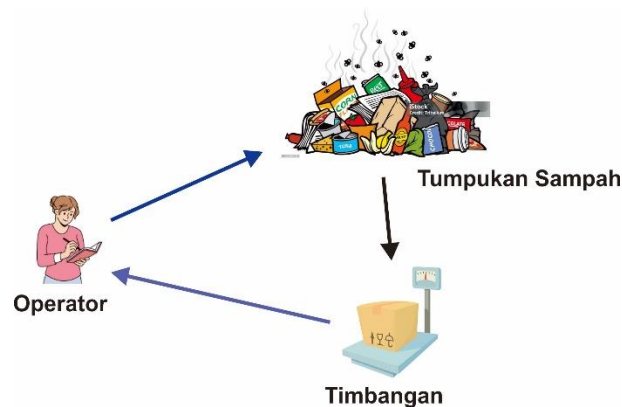
Dalam praktiknya, Arduino IDE menyediakan sebuah lingkungan yang komprehensif termasuk editor teks untuk menulis kode program (*sketches*), konsol pesan untuk menampilkan hasil penyusunan kode, serta alat untuk memeriksa kesalahan dan mengunggah program ke papan mikrokontroler. Lingkungan ini mengintegrasikan perpustakaan kode (*libraries*) serta alat bantu lain yang memungkinkan pengguna mengakses fungsi perangkat keras dengan lebih mudah, sehingga cocok untuk pemula maupun pengembang yang ingin membangun aplikasi elektronik dan IoT (*Arduino.cc*, 2025).

Pemasangan dan penggunaan Arduino IDE relatif sederhana; setelah perangkat lunak diunduh dari situs resmi, pengguna dapat langsung memilih jenis papan dan port komunikasi yang sesuai, lalu menulis dan mengunggah program melalui kabel USB atau koneksi lain yang didukung. Kemampuan *IDE* ini dalam menyediakan antarmuka yang intuitif dan dukungan beragam pustaka menjadi salah satu faktor utama mengapa platform *Arduino* banyak dipilih untuk prototipe, pendidikan, dan aplikasi *IoT* seperti pengendalian sensor dan aktuator pada sistem monitoring berbasis ESP32 (*Arduino.cc*, 2025).

BAB 3 METODE PENELITIAN

3.1 Konsep Penelitian

Konsep penelitian ini berfokus pada pengembangan sistem pengukuran volume dan berat sampah pada prototipe kapal pengambil sampah yang sebelumnya masih dilakukan secara manual. Sistem yang telah ada menjadi dasar untuk dilakukan perancangan dan pengembangan sistem baru yang lebih efektif, efisien, serta terintegrasi dengan teknologi *monitoring* berbasis *web server*. Pada sistem yang sudah ada sebelumnya, sebagaimana ditunjukkan pada Gambar 3.1, proses pengukuran volume dan berat sampah dilakukan secara manual oleh operator. Sampah yang telah dikumpulkan pada kapal pengangkut ditimbang menggunakan timbangan konvensional, sementara volume sampah diperkirakan secara visual berdasarkan tingkat kepenuhan bak penampung. Data hasil pengukuran tersebut kemudian dicatat secara manual oleh operator tanpa adanya sistem pemantauan jarak jauh. Kondisi ini memiliki beberapa keterbatasan, antara lain ketergantungan pada ketelitian operator, potensi kesalahan pencatatan, serta tidak tersedianya data *real-time* yang dapat diakses secara daring.



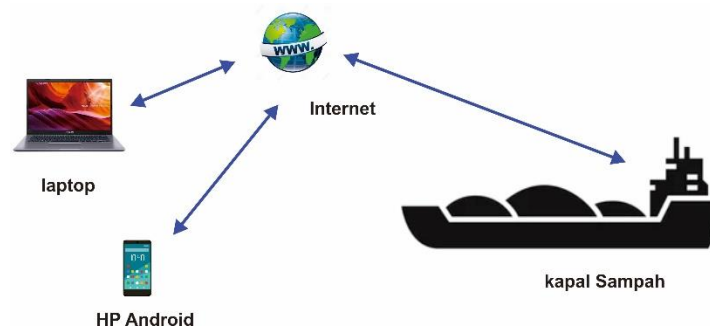
Gambar 3. 1 Ilustrasi Sistem yang sudah ada

Dengan berkembangnya teknologi *Internet of Things (IoT)*, sistem manual tersebut dikembangkan menjadi sistem otomatis dan terintegrasi secara daring. Konsep sistem yang akan dikembangkan pada penelitian ini ditunjukkan pada

Gambar 3.2. Pada sistem ini, pengukuran volume sampah dilakukan menggunakan sensor jarak ultrasonik yang dipasang pada bagian atas bak penampung sampah. Sensor ini berfungsi untuk mendeteksi jarak antara sensor dan permukaan sampah, sehingga tingkat kepenuhan bak dapat dihitung secara otomatis. Sementara itu, pengukuran berat sampah dilakukan menggunakan sensor berat (*load cell*) yang terpasang pada dasar penampung sampah.

Data hasil pembacaan sensor jarak dan sensor berat akan diproses oleh mikrokontroler sebagai pusat pengendali sistem. Informasi volume dan berat sampah selanjutnya dikirimkan ke *web server* melalui koneksi internet, sehingga dapat dipantau secara *real-time* menggunakan perangkat seperti laptop maupun *smartphone*. Dengan adanya sistem ini, operator tidak perlu lagi melakukan pengukuran dan pencatatan secara manual.

Selain sistem *monitoring* daring, penelitian ini juga mengembangkan sistem keluaran berupa alarm dan *conveyor* sampah. Alarm akan aktif secara otomatis ketika volume atau berat sampah telah mencapai batas maksimum yang telah ditentukan. Sementara itu, *conveyor* sampah berfungsi untuk membantu proses pemindahan dan penataan sampah di dalam kapal agar distribusi beban lebih merata dan kapasitas penampungan dapat dimanfaatkan secara optimal.



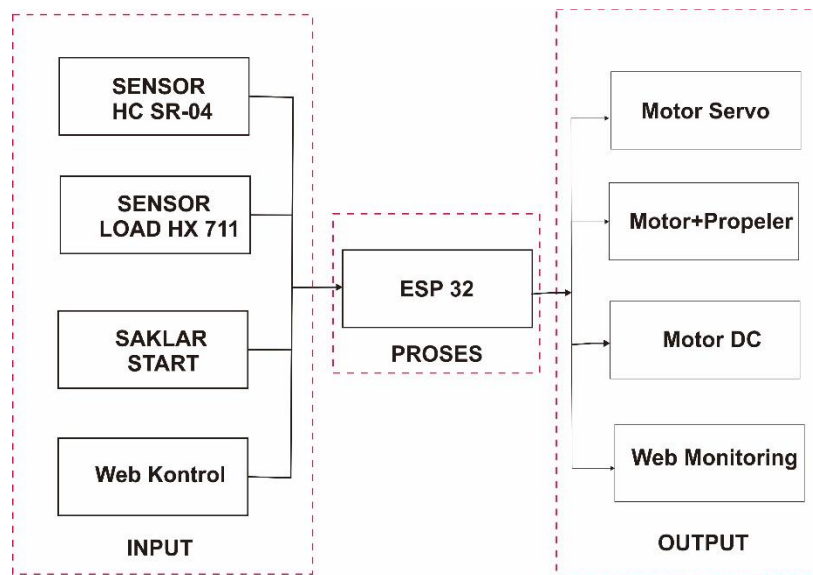
Gambar 3. 2 Ilustrasi sistem yang akan dibuat

Dengan diterapkannya sistem alarm berbasis sensor jarak dan sensor berat yang terintegrasi dengan *web server*, diharapkan proses pengelolaan sampah pada kapal pengambil sampah menjadi lebih efektif, akurat, dan efisien. Sistem ini juga memberikan kemudahan dalam pemantauan kondisi kapal secara daring, sehingga

dapat mendukung pengambilan keputusan yang lebih cepat dan tepat oleh operator maupun pihak pengelola.

3.1.1 Blok Diagram

Untuk mempermudah pemahaman terhadap alur kerja dan hubungan antar komponen pada sistem yang dirancang, maka disusun diagram blok sistem. Diagram blok ini menggambarkan proses kerja sistem secara keseluruhan mulai dari bagian input, proses, hingga output. Setiap blok menunjukkan fungsi utama dari masing-masing komponen yang saling terintegrasi dalam sistem pengukuran volume dan berat sampah pada prototipe kapal pengambil sampah. Diagram blok sistem yang dirancang ditunjukkan pada Gambar 3.3.



Gambar 3. 3 Diagram Blok

Berdasarkan Gambar 3.3, bagian input sistem terdiri dari sensor ultrasonik HC-SR04, sensor berat *load cell* dengan modul HX711, saklar (*switch*) start, serta *web kontrol*. Sensor ultrasonik HC-SR04 berfungsi untuk mendeteksi ketinggian sampah yang telah menumpuk di dalam bak penampungan kapal. Sensor ini bekerja dengan memancarkan gelombang ultrasonik dan menerima pantulan gelombang dari permukaan sampah, sehingga jarak antara sensor dan permukaan sampah dapat

diketahui. Dengan penggunaan lebih dari satu sensor ultrasonik, sistem mampu memperkirakan volume sampah di dalam penampungan secara lebih akurat berdasarkan tingkat kepenuhan bak. Selain pengukuran volume, sistem juga dilengkapi dengan sensor berat berupa *load cell* yang terhubung dengan modul HX711. Sensor ini berfungsi untuk mengukur berat total sampah yang berada di dalam penampungan. Data berat yang diperoleh dari *load cell* akan dikonversi menjadi data digital oleh modul HX711 sebelum diproses oleh mikrokontroler. Kombinasi data dari sensor ultrasonik dan sensor berat memungkinkan sistem untuk mengetahui kondisi penampungan sampah secara menyeluruh, baik dari sisi volume maupun beban yang diterima kapal. Alasan penggunaan kombinasi sensor ultrasonik dan sensor berat adalah karena volume sampah tidak selalu berbanding lurus dengan beratnya. Sampah berukuran besar dapat memiliki massa ringan, sedangkan sampah berukuran kecil dapat memiliki massa yang besar. Oleh karena itu kedua sensor digunakan secara bersamaan agar sistem mampu mengevaluasi kapasitas ruang penampungan dan beban maksimum kapal secara lebih akurat.

Seluruh data dari sensor-sensor input tersebut diproses oleh mikrokontroler ESP32 yang berperan sebagai pusat pengendali sistem. ESP32 menerima data jarak, volume estimasi, dan berat sampah, kemudian membandingkannya dengan nilai batas (*threshold*) yang telah ditentukan. Sistem dapat dijalankan melalui dua cara, yaitu secara manual menggunakan saklar (*switch*) yang terpasang pada miniatur kapal, serta secara jarak jauh melalui *web kontrol* yang dapat diakses menggunakan laptop atau *smartphone*. Dengan adanya *web kontrol*, operator dapat mengaktifkan atau menonaktifkan sistem tanpa harus berada langsung di lokasi kapal.

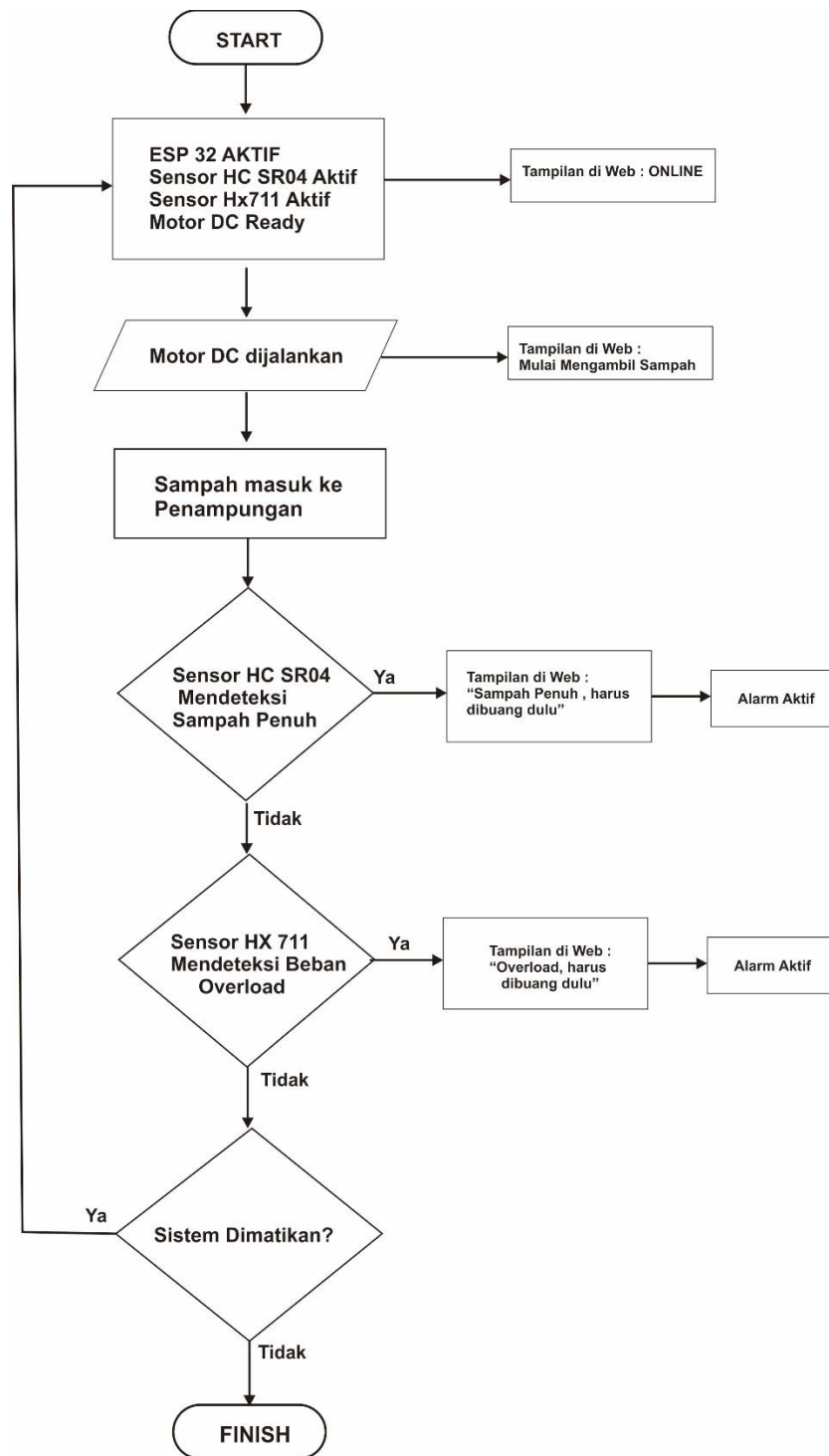
Bagian output sistem terdiri dari motor DC, motor DC BLDC, Servo, alarm, dan *web server*. Motor DC digunakan untuk menggerakkan *conveyor* yang berfungsi memasukkan dan mengatur posisi sampah di dalam penampungan agar distribusi beban lebih merata. Alarm akan aktif secara otomatis ketika volume sampah telah melebihi batas maksimum atau ketika berat sampah mencapai batas aman yang telah ditentukan. Selain itu, seluruh parameter sensor seperti ketinggian sampah, estimasi volume, dan berat sampah dapat dipantau secara *real-time* melalui *web server*. Dengan demikian, sistem ini tidak hanya meningkatkan efisiensi pengelolaan sampah, tetapi juga memberikan kemudahan dalam pemantauan dan

pengendalian kapal pengambil sampah secara daring. Servo berperan sebagai sistem kemudi yang dikendalikan dari web, dan Motor DC BLDC sebagai penggerak *propeller*.

3.1.2 Flow Chart Sistem

Flow chart sistem digunakan untuk menggambarkan alur kerja sistem secara keseluruhan mulai dari proses inisialisasi, pembacaan sensor, pengambilan keputusan, hingga keluaran sistem. *Flowchart* ini memberikan gambaran yang jelas mengenai urutan proses serta hubungan antar komponen yang terlibat dalam sistem alarm berbasis sensor jarak dan sensor berat pada prototipe kapal pengambil sampah. Dengan adanya *Flowchart* ini, proses kerja sistem dapat dipahami secara sistematis sehingga memudahkan dalam proses perancangan, implementasi, maupun analisis kinerja sistem. *Flowchart* sistem yang dirancang pada penelitian ini ditunjukkan pada Gambar 3.4.

Flowchart sistem juga menunjukkan kondisi percabangan yang digunakan sebagai dasar pengambilan keputusan berdasarkan nilai ambang (*threshold*) dari sensor jarak dan sensor berat. Setiap hasil pembacaan sensor dibandingkan dengan batas yang telah ditetapkan untuk menentukan status kapasitas penampungan sampah. Apabila kondisi normal terpenuhi, sistem akan kembali melakukan pembacaan sensor secara berulang. Sebaliknya, jika salah satu atau kedua sensor menunjukkan nilai melebihi batas, maka sistem akan mengaktifkan alarm sebagai indikator peringatan. Keluaran sistem berupa alarm visual maupun audio bertujuan untuk memberi informasi kepada operator kapal secara real-time. Dengan demikian, *Flowchart* ini berfungsi sebagai acuan utama dalam memastikan sistem alarm bekerja sesuai dengan perancangan yang telah ditetapkan.



Gambar 3. 4 Flow Chart Sistem

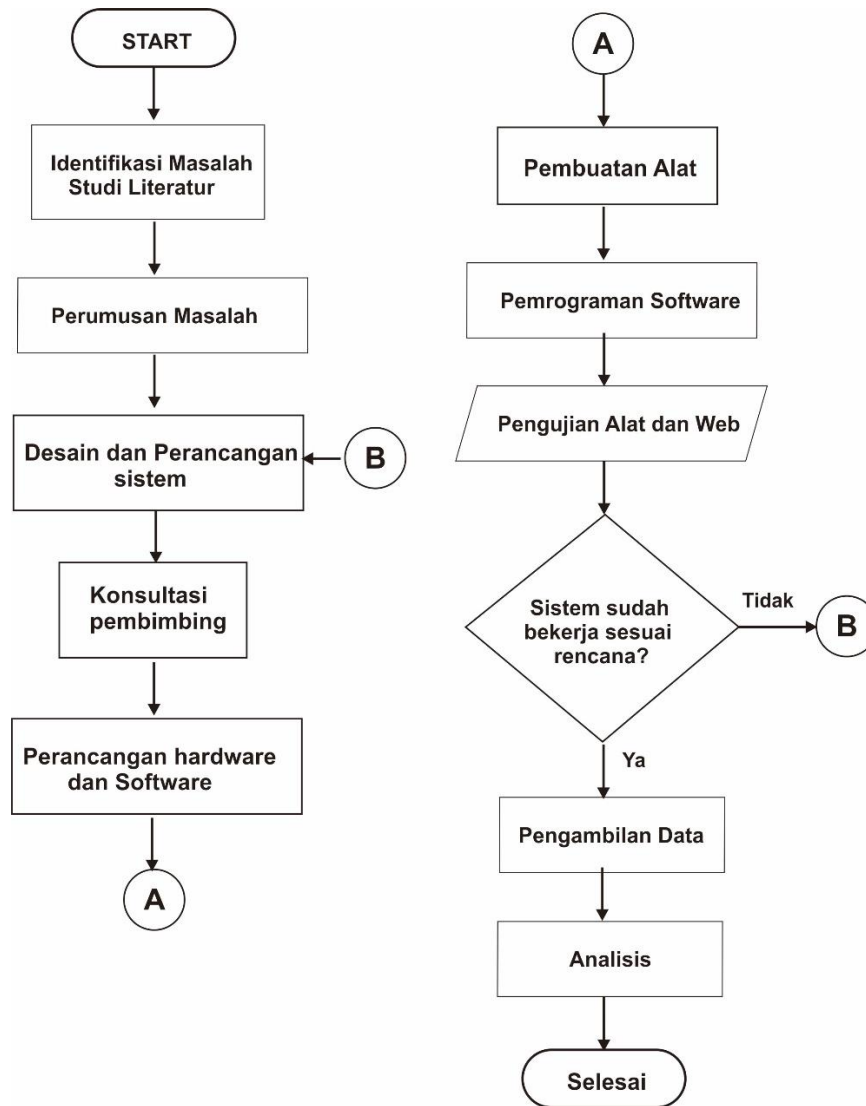
Berdasarkan *Flowchart* pada Gambar 3.4, proses sistem diawali dengan kondisi *start* yang menandakan sistem mulai diaktifkan. Pada tahap ini, mikrokontroler ESP32 melakukan inisialisasi seluruh perangkat keras yang terhubung, meliputi sensor ultrasonik HC-SR04 sebagai pendeteksi ketinggian sampah, sensor berat *load cell* dengan modul HX711 sebagai pengukur beban sampah, serta motor DC yang berfungsi menggerakkan *conveyor*. Setelah seluruh komponen berada dalam kondisi siap, sistem akan mengirimkan status *online* ke *web server* sebagai indikator bahwa sistem telah aktif dan dapat dipantau secara daring. Selanjutnya, motor DC dijalankan untuk mulai mengoperasikan *conveyor* pengambil sampah, dan informasi proses pengambilan sampah ditampilkan pada antarmuka web agar operator mengetahui bahwa sistem sedang bekerja.

Sampah yang masuk ke dalam penampungan kemudian dipantau secara terus-menerus oleh sensor ultrasonik dan sensor berat. Sensor HC-SR04 akan mendeteksi ketinggian sampah di dalam bak penampungan untuk menentukan apakah volume sampah telah mencapai batas maksimum yang telah ditetapkan. Apabila sensor ultrasonik mendeteksi kondisi penampungan penuh, sistem akan menampilkan peringatan pada *web server* berupa informasi bahwa sampah harus segera dibuang, sekaligus mengaktifkan alarm sebagai tanda peringatan. Jika kondisi penampungan belum penuh, sistem akan melanjutkan pemantauan melalui sensor HX711 untuk mendeteksi apakah beban sampah telah melebihi batas aman. Apabila terdeteksi kondisi *overload*, sistem akan kembali menampilkan notifikasi peringatan pada *web server* dan mengaktifkan alarm. Namun, jika volume dan berat sampah masih berada dalam batas aman, sistem akan terus beroperasi secara berulang dan melakukan pemantauan secara *real-time*, sehingga proses pengambilan sampah dapat berjalan secara aman, terkontrol, dan efisien.

3.2 Diagram Alir Penelitian

Flow chart penelitian digunakan untuk menggambarkan tahapan dan alur pelaksanaan penelitian secara sistematis mulai dari tahap awal hingga tahap akhir. Alur ini disusun untuk memastikan bahwa proses penelitian berjalan secara terstruktur, terencana, dan sesuai dengan tujuan yang telah ditetapkan. Dengan adanya flow chart penelitian, setiap tahapan dapat dipahami dengan jelas, baik dari

proses identifikasi masalah, perancangan sistem, hingga tahap pengujian dan analisis hasil penelitian. Flow chart penelitian yang digunakan pada penelitian ini ditunjukkan pada Gambar 3.5.



Gambar 3. 5 Flow Chart Penelirian

Berdasarkan flow chart pada Gambar 3.5, penelitian diawali dengan tahap identifikasi masalah melalui studi literatur yang bertujuan untuk memahami kondisi sistem yang telah ada serta menemukan permasalahan yang relevan dengan topik penelitian. Tahap selanjutnya adalah perumusan masalah dan dilanjutkan dengan desain serta perancangan sistem yang disesuaikan dengan kebutuhan penelitian, kemudian dikonsultasikan kepada pembimbing untuk memperoleh masukan dan perbaikan. Setelah desain disetujui, dilakukan perancangan perangkat keras dan

perangkat lunak, dilanjutkan dengan pembuatan alat dan pemrograman *software*. Sistem yang telah dibuat kemudian diuji baik dari sisi alat maupun *web* untuk memastikan seluruh komponen bekerja sesuai dengan rencana. Apabila sistem belum bekerja dengan baik, maka dilakukan perbaikan dengan kembali ke tahap perancangan, sedangkan jika sistem telah bekerja sesuai rencana, penelitian dilanjutkan ke tahap pengambilan data, analisis data, hingga akhirnya penelitian dinyatakan selesai.

3.3. Perencanaan dan Desain

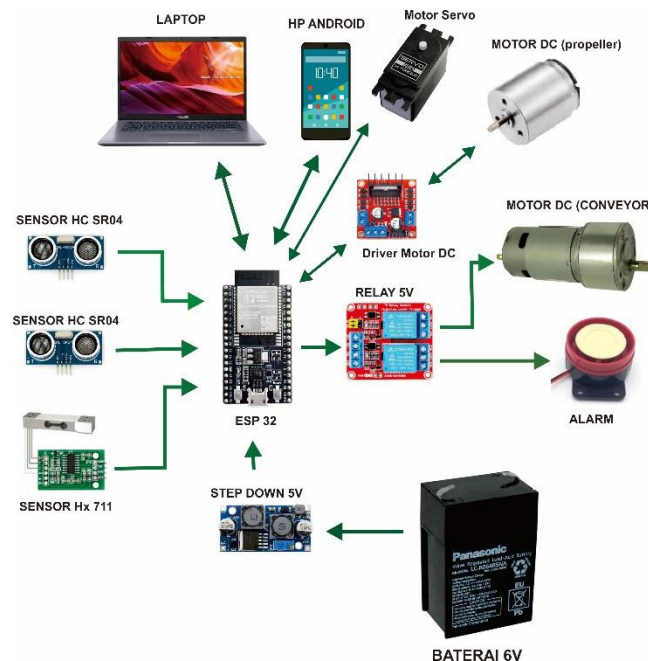
Bagian ini membahas tahap perencanaan dan desain sebagai dasar dalam pengembangan sistem yang diusulkan. Tahapan ini bertujuan untuk memastikan bahwa sistem yang dirancang sesuai dengan kebutuhan dan spesifikasi yang telah ditentukan. Perencanaan dilakukan secara menyeluruh dengan mempertimbangkan aspek teknis dan fungsional sistem. Desain sistem meliputi perancangan perangkat keras dan perancangan perangkat lunak yang saling terintegrasi. Selain itu, pada sub bab ini juga disajikan desain prototipe sebagai gambaran awal dari sistem yang akan dibangun.

3.3.1 Perancangan Hardware

Perancangan perangkat keras dalam penelitian ini diarahkan untuk mewujudkan sistem pengukuran volume dan berat sampah yang bekerja secara otomatis serta terhubung dengan sistem pemantauan berbasis web. Penyusunan rancangan hardware didasarkan pada kebutuhan sistem dalam melakukan akuisisi data volume dan berat sampah secara waktu nyata, memproses data yang diperoleh, serta menghasilkan keluaran berupa pengoperasian *conveyor* dan aktivasi alarm sebagai indikator peringatan. Seluruh elemen perangkat keras dirancang agar dapat saling terhubung dan bekerja secara optimal, efektif, serta sejalan dengan penerapan konsep Internet of Things (IoT) yang telah dibahas pada bagian sebelumnya.

Secara garis besar, sistem perangkat keras terdiri atas tiga bagian utama, yaitu bagian masukan, bagian pemrosesan, dan bagian keluaran. Bagian masukan bertugas mendeteksi kondisi fisik sampah, bagian pemrosesan berfungsi sebagai pusat kendali sekaligus pengolah data, sedangkan bagian keluaran berperan sebagai

penggerak dan sarana pemberian peringatan serta pemantauan sistem. Rancangan perangkat keras sistem secara keseluruhan ditampilkan pada Gambar 3.6.



Gambar 3. 6 Perancangan Hardware

Berdasarkan Gambar 3.6, susunan kerja serta peran setiap komponen pada sistem perancangan perangkat keras dapat diuraikan sebagai berikut :

a. Sensor Ultrasonik HC-SR04

Sensor ultrasonik digunakan sebagai elemen masukan yang berfungsi untuk mendeteksi jarak antara sensor dan permukaan sampah di dalam wadah penampungan.

- 1) Sensor dipasang pada bagian atas penampungan untuk memperoleh data jarak permukaan sampah secara langsung.
- 2) Nilai jarak yang terukur dimanfaatkan sebagai acuan dalam menentukan tingkat kepenuhan atau estimasi volume sampah.
- 3) Penerapan lebih dari satu sensor dilakukan untuk memperoleh hasil pengukuran yang lebih akurat.
- 4) Data hasil pembacaan sensor diteruskan ke mikrokontroler untuk diproses pada tahap selanjutnya.

b. Sensor Berat (*Load cell*) dengan Modul *HX711*

Sensor berat berfungsi untuk mendeteksi besarnya beban sampah yang tertampung di dalam sistem.

- 1) Sensor berat ditempatkan pada bagian bawah penampungan untuk membaca perubahan beban yang terjadi.
- 2) Sinyal keluaran dari sensor berupa sinyal analog yang diperkuat dan diubah menjadi data digital melalui modul *HX711*.
- 3) Data berat yang telah dikonversi selanjutnya dikirimkan ke mikrokontroler sebagai bahan pemrosesan.
- 4) Informasi berat digunakan sebagai parameter dalam menentukan kondisi aman maupun kondisi kelebihan beban.

c. Mikrokontroler ESP32

Mikrokontroler berperan sebagai unit utama yang mengatur dan mengolah seluruh data dalam sistem.

- 1) Mikrokontroler menerima data masukan yang berasal dari sensor jarak dan sensor berat.
- 2) Data tersebut diolah untuk mengevaluasi kondisi volume dan berat sampah berdasarkan nilai ambang yang telah ditentukan.
- 3) Hasil pengolahan data digunakan untuk mengendalikan perangkat keluaran seperti *relay*, motor, dan alarm.
- 4) Mikrokontroler juga mengirimkan informasi hasil pemantauan ke sistem monitoring yang dapat diakses melalui laptop maupun perangkat *Android*.

d. Modul Relay 5V

Modul relay digunakan sebagai media penghubung antara mikrokontroler dan perangkat keluaran yang membutuhkan daya lebih besar.

- 1) Relay menerima sinyal kontrol yang berasal dari mikrokontroler.
- 2) Relay bekerja sebagai saklar elektronik untuk mengatur kondisi aktif atau nonaktif motor dan alarm.
- 3) Penggunaan relay bertujuan untuk menjaga keamanan mikrokontroler dari pengaruh arus dan tegangan tinggi.
- 4) Relay memastikan perangkat keluaran beroperasi sesuai dengan perintah sistem.

e. Motor DC (*Conveyor*)

Motor DC digunakan sebagai aktuator untuk menggerakkan sistem *conveyor* sampah.

- 1) Motor berfungsi menggerakkan *conveyor* dalam proses pemindahan dan pengaturan posisi sampah.
- 2) Pengoperasian motor dikendalikan oleh *relay* berdasarkan perintah dari mikrokontroler.
- 3) Sistem *conveyor* membantu distribusi sampah agar lebih merata di dalam penampungan.
- 4) Motor akan berhenti beroperasi apabila sistem mendeteksi kondisi penuh atau beban berlebih.

f. Sistem Penggerak dan Kemudi

Sistem penggerak menggunakan motor DC yang terhubung dengan *propeller* sedangkan kemudi menggunakan motor servo.

g. Alarm

sistem.

- 1) Alarm akan aktif ketika volume atau berat sampah telah melampaui batas yang ditentukan.
- 2) Peringatan diberikan dalam bentuk suara sebagai indikasi kondisi tidak normal.
- 3) Aktivasi alarm dikendalikan oleh mikrokontroler melalui modul relay.
- 4) Alarm berfungsi membantu operator dalam merespons kondisi sistem secara cepat.

h. Catu Daya (Baterai 6V dan Step Down 5V)

Catu daya berperan sebagai sumber energi listrik bagi seluruh rangkaian sistem.

- 1) Baterai digunakan sebagai sumber daya utama untuk menyuplai sistem.
- 2) Modul penurun tegangan digunakan untuk menyesuaikan nilai tegangan sesuai kebutuhan komponen.
- 3) Tegangan yang stabil diperlukan agar sistem dapat bekerja secara optimal.

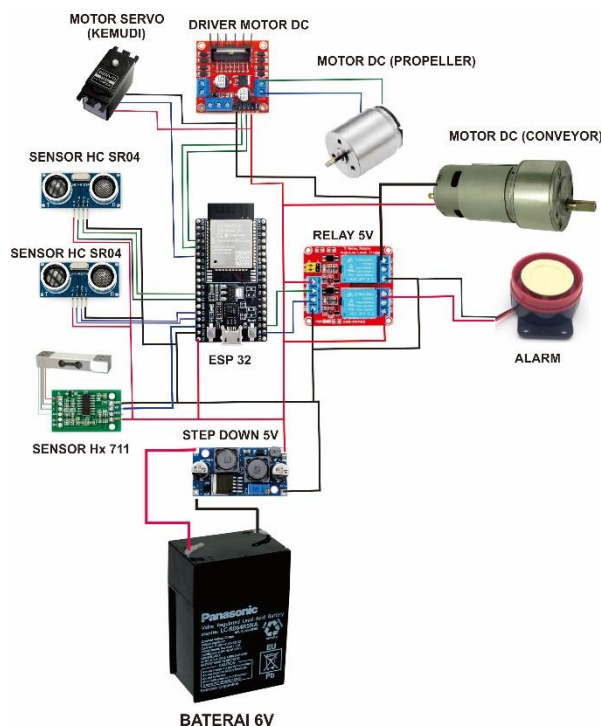
- 4) Rancangan catu daya dibuat agar mampu mendukung pengoperasian sistem dalam jangka waktu tertentu.

i. Perangkat Monitoring (Laptop dan HP *Android*)

Perangkat *monitoring* berfungsi sebagai antarmuka pengguna dalam memantau kondisi sistem.

- 1) Perangkat digunakan untuk menampilkan data volume dan berat sampah secara waktu nyata.
- 2) Informasi sistem dapat diakses melalui jaringan internet menggunakan *web server*.
- 3) Perangkat memungkinkan pemantauan sistem tanpa harus berada langsung di lokasi alat.
- 4) Monitoring membantu pengguna dalam melakukan pengawasan dan pengambilan keputusan secara efektif.

Diagram pengawatan *hardware* dapat dilihat pada gambar 3.7 di bawah ini beserta table koneksi di table 3.1.



Gambar 3. 7 Diagram Pengawatan Hardware

Tabel 3. 1 Koneksi PIN IO ESP32

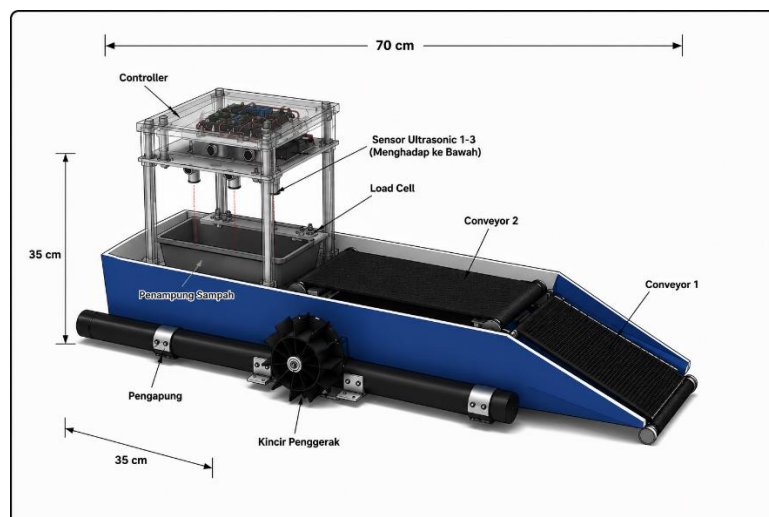
No	Komponen	Pin ESP32	Fungsi
1	Sensor Ultrasonik HC-SR04 1	GPIO 14	Pin trigger sensor ultrasonik pertama
		GPIO 27	Pin echo sensor ultrasonik pertama
		5V	Catu daya sensor
		GND	Ground sensor
2	Sensor Ultrasonik HC-SR04 2	GPIO 26	Pin trigger sensor ultrasonik kedua
		GPIO 25	Pin echo sensor ultrasonik kedua
		5V	Catu daya sensor
		GND	Ground sensor
3	Sensor Berat (HX711)	GPIO 32	Pin data (DT) sensor berat
		GPIO 33	Pin clock (SCK) sensor berat
		5V	Catu daya modul HX711
		GND	Ground modul HX711

No	Komponen	Pin ESP32	Fungsi
4	Modul Relay 5V (Motor)	GPIO 18	Sinyal kendali relay motor
5	Modul Relay 5V (Alarm)	GPIO 19	Sinyal kendali relay alarm
6	Modul Relay 5V	5V	Catu daya modul relay
		GND	Ground modul relay
7	Motor DC (Conveyor)	—	Digendalikan melalui relay
8	Motor DC Propeler	D34 dan D35	Menggerakkan baling-baling
9	Alarm	—	Diaktifkan melalui relay
10	Motor Servo	D32	Penggerak kemudi kapal
11	Step Down 5V	5V (VIN ESP32)	Menyuplai tegangan ke ESP32
		GND	Ground sistem
12	Baterai 6V	—	Sumber daya utama sistem

3.3.2 Desain hardware

Desain alat pada penelitian ini disusun untuk mendukung penerapan sistem pengukuran volume dan berat sampah secara otomatis pada prototipe kapal pengambil sampah. Rancangan alat dibuat dengan mempertimbangkan kondisi operasional kapal, keterbatasan ruang, serta kebutuhan integrasi antara sensor, aktuator, dan sistem kendali. Desain ini bertujuan agar proses pengambilan, penampungan, dan pemantauan sampah dapat dilakukan secara efektif, akurat, dan terkontrol tanpa ketergantungan pada pengukuran manual oleh operator.

Secara umum, alat terdiri dari bagian mekanik dan bagian elektronik yang saling terintegrasi. Bagian mekanik berfungsi sebagai media fisik pengambilan dan penampungan sampah, sedangkan bagian elektronik berperan dalam proses pendeteksian, pengolahan data, serta pengendalian sistem. Tata letak komponen dirancang sedemikian rupa agar alur masuknya sampah, proses pengukuran, serta distribusi beban dapat berlangsung secara optimal. Desain alat sistem ditunjukkan pada Gambar 3.8.



Gambar 3. 8 Desain Prototipe Alat

Berdasarkan desain maka ukuran dari prototipe adalah sebagai berikut :

- Lebar prototipe kapal adalah 30cm
- Tinggi total 35 cm
- Tinggi Bak sampah 17 cm

- Panjang 70 cm

b. Conveyor 1

Conveyor 1 merupakan bagian awal pada sistem mekanik yang berfungsi sebagai media pemindahan sampah dari air menuju bak penampungan melalui *conveyor 2*.

- 1) *Conveyor* dirancang dengan posisi miring untuk memudahkan proses pengangkutan sampah dari air ke *conveyor 2*.
- 2) Mekanisme *conveyor* membantu mengatur aliran sampah agar tidak menumpuk pada satu titik tertentu.
- 3) Penggunaan *conveyor* memungkinkan proses pengambilan sampah berlangsung secara kontinu.

Conveyor digerakkan secara otomatis sesuai dengan kendali sistem.

c. Conveyor 2

Conveyor 2 merupakan bagian utama pada sistem mekanik yang berfungsi sebagai media pemindahan sampah dari area pengambilan menuju bak penampungan.

- 1) *Conveyor* dirancang dengan posisi miring untuk memudahkan proses pengangkutan sampah ke dalam penampungan.
- 2) Mekanisme *conveyor* membantu mengatur aliran sampah agar tidak menumpuk pada satu titik tertentu.
- 3) Penggunaan *conveyor* memungkinkan proses pengambilan sampah berlangsung secara kontinu.
- 4) *Conveyor* digerakkan secara otomatis sesuai dengan kendali sistem.

d. Motor DC

Motor DC berfungsi sebagai penggerak *conveyor* pada sistem.

- 1) Motor dipasang pada bagian samping *conveyor* untuk menggerakkan sabuk atau bidang *conveyor*.
- 2) Putaran motor mengatur kecepatan pemindahan sampah menuju penampungan.
- 3) Motor bekerja berdasarkan perintah dari sistem kendali.

- 4) Pengoperasian motor dihentikan apabila sistem mendeteksi kondisi tertentu, seperti penampungan penuh atau beban berlebih.

e. Sensor Ultrasonik

Sensor ultrasonik digunakan untuk mendeteksi ketinggian permukaan sampah di dalam bak penampungan. Ketinggian sampah dalam penampungan pada prototipe ini ditentukan setinggi 17cm. Nilai batas 17 cm dipilih berdasarkan dimensi efektif bak penampungan. Tinggi fisik bak adalah sekitar 20 cm, sedangkan sensor ultrasonik dipasang pada bagian atas dan memerlukan ruang bebas (*dead zone*) sekitar 3 cm agar pembacaan tetap stabil. Dengan demikian tinggi maksimum sampah ditetapkan sebesar 17 cm sebagai batas alarm.

- 1) Sensor ditempatkan pada bagian atas penampungan dengan arah pengukuran menghadap ke permukaan sampah.
- 2) Data jarak yang diperoleh digunakan untuk menentukan tingkat kepenuhan penampungan.
- 3) Penggunaan lebih dari satu sensor bertujuan meningkatkan keakuratan estimasi volume sampah.
- 4) Informasi hasil pengukuran dikirimkan ke sistem kendali untuk diproses lebih lanjut.

f. Sensor Berat

Sensor berat berfungsi untuk mengukur beban sampah yang berada di dalam bak penampungan. Pada prototipe ini berat maksimal ditentukan adalah sebesar 2 Kg. Sensor *load cell* memiliki kapasitas maksimum 3 kg, namun penelitian menetapkan batas operasi sebesar 2 kg. Penetapan ini memberikan faktor keamanan sehingga sensor tetap bekerja pada daerah linier, menjaga akurasi pembacaan, menghindari *overload*, dan memperpanjang umur pakai sensor.

- 1) Sensor dipasang pada bagian dasar penampungan sebagai titik tumpu beban.

- 2) Perubahan beban sampah akan terdeteksi sebagai perubahan nilai berat.
- 3) Data berat digunakan untuk mengetahui kondisi aman atau kelebihan beban pada penampungan.
- 4) Informasi berat menjadi salah satu parameter pengendalian sistem.

g. Kemudi dan *Propeller*

Kemudi dan propeller memnentukan arah Gerakan kapal pengangkut sampah, seluruh pengendalian dilakukan dari web (*android* atau PC)

h. Unit Kendali (Controller)

Unit kendali merupakan pusat pengolahan data dan pengendalian sistem.

- 1) Unit kendali menerima data dari sensor ultrasonik dan sensor berat.
- 2) Data yang diterima diolah untuk menentukan kondisi volume dan berat sampah.
- 3) Unit kendali mengatur kerja motor *conveyor* dan sistem peringatan.
- 4) Informasi kondisi sistem dapat dikirimkan ke perangkat monitoring untuk pemantauan secara *online*.

3.3.3 Perancangan Software

a. Arduino IDE

Perancangan perangkat lunak *Arduino* difokuskan pada pengembangan program yang berfungsi untuk mengendalikan seluruh proses kerja sistem secara otomatis. Program ini dirancang agar mikrokontroler mampu membaca data dari sensor ultrasonik dan sensor berat secara berkala, kemudian mengolah data tersebut untuk menentukan kondisi volume dan berat sampah. Selain itu, perangkat lunak *Arduino* juga

berperan dalam mengatur logika pengambilan keputusan berdasarkan nilai ambang yang telah ditentukan, sehingga sistem dapat bekerja sesuai dengan kebutuhan operasional.

Perangkat lunak *Arduino* juga dirancang untuk mengendalikan perangkat keluaran seperti motor *conveyor* dan alarm melalui modul *relay*. Ketika sistem mendeteksi kondisi penampungan penuh atau beban melebihi batas aman, program akan memberikan perintah untuk menghentikan atau menyesuaikan kerja motor serta mengaktifkan alarm sebagai peringatan. Selain itu, data hasil pembacaan sensor dan status sistem dikirimkan secara berkala untuk ditampilkan pada sistem *monitoring* berbasis *web*, sehingga kondisi sistem dapat dipantau secara *real-time*.

b. Web Browser dan Kendali Berbasis ESP32

Perancangan perangkat lunak *web browser* dilakukan dengan memanfaatkan kemampuan *ESP32* sebagai *web server* lokal. Pada sistem ini, *ESP32* menjalankan layanan *web* yang dapat diakses melalui *web browser* tanpa memerlukan *server* eksternal. Antarmuka *web* dirancang untuk menampilkan informasi kondisi sistem, seperti nilai *volume* dan berat sampah, status kerja *conveyor*, serta kondisi alarm secara waktu nyata. Dengan pendekatan ini, pemantauan sistem dapat dilakukan menggunakan perangkat seperti *laptop* atau *smartphone* yang terhubung ke jaringan *ESP32*.

Selain sebagai media *monitoring*, *web browser* berbasis *ESP32* juga dirancang untuk mendukung fungsi kendali sistem. Melalui antarmuka *web*, pengguna dapat memberikan perintah kepada sistem, seperti mengaktifkan atau menonaktifkan proses kerja sesuai kebutuhan. Perintah yang diberikan melalui *web browser* akan diproses langsung oleh *ESP32*, sehingga memungkinkan terjadinya komunikasi dua arah antara pengguna dan sistem. Integrasi ini menjadikan sistem lebih fleksibel, efisien, dan sesuai dengan konsep *Internet of Things (IoT)* tanpa ketergantungan pada infrastruktur *server* tambahan.

3.4 Rencana Pengujian

Rencana pengujian dilakukan untuk memastikan bahwa sistem alarm berbasis sensor jarak dan sensor berat pada prototipe kapal pengambil sampah dapat bekerja sesuai dengan spesifikasi dan tujuan penelitian. Pengujian difokuskan pada kinerja sensor, pemrosesan data oleh mikrokontroler ESP32, fungsi alarm, serta tampilan monitoring berbasis *web server*. Pengujian dilakukan secara bertahap meliputi pengujian komponen, pengujian integrasi sistem, dan pengujian fungsional keseluruhan.

3.4.1 Skenario Pengujian

Skenario pengujian adalah Langkah-langkah pengujian yang dilakukan untuk memperoleh data hasil pengujian. Pengujian yang akan dilakukan adalah sebagai berikut:

1. Pengujian sensor ultrasonik untuk mendeteksi kinerja pengukuran jarak permukaan sampah.
2. Pengujian sensor berat (*load cell*) untuk menguji stabilitas sensor dalam mengukur beban sampah.
3. Pengujian integrasi sensor jarak dan berat.
4. Pengujian sistem alarm saat melewati batas.
5. Pengujian sistem *conveyor*.
6. Pengujian *web server*.

3.4.2 Tabel Rencana Data Uji

Untuk memperoleh data hasil pengujian yang memadai sebagai bahan analisi, diperlukan rencana data hasil pengujian. Tabel di bawah ini menunjukkan rencana data hasil pengujian :

Tabel 3. 2 Rencana data uji

No	Objek Uji	Parameter	Kondisi Uji	Data Masukan	Output Diharapkan
1	Sensor Ultrasonik	Jarak	Kosong	Jarak > batas	Status kosong
2	Sensor Ultrasonik	Jarak	Penuh	Jarak $\leq$ batas	Alarm aktif
3	Sensor Berat	Berat	Normal	< batas	Normal
4	Sensor Berat	Berat	Overload	> batas	Alarm aktif
5	Integrasi	Volume & berat	Normal	Normal	Data tampil web

(Halaman ini sengaja dikosongkan)

BAB 4

HASIL DAN PEMBAHASAN

Bab ini menguraikan proses pengujian alat yang dilakukan secara terpisah maupun secara terintegrasi, serta pembahasan terhadap data yang telah diperoleh pada subbab sebelumnya. Kegiatan pengujian dan analisis tersebut bertujuan untuk mengevaluasi tingkat keberhasilan kinerja alat secara menyeluruh. Setelah sebagian besar alat telah dibuat maka alat akan dijabarkan sesuai dengan sistem yang telah direncanakan pada Bab sebelumnya.

Perancangan miniatur dilakukan sebagai media implementasi sistem monitoring kapasitas dan pemilahan sampah berbasis ESP32. Miniatur dirancang menyerupai tempat sampah yang dilengkapi dengan *conveyor* sebagai jalur perpindahan sampah sertaudukan untuk pemasangan sensor dan komponen pendukung lainnya.



Gambar 4. 1 Hasil Perancangan Miniatur

Berdasarkan Gambar 4.1, miniatur telah selesai dirancang dan seluruh bagian mekanik terpasang sesuai dengan desain yang telah dibuat. Miniatur ini digunakan sebagai media pengujian untuk mengetahui kinerja sistem monitoring, pemilahan, dan pengendalian *conveyor* pada berbagai kondisi pengujian.

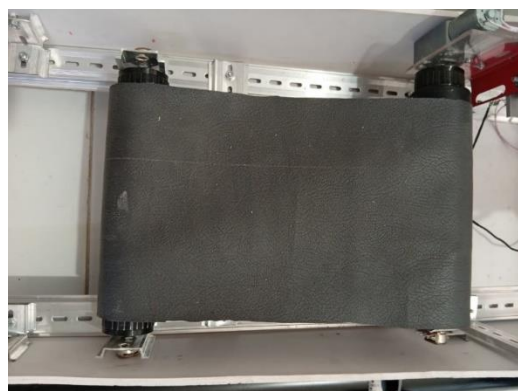
Penyusunan rangkaian controller dilakukan dengan menghubungkan seluruh komponen elektronik, seperti ESP32, relay, sensor, buzzer, dan modul pendukung lainnya ke dalam satu rangkaian kontrol. Seluruh komponen dipasang pada panel akrilik agar lebih rapi, mudah dalam proses pengujian, serta memudahkan perawatan sistem.



Gambar 4. 2 Rangkaian Controller

Berdasarkan Gambar 4.2, rangkaian controller telah terpasang dengan baik dan seluruh komponen saling terhubung sesuai dengan perancangan sistem. Rangkaian ini berfungsi sebagai pusat kendali yang memproses data dari sensor, mengendalikan aktuator, serta mengirimkan informasi hasil monitoring ke Serial Monitor dan *web server* secara real-time.

Penyusunan *conveyor* dilakukan sebagai media untuk memindahkan sampah menuju area penampungan secara otomatis. *Conveyor* dirancang menggunakan sabuk (belt) yang digerakkan oleh motor DC sehingga mampu mengangkat sampah sesuai dengan perintah dari sistem.

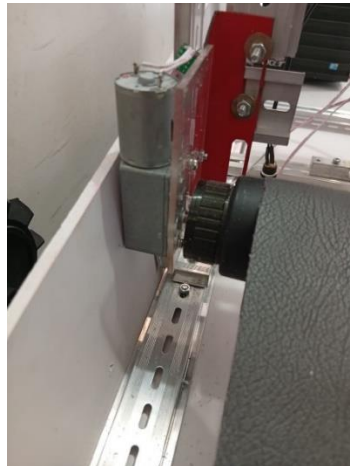


Gambar 4. 3 *Conveyor*

Berdasarkan Gambar 4.3, *conveyor* telah terpasang dengan baik pada rangka miniatur dan siap digunakan sebagai media perpindahan sampah selama proses

pengujian sistem.

Motor penggerak dipasang pada salah satu sisi *conveyor* untuk menghasilkan putaran yang menggerakkan sabuk *conveyor*. Motor dikendalikan oleh ESP32 melalui rangkaian driver sehingga dapat beroperasi secara otomatis maupun manual sesuai dengan program.



Gambar 4. 4 Motor Penggerak *Conveyor*

Berdasarkan Gambar 4.4, motor penggerak telah terpasang dengan baik dan terhubung dengan *conveyor* sehingga mampu menggerakkan sabuk *conveyor* selama proses pengujian. Motor bekerja sesuai dengan perintah yang diberikan oleh ESP32 melalui rangkaian pengendali, sehingga *conveyor* dapat beroperasi secara otomatis maupun manual sesuai dengan program yang telah dirancang. Hal ini menunjukkan bahwa sistem penggerak *conveyor* telah berfungsi dengan baik dan siap digunakan pada seluruh skenario pengujian sistem.

Sensor load cell dipasang pada bagian dasar tempat penampungan untuk mengukur berat sampah yang masuk ke dalam tempat sampah. Pengukuran dilakukan berdasarkan perubahan gaya tekan yang diterima sensor akibat adanya beban sampah. Data hasil pengukuran kemudian dikirimkan ke ESP32 untuk diproses sebagai salah satu parameter monitoring kapasitas. Nilai berat tersebut selanjutnya digunakan oleh sistem untuk menentukan kondisi tempat sampah serta ditampilkan pada *Serial Monitor* dan *web server* secara *real-time*.



Gambar 4. 5 Sensor *Load cell*

Berdasarkan Gambar 4.5, sensor load cell telah terpasang dengan baik pada rangka miniatur sehingga dapat digunakan untuk mendeteksi perubahan berat sampah selama proses pengujian.

Sensor ultrasonik dipasang pada bagian atas tempat sampah untuk mengukur jarak antara sensor dengan permukaan sampah. Perubahan jarak yang terbaca digunakan untuk menentukan tingkat kapasitas tempat sampah secara real-time.



Gambar 4. 6 Sensor Ultrasonik

Berdasarkan Gambar 4.6, sensor ultrasonik telah terpasang dengan baik pada posisi pengukuran sehingga dapat mendeteksi perubahan ketinggian sampah dan mengirimkan data ke ESP32 untuk diproses lebih lanjut.

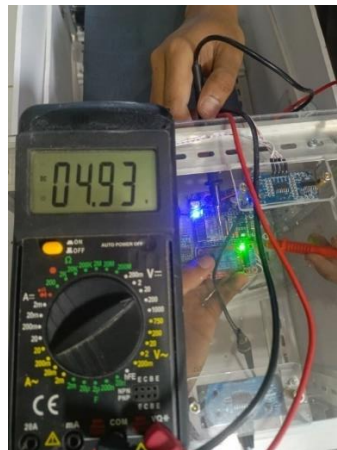
Pada Gambar 4.1 sampai dengan Gambar 4.6 ditampilkan hasil perakitan sementara yang meliputi miniatur alat, rangkaian controller, *conveyor*, motor penggerak *conveyor*, sensor *load cell*, dan sensor ultrasonik. Seluruh bagian tersebut menunjukkan bahwa perancangan mekanik dan rangkaian utama telah disusun sesuai kebutuhan sistem monitoring volume dan berat sampah pada prototipe kapal.

4.1 Pengujian Parsial

Pengujian dilakukan untuk memastikan bahwa mikrokontroler mendapatkan suplai tegangan yang stabil dan sesuai dengan kebutuhan operasionalnya. Pengujian ini penting karena kestabilan tegangan akan mempengaruhi kinerja sistem secara keseluruhan.

4.1.1 Pengujian ESP32

Pengujian ESP32 dilakukan untuk memastikan bahwa mikrokontroler mendapatkan suplai tegangan yang stabil dan sesuai dengan kebutuhan operasionalnya. Pengujian ini penting karena kestabilan tegangan akan mempengaruhi kinerja sistem secara keseluruhan.



Gambar 4. 7 Pengujian Tegangan ESP32

Sumber : Dokumentasi penelitian

Pengujian pada Gambar 4.7 dilakukan dengan cara mengukur tegangan pada pin VIN ESP32 menggunakan multimeter digital. Sumber tegangan berasal dari modul step-down yang telah diturunkan dari adaptor 12V. Pengukuran dilakukan dalam kondisi sistem aktif untuk mengetahui kestabilan tegangan saat ESP32 beroperasi. Setelah dilakukan pengujian sebanyak 10 kali, diperoleh hasil sebagai berikut :

Tabel 4. 1 Pengujian Tegangan ESP32

No	Pengujian	Hasil (V)	Tegangan Ideal (V)	Selisih (V)	Persentase Error (%)
1	Pengujian 1	4,90	5,00	0,10	2,00
2	Pengujian 2	4,89	5,00	0,11	2,20
3	Pengujian 3	4,91	5,00	0,09	1,80
4	Pengujian 4	4,90	5,00	0,10	2,00
5	Pengujian 5	4,90	5,00	0,10	2,00

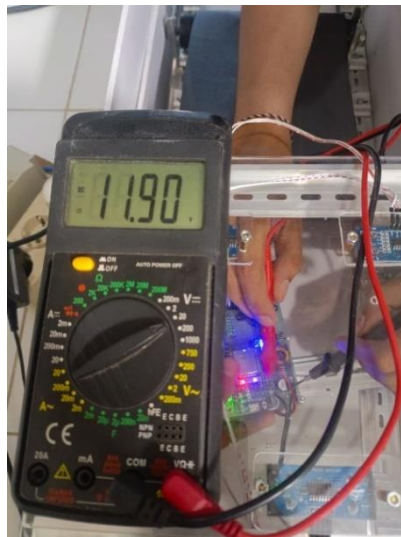
6	Pengujian 6	4,92	5,00	0,08	1,60
7	Pengujian 7	4,89	5,00	0,11	2,20
8	Pengujian 8	4,90	5,00	0,10	2,00
9	Pengujian 9	4,91	5,00	0,09	1,80
10	Pengujian 10	4,90	5,00	0,10	2,00

Sumber : Dokumen penelitian

Berdasarkan Tabel 4.1, hasil pengujian menunjukkan bahwa tegangan yang diterima ESP32 berada pada kisaran 4,89 V hingga 4,92 V dengan nilai rata-rata sebesar 4,902 V. Selisih terhadap tegangan ideal 5 V berkisar antara 0,08 V hingga 0,11 V, sedangkan rata-rata selisih pengukuran sebesar 0,098 V. Perhitungan persentase *error* menghasilkan nilai antara 1,60% hingga 2,20%, dengan rata-rata *error* sebesar 1,96%. Nilai *error* yang relatif kecil menunjukkan bahwa modul catu daya mampu memberikan tegangan yang stabil dan masih berada dalam batas toleransi kerja ESP32, sehingga mikrokontroler dapat beroperasi secara optimal tanpa mengalami gangguan akibat penurunan tegangan.

4.1.2 Pengujian Regulator Tegangan (Step Down)

Pengujian regulator tegangan dilakukan untuk mengetahui kemampuan modul step-down dalam menurunkan dan menstabilkan tegangan dari sumber utama sebelum didistribusikan ke komponen lain.



Gambar 4. 8 Pengujian Regulator Tegangan

Sumber : Dokumentasi penelitian

Pengujian dilakukan dengan mengukur tegangan output dari modul step-down menggunakan multimeter. Tegangan input berasal dari adaptor sekitar 12V, kemudian diturunkan menjadi sekitar 5V. Pengukuran dilakukan berulang sebanyak 10 kali untuk memastikan kestabilan output :

Tabel 4. 2 Pengujian Regulator Tegangan

No	Pengujian	Hasil (V)	Tegangan Ideal (V)	Selisih (V)	Persentase Error (%)
1	Pengujian 1	4,91	5,00	0,09	1,80
2	Pengujian 2	4,90	5,00	0,10	2,00
3	Pengujian 3	4,92	5,00	0,08	1,60
4	Pengujian 4	4,89	5,00	0,11	2,20
5	Pengujian 5	4,90	5,00	0,10	2,00
6	Pengujian 6	4,91	5,00	0,09	1,80
7	Pengujian 7	4,90	5,00	0,10	2,00
8	Pengujian 8	4,89	5,00	0,11	2,20
9	Pengujian 9	4,91	5,00	0,09	1,80
10	Pengujian 10	4,90	5,00	0,10	2,00

Sumber : Dokumentasi penelitian

Berdasarkan Tabel 4.2, hasil pengujian menunjukkan bahwa tegangan keluaran regulator berada pada kisaran 4,89 V hingga 4,92 V dengan nilai rata-rata sebesar 4,903 V. Selisih terhadap tegangan ideal 5 V berkisar antara 0,08 V hingga 0,11 V, sedangkan rata-rata selisih yang diperoleh sebesar 0,097 V. Hasil perhitungan menunjukkan bahwa nilai persentase *error* berada pada rentang 1,60% hingga 2,20%, dengan rata-rata *error* sebesar 1,94%.

Nilai *error* yang relatif kecil tersebut menunjukkan bahwa modul *step-down* regulator mampu menurunkan sekaligus menjaga kestabilan tegangan keluaran dengan baik. Fluktuasi tegangan yang terjadi masih berada dalam batas toleransi yang umum dijumpai pada modul regulator DC-DC switching dan dapat dipengaruhi oleh karakteristik komponen elektronik, ketelitian alat ukur, maupun perubahan beban selama proses pengujian. Dengan demikian, regulator tegangan dapat dinyatakan bekerja dengan baik dan mampu menyediakan catu daya yang stabil sehingga layak digunakan sebagai sumber tegangan untuk ESP32 maupun sensor-sensor yang membutuhkan suplai sekitar 5 V.

4.1.3 Pengujian Sensor Ultrasonik

Pengujian sensor ultrasonik dilakukan untuk memastikan bahwa sensor mendapatkan tegangan kerja yang sesuai sehingga mampu mendeteksi objek dengan akurat.



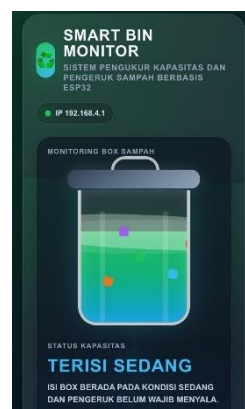
Gambar 4. 9 Pengujian Tegangan Sensor Ultrasonik

Sumber : Dokumentasi penelitian

Pengujian sensor ultrasonik dilakukan dengan mengukur tegangan supply pada pin VCC sensor menggunakan multimeter. Sensor dihubungkan dengan ESP32 dan diberi catu daya dari step-down. Pengujian dilakukan saat sensor dalam kondisi aktif.



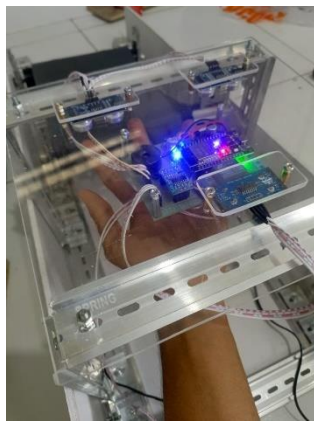
(a)



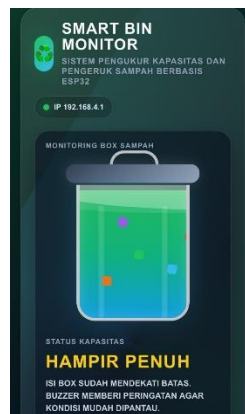
(b)

Gambar 4. 10 Pengujian Sensor Ultrasonic Terisi Sedang
Sumber : Dokumentasi penelitian

Pengujian sensor ultrasonik dilakukan dengan mendeteksi permukaan objek pada jarak 16 cm untuk memastikan sensor mampu membaca jarak dengan baik. Pengujian dilakukan dengan menghubungkan sensor ke ESP32, kemudian hasil pembacaan ditampilkan pada *web server*. Berdasarkan Gambar 4.10, sensor berhasil mendeteksi jarak 16 cm dan *web server* menampilkan kondisi Terisi Sedang sesuai dengan level yang telah ditentukan.



(a)



(b)

Gambar 4. 11 Pengujian Ultrasonic Hampir Penuh
Sumber : Dokumentasi penelitian

Pengujian sensor ultrasonik dilakukan dengan mendekatkan permukaan objek sehingga level sampah berada pada kondisi Hampir Penuh. Sensor membaca perubahan jarak kemudian mengirimkan data ke ESP32 untuk diproses dan ditampilkan pada *web server*. Berdasarkan Gambar 4.11, sistem berhasil mendeteksi kondisi Hampir Penuh dan menampilkan status tersebut pada *web server* secara real-time, sehingga menunjukkan bahwa sensor ultrasonik bekerja dengan baik sesuai perubahan level yang terjadi.

Tabel 4. 3 Pengujian Sensor Ultrasonik

No	Jarak Penggaris			Sensor Ultrasonic 1		Sensor Ultrasonic 2		Sensor Ultrasonic 3	
	Sensor 1	Sensor 2	Sensor 3	Rata-rata	Selisih	Rata-rata	Selisih	Rata-rata	Selisih
1	16.5	16.5	17	16,3	0.1	16,6	0.1	16,9	0.1
2	8.5	12.5	16.5	8,5	0	12,4	0.1	16,7	0.2
3	10.0	13.0	16.0	10.0	0	12..9	0.1	16.1	0.1
4	3.0	5.0	5.0	NO	0	5.0	0	4.9	0.1

Sumber : Dokumentasi penelitian

Berdasarkan Tabel 4.3, hasil pengujian menunjukkan bahwa sensor ultrasonik mampu mengukur jarak dengan tingkat akurasi yang baik. Nilai selisih (error) antara jarak referensi yang diukur menggunakan penggaris dan hasil pembacaan sensor berada pada rentang 0 cm hingga 0,2 cm.

1. Sensor Ultrasonik 1 menunjukkan selisih sebesar 0 cm hingga 0,1 cm,
2. Sensor Ultrasonik 2 memiliki selisih 0 cm hingga 0,1 cm, sedangkan
3. Sensor Ultrasonik 3 memiliki selisih terbesar yaitu 0,2 cm pada salah satu pengujian.

Secara keseluruhan, rata-rata galat yang dihasilkan masih sangat kecil dibandingkan jarak pengukuran yang dilakukan, sehingga dapat disimpulkan bahwa ketiga sensor memiliki tingkat akurasi yang baik dan bekerja secara normal. Perbedaan hasil pengukuran yang muncul kemungkinan disebabkan oleh karakteristik sensor ultrasonik, sudut pantulan gelombang suara, kondisi permukaan objek, serta toleransi pengukuran pada alat ukur referensi. Selain itu, pada pengujian keempat Sensor Ultrasonik 1 tidak mendeteksi objek (NO) karena jarak objek berada di luar batas minimum pembacaan sensor, namun kondisi tersebut tidak memengaruhi kesimpulan umum bahwa sistem sensor ultrasonik mampu mendeteksi jarak dengan baik dan menghasilkan galat yang relatif rendah.

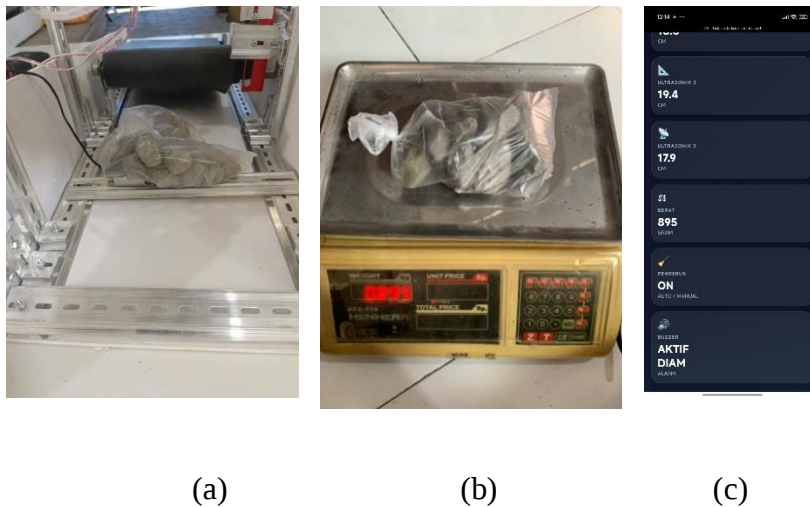
4.1.4 Pengujian Modul *Load cell*

Pengujian *load cell* dilakukan untuk memastikan modul HX711 dan sensor dapat bekerja dengan tegangan yang stabil sehingga pembacaan berat dapat dilakukan dengan baik.



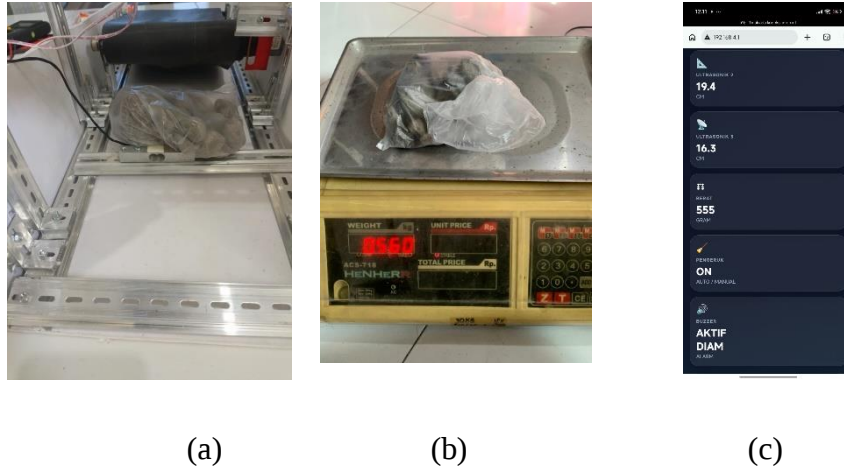
Gambar 4. 12 Pengujian Tegangan Load cell
Sumber : Dokumentasi penelitian

Pengujian *load cell* dilakukan dengan mengukur tegangan pada modul HX711 menggunakan multimeter. Modul dihubungkan ke ESP32 dan diberi catu daya. Pengukuran dilakukan untuk memastikan tegangan kerja modul sesuai spesifikasi.



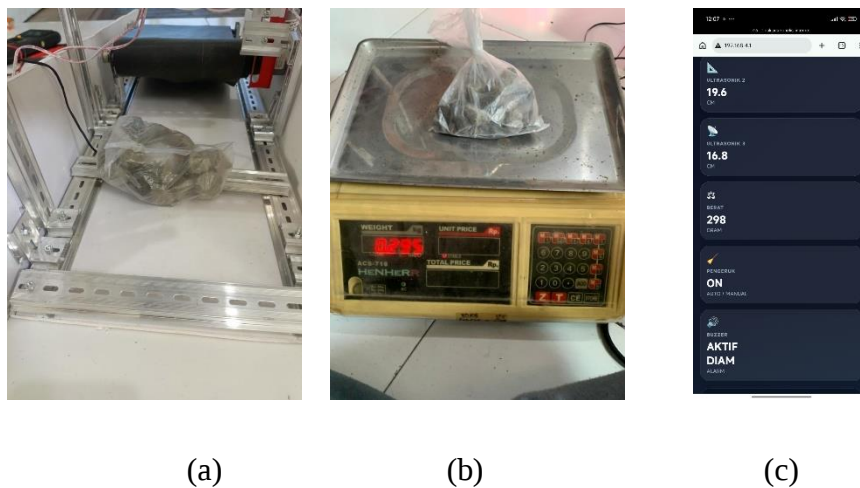
Gambar 4. 13 Pengujian Beban 895 gram
Sumber : Dokumentasi penelitian

Pengujian dilakukan dengan memberikan beban sebesar 895 gram pada prototipe untuk mengetahui respons sensor terhadap perubahan berat. Data hasil pembacaan kemudian diproses oleh ESP32 dan ditampilkan pada *web server*. Berdasarkan Gambar 4.13, sistem berhasil membaca beban yang diberikan dan menampilkan nilai pengukuran pada *web server* secara real-time.



Gambar 4. 14 Pengujian Beban 560 gram

Pengujian dilakukan dengan memberikan beban sebesar 560 gram pada prototipe untuk menguji kemampuan sensor dalam mendeteksi perubahan berat. Hasil pembacaan dikirim ke ESP32 dan ditampilkan pada *web server*. Berdasarkan Gambar 4.14, sistem mampu mendeteksi beban dengan baik serta menampilkan informasi hasil pengukuran secara real-time.



Gambar 4. 15 Pengujian Beban 295 Gram

Pengujian dilakukan dengan memberikan beban sebesar 295 gram pada prototipe untuk memastikan sensor tetap dapat melakukan pembacaan pada beban yang lebih kecil. Data hasil pengukuran diproses oleh ESP32 dan ditampilkan pada

web server. Berdasarkan Gambar 4.15, sistem berhasil membaca perubahan beban dan menampilkan hasil pengukuran sesuai dengan kondisi yang diberikan secara real-time.

Tabel 4. 4 Pengujian Beban *Load cell*

No	Pengujian	Hasil Berat (gram)	Beban Asli (gram)	Selisih (gram)	Persentase Error (%)
1	Pengujian 1	298	295	3	1,02
2	Pengujian 2	296	295	1	0,34
3	Pengujian 3	294	295	1	0,34
4	Pengujian 4	295	295	0	0,00
5	Pengujian 5	297	295	2	0,68
6	Pengujian 6	555	560	5	0,89
7	Pengujian 7	559	560	1	0,18
8	Pengujian 8	561	560	1	0,18
9	Pengujian 9	560	560	0	0,00
10	Pengujian 10	558	560	2	0,36
11	Pengujian 11	895	895	0	0,00
12	Pengujian 12	894	895	1	0,11
13	Pengujian 13	897	895	2	0,22
14	Pengujian 14	896	895	1	0,11
15	Pengujian 15	893	895	2	0,22

Berdasarkan Tabel 4.4, hasil pengujian menunjukkan bahwa sensor load cell mampu mengukur berat dengan tingkat akurasi yang baik pada berbagai variasi beban. Pengujian dilakukan sebanyak 15 kali menggunakan tiga variasi beban asli, yaitu 295 gram, 560 gram, dan 895 gram, dengan masing-masing beban diuji sebanyak lima kali. Hasil pengukuran menunjukkan bahwa nilai yang diperoleh sangat mendekati berat sebenarnya, dengan selisih pengukuran berada pada rentang 0 hingga 5 gram. Nilai persentase *error* yang diperoleh berkisar antara 0,00% hingga 1,02%, dengan rata-rata *error* sebesar 0,31%. Nilai *error* yang relatif kecil tersebut menunjukkan bahwa sensor memiliki tingkat ketelitian yang tinggi dalam mengukur berat. Perbedaan hasil pengukuran yang masih muncul dapat dipengaruhi oleh proses kalibrasi sensor, posisi peletakan beban pada permukaan load cell, resolusi modul HX711, serta toleransi komponen elektronik yang digunakan. Secara keseluruhan, hasil pengujian membuktikan bahwa sensor load cell bekerja secara normal, stabil, dan memiliki tingkat akurasi yang baik sehingga layak digunakan sebagai sensor pengukuran berat pada sistem yang dirancang.

4.1.5 Pengujian Motor Penggerak Conveyor

Pengujian motor dilakukan untuk mengetahui apakah motor mendapatkan suplai tegangan yang cukup saat beroperasi dalam menggerakkan conveyor.



Gambar 4. 16 Pengujian Motor DC

Pengujian motor dilakukan dengan mengukur tegangan pada terminal motor saat motor dalam kondisi menyala. Motor diaktifkan melalui relay yang dikendalikan oleh ESP32. Pengukuran dilakukan menggunakan multimeter pada kondisi beban kerja normal.

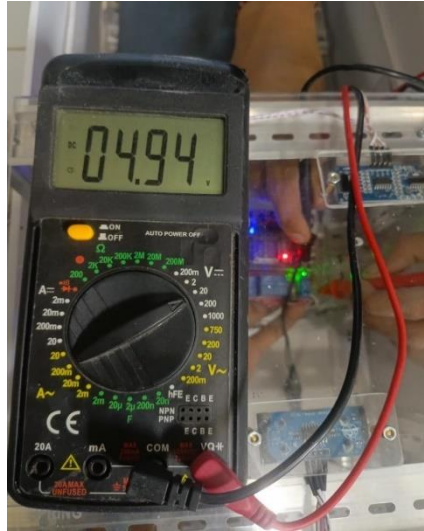
Tabel 4. 5 Pengujian Tegangan Motor DC

No	Pengujian	Hasil (V)
1	Pengujian 1	12,22
2	Pengujian 2	12,21
3	Pengujian 3	12,23
4	Pengujian 4	12,22
5	Pengujian 5	12,20
6	Pengujian 6	12,22
7	Pengujian 7	12,21
8	Pengujian 8	12,23
9	Pengujian 9	12,22
10	Pengujian 10	12,21

Berdasarkan Tabel 4.5, tegangan motor berada pada kisaran 12,20 V hingga 12,23 V. Nilai ini menunjukkan bahwa motor mendapatkan suplai tegangan yang cukup dan stabil untuk beroperasi. Dengan tegangan tersebut, motor mampu menggerakkan conveyor secara optimal.

4.1.6 Pengujian Relay

Pengujian relay dilakukan untuk memastikan bahwa komponen ini dapat bekerja dengan baik dalam proses penghubung dan pemutus arus listrik pada sistem.



Gambar 4. 17 Pengujian Tegangan Relay Saat Aktif

Pengujian relay dilakukan dengan mengukur tegangan pada input relay saat relay dalam kondisi aktif. Relay dikendalikan oleh ESP32, dan pengukuran dilakukan menggunakan multimeter untuk memastikan tegangan pemicu relay sesuai.

Tabel 4. 6 Pengujian Tegangan Relay

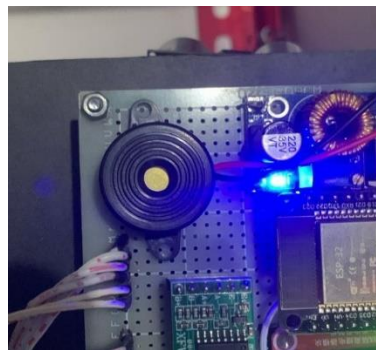
No	Pengujian	Hasil (V)
1	Pengujian 1	4,90
2	Pengujian 2	4,91
3	Pengujian 3	4,90
4	Pengujian 4	4,89
5	Pengujian 5	4,90
6	Pengujian 6	4,91
7	Pengujian 7	4,90
8	Pengujian 8	4,92
9	Pengujian 9	4,89
10	Pengujian 10	4,90

Berdasarkan Tabel 4.6, tegangan pada relay berada pada kisaran 4,89 V hingga 4,92 V. Hal ini menunjukkan bahwa relay mendapatkan tegangan yang cukup untuk aktif dan bekerja dengan baik dalam proses switching. Relay dapat berfungsi secara stabil sebagai pengendali beban.

Berdasarkan seluruh hasil pengujian parsial terutama di tegangan tiap komponen, dapat disimpulkan bahwa setiap komponen utama dalam sistem telah bekerja dengan baik dan berada dalam kondisi stabil, sehingga sistem layak untuk dilanjutkan ke tahap pengujian fungsi tiap komponen dan pengujian terintegrasi.

4.1.7 Pengujian *Buzzer* (Alarm)

Pengujian buzzer atau alarm dilakukan untuk memastikan bahwa indikator peringatan dapat aktif ketika kondisi volume maupun berat sampah melewati batas yang telah ditentukan.



Gambar 4. 18 Pengujian Alarm

Pengujian *buzzer* atau alarm dilakukan untuk memastikan bahwa indikator peringatan dapat aktif ketika kondisi volume maupun berat sampah melewati batas yang telah ditentukan.

Table 4. 7 Pengujian *Buzzer*

No	Tegangan Ke <i>Buzzer</i>	Hasil
1	4.9 V	<i>Buzzer</i> berbunyi
2	0 V	<i>Buzzer</i> Mati

Berdasarkan hasil pengujian pada tabel, *buzzer* dapat berfungsi sesuai dengan kondisi tegangan yang diberikan. Pada saat *buzzer* menerima tegangan sebesar 4,9 V, *buzzer* berhasil berbunyi sebagai indikator peringatan. Sebaliknya, ketika tidak diberikan tegangan (0 V), *buzzer* berada dalam kondisi mati dan tidak menghasilkan suara. Hasil ini menunjukkan bahwa rangkaian kontrol *buzzer* bekerja dengan baik dalam mengaktifkan maupun menonaktifkan alarm sesuai perintah sistem. Dengan demikian, *buzzer* dapat digunakan sebagai indikator

peringatan yang andal untuk memberikan notifikasi ketika volume atau berat sampah telah mencapai batas yang ditentukan. Adanya alarm suara ini membantu operator dalam memantau kondisi tempat sampah secara lebih cepat dan efektif sehingga tindakan penanganan dapat segera dilakukan ketika kondisi penuh atau overload terdeteksi oleh sistem.

4.2 Pengujian Terintegrasi

Pengujian terintegrasi dilakukan setelah seluruh komponen sistem diuji secara parsial. Pengujian ini bertujuan untuk memastikan bahwa seluruh komponen seperti sensor ultrasonik, sensor berat (*load cell*), ESP32, sistem *conveyor*, alarm, dan *web server* dapat bekerja secara bersamaan sesuai dengan perancangan pada Bab 3. Pengujian dilakukan berdasarkan skenario yang telah direncanakan, yaitu pengujian integrasi sensor, sistem alarm, *conveyor*, dan Monitoring berbasis web.

4.2.1 Pengujian Integrasi Sensor Ultrasonik dan Sensor Berat

Pengujian ini bertujuan untuk mengetahui kemampuan sistem dalam membaca data volume (jarak) dan berat secara bersamaan serta menampilkannya pada sistem Monitoring.

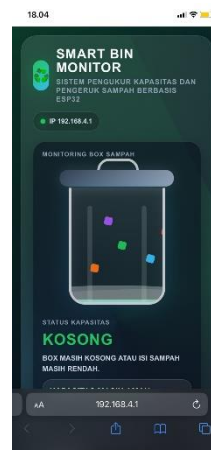
Langkah pengujian :

2. Sistem diaktifkan dan seluruh sensor dalam kondisi siap.
3. Objek sampah dimasukkan ke dalam penampungan secara bertahap.
4. Pengujian Kondisi Kosong
5. Pengujian Kondisi Sedang:
 - Kosong
 - Setengah penuh
 - Penuh

```

MODE           : AUTO
PENGGERUK      : OFF
BUZZER ENABLE  : AKTIF
BUZZER OUTPUT  : DIAM
JARAK 1        : 19.6 CM
JARAK 2        : 19.7 CM
JARAK 3        : NO DATA
JARAK RATA-RATA : 19.6 CM
BERAT          : 46 GRAM
KAPASITAS MAX  : 2000 GRAM
LEVEL JARAK    : 0.0 %
LEVEL BERAT    : 2.3 %
LEVEL INTERNAL : 2.3 %
STATUS         : KOSONG
CLIENT WIFI    : 1

```



(a)

(b)

Gambar 4. 19 Tampilan Serial Monitor (a) dan (b) Web server Kondisi Kosong

Sumber: Dokumentasi penelitian

Pengujian dilakukan dengan mengamati data yang ditampilkan pada Serial Monitor ketika tempat sampah berada pada kondisi kosong. Data yang ditampilkan meliputi hasil pembacaan sensor ultrasonik, berat sampah, persentase kapasitas, status level, serta estimasi waktu. Berdasarkan Gambar 4.19, seluruh parameter berhasil ditampilkan dengan baik pada Serial Monitor, sehingga menunjukkan bahwa ESP32 mampu memproses dan mengirimkan data hasil pembacaan sensor secara normal.

```

MODE           : AUTO
PENGGERUK      : ON
BUZZER ENABLE  : AKTIF
BUZZER OUTPUT  : DIAM
JARAK 1        : 3.0 CM
JARAK 2        : 2.3 CM
JARAK 3        : NO DATA
JARAK RATA-RATA : 2.6 CM
BERAT          : 46 GRAM
KAPASITAS MAX  : 2000 GRAM
LEVEL JARAK    : 96.3 %
LEVEL BERAT    : 2.3 %
LEVEL INTERNAL : 96.3 %
STATUS         : HAMPIR PENUH
CLIENT WIFI    : 1

```



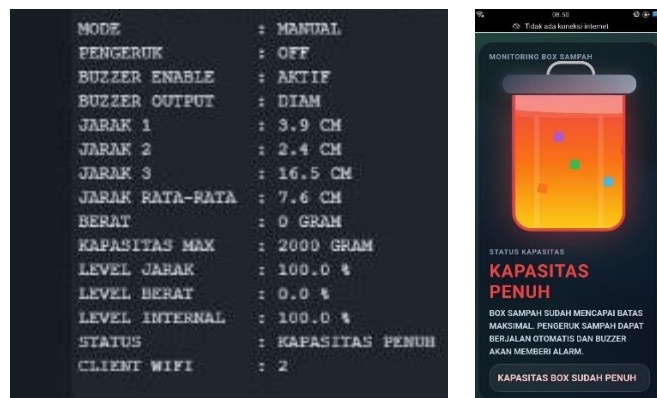
(a)

(b)

Gambar 4. 20 Tampilan Serial Monitor (a) dan (b) Web server Kondisi Hampir Penuh

Sumber: Dokumentasi penelitian

Pengujian dilakukan untuk memastikan website monitoring mampu menampilkan kondisi kapasitas tempat sampah secara real-time. Data yang diterima dari ESP32 ditampilkan dalam bentuk indikator level, persentase kapasitas, serta status kondisi tempat sampah. Berdasarkan Gambar 4.20, website berhasil menampilkan kondisi Terisi Sedang sesuai dengan data hasil pembacaan sensor, sehingga pengguna dapat memantau kapasitas tempat sampah melalui antarmuka web dengan baik.



(a)

(b)

Gambar 4. 21 Serial Monitor Kapasitas Penuh (a) dan (b) Data *Web server*

Sumber: Dokumentasi penelitian

Pengujian dilakukan dengan mengamati data yang ditampilkan pada Serial Monitor ketika tempat sampah berada pada kondisi kapasitas penuh. Data yang ditampilkan meliputi hasil pembacaan sensor ultrasonik, berat sampah, persentase kapasitas, status tempat sampah, serta kondisi aktuator yang dikendalikan oleh ESP32. Berdasarkan Gambar 4.21, Serial Monitor berhasil menampilkan status Kapasitas Penuh beserta nilai hasil pembacaan sensor secara real-time, sehingga menunjukkan bahwa ESP32 mampu memproses dan mengirimkan informasi kondisi tempat sampah dengan baik sesuai keadaan sebenarnya.

Pengujian dilakukan untuk memastikan *web server* mampu menampilkan kondisi tempat sampah ketika berada pada keadaan kosong. Data hasil pembacaan sensor yang diproses oleh ESP32 dikirimkan ke *web server* dan ditampilkan dalam bentuk persentase kapasitas, status tempat sampah, serta mode operasi sistem. Berdasarkan Gambar 4.22, *web server* berhasil menampilkan status Kosong dengan

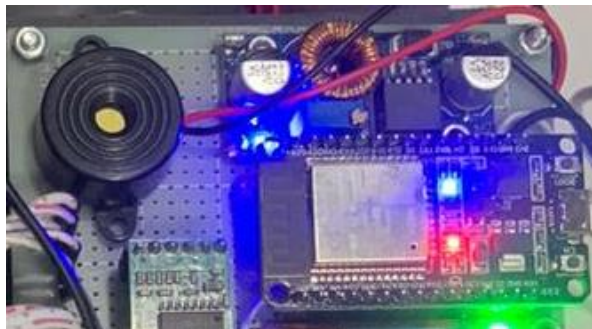
informasi kapasitas masih aman secara real-time, sehingga pengguna dapat memantau kondisi tempat sampah melalui antarmuka web dengan baik.

Berdasarkan tampilan serial monitor dan *web server*, data sensor dapat terbaca secara bersamaan serta menampilkan kondisi kapasitas penampungan. Ketika jarak permukaan sampah dan nilai berat menunjukkan kondisi penuh, sistem memberikan informasi status secara real-time pada halaman monitoring.

4.2.2 Pengujian Sistem Alarm

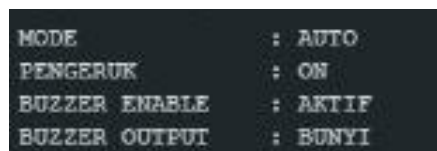
Pengujian ini dilakukan untuk memastikan alarm bekerja ketika kondisi melebihi batas yang telah ditentukan. Langkah pengujian:

- Sistem dioperasikan dalam kondisi normal.
- Sampah ditambahkan hingga:
- Melebihi batas ketinggian (≤ 17 cm).
- Melebihi batas berat (> 2 kg).
- Amati respon sistem.

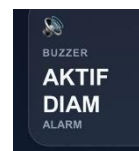


Gambar 4. 22 Alarm Aktif pada ESP32

Sumber: Dokumentasi penelitian



(a)



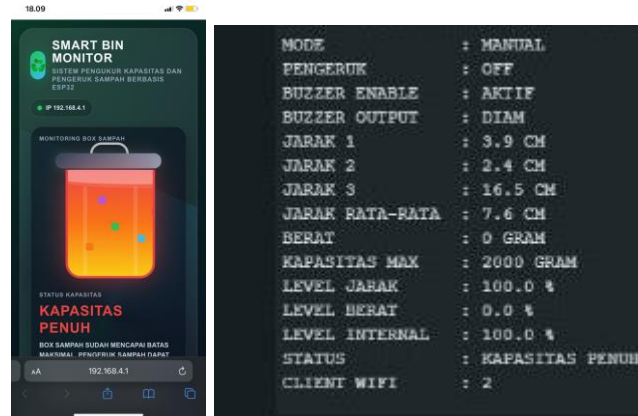
(b)

Gambar 4. 23 Tampilan Serial Monitor dan *Web server* Saat Alarm Aktif

Sumber: Dokumentasi penelitian

Berdasarkan Gambar 4.23, terlihat bahwa buzzer pada rangkaian ESP32 berada dalam kondisi aktif sebagai indikator alarm ketika kapasitas tempat sampah telah mencapai batas yang ditentukan. Indikator LED pada ESP32 juga

menunjukkan bahwa sistem sedang beroperasi.

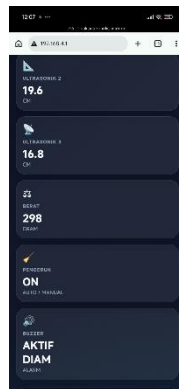


(a)

(b)

Gambar 4. 24 Alarm Pada Kapasitas penuh
Sumber: Dokumentasi penelitian

Berdasarkan Gambar 4.24, Serial Monitor menampilkan status Kapasitas Penuh dengan *Buzzer Output Bunyi*, disertai informasi hasil pembacaan sensor dan kondisi sistem secara *real-time*. Tampilan tersebut menunjukkan bahwa alarm berhasil diaktifkan dan data monitoring dapat ditampilkan dengan baik.



Gambar 4. 25 Tampilan saat alarm aktif
Sumber : Dokumentasi penelitian

Berdasarkan Gambar 4.25, *web server* menampilkan informasi hasil pembacaan sensor ultrasonik, status penggerak, dan kondisi *buzzer* ketika alarm aktif. Tampilan tersebut menunjukkan bahwa sistem berhasil memberikan informasi kondisi tempat sampah secara *real-time* melalui antarmuka *web server*.

Pada pengujian sistem alarm, LED indikator dan *buzzer* aktif ketika parameter pengujian melewati ambang batas. Hal ini menunjukkan bahwa logika

peringatan pada ESP32 telah bekerja sesuai dengan rancangan, yaitu memberikan respon ketika sistem mendeteksi kondisi penuh atau kelebihan beban.

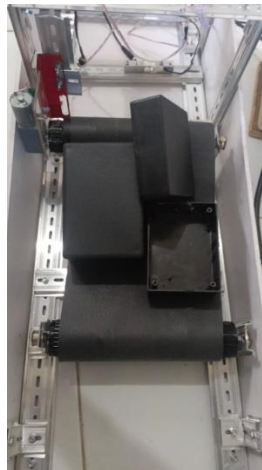
Hasil yang diharapkan:

- Alarm aktif ketika volume penuh.
- Alarm aktif ketika berat melebihi batas.
- Alarm tidak aktif saat kondisi normal.

4.2.3 Pengujian Sistem *Conveyor*

Pengujian ini bertujuan untuk mengetahui apakah *conveyor* dapat bekerja sesuai kondisi sistem.

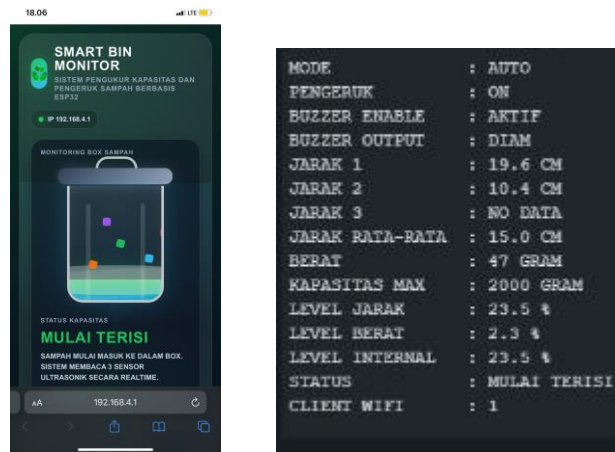
a. Pengujian Keadaan Normal



Gambar 4. 26 *Conveyor* Berjalan Normal

Sumber: Dokumentasi penelitian

Berdasarkan Gambar 4.26, *conveyor* terlihat beroperasi dalam kondisi normal untuk mengangkut sampah menuju area penampungan. Kondisi tersebut menunjukkan bahwa motor penggerak *conveyor* bekerja sesuai dengan perintah yang diberikan oleh ESP32.



(a)

(b)

Gambar 4. 27 Tampilan *Web server* (a) dan (b) Serial Monitor *Conveyor Normal*

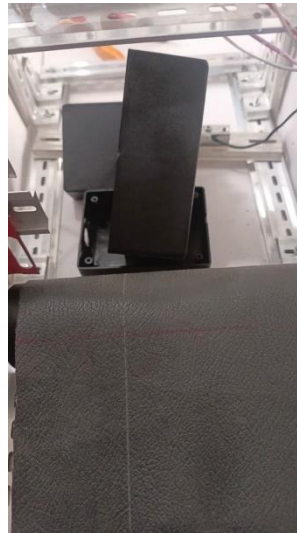
Sumber: Dokumentasi penelitian

Berdasarkan Gambar 4.27 Serial Monitor dan *web server* berhasil menampilkan hasil pembacaan sensor serta kondisi sistem ketika *conveyor* bekerja secara normal. Informasi yang ditampilkan meliputi data sensor, status kapasitas tempat sampah, kondisi *conveyor*, dan status buzzer secara real-time, sehingga pengguna dapat memantau proses monitoring dan kondisi sistem dengan baik melalui kedua media tersebut.

Pada kondisi normal, *conveyor* dapat berjalan untuk membantu proses pemindahan sampah ke bagian penampungan. Ketika status sistem masih berada pada kondisi aman, motor penggerak *conveyor* tetap mendapat suplai dan dapat bekerja sesuai perintah kendali.

b. Pengujian Keadaan Sampah Penuh

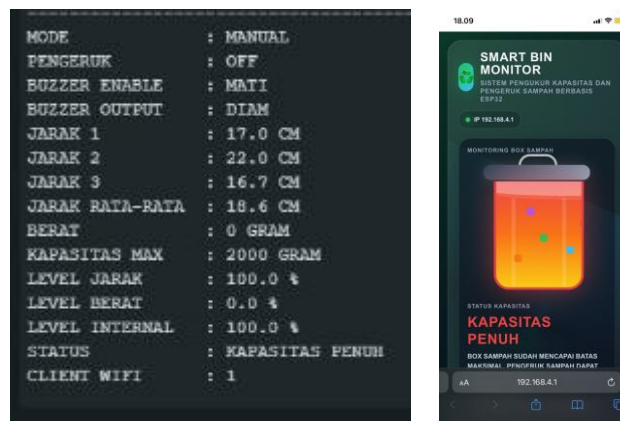
Pengujian keadaan sampah penuh dilakukan untuk memastikan sistem mampu mendeteksi ketika kapasitas tempat sampah telah mencapai batas maksimum. Pada kondisi ini diamati respons sensor, *conveyor*, buzzer, serta tampilan informasi pada Serial Monitor dan *web server* untuk memastikan seluruh komponen bekerja sesuai dengan logika program yang telah dirancang.



Gambar 4. 28 Conveyor Berhenti Saat Sampah Penuh

Sumber: Dokumentasi penelitian

Berdasarkan Gambar 4.28, conveyor berada pada kondisi berhenti setelah kapasitas tempat sampah mencapai batas maksimum. Hal tersebut menunjukkan bahwa sistem berhasil menghentikan conveyor secara otomatis sesuai dengan logika program yang telah dirancang.



(a)

(b)

Gambar 4. 29 Tampilan Serial Monitor (a) dan (b) Web server Saat Sampah Penuh

Sumber: Dokumentasi penelitian

Berdasarkan Gambar 4.29, web server menampilkan hasil pembacaan sensor ultrasonik dan berat sampah ketika tempat sampah berada pada kondisi penuh. Informasi tersebut ditampilkan secara real-time sehingga pengguna dapat

memantau kondisi kapasitas tempat sampah dengan mudah.

Pada kondisi sampah penuh, sistem menghentikan kerja *conveyor* sebagai bentuk pengamanan agar penampungan tidak mengalami kelebihan kapasitas. Respon tersebut memperlihatkan bahwa integrasi antara sensor, sistem kendali, dan aktuator telah berjalan sesuai fungsi yang direncanakan.

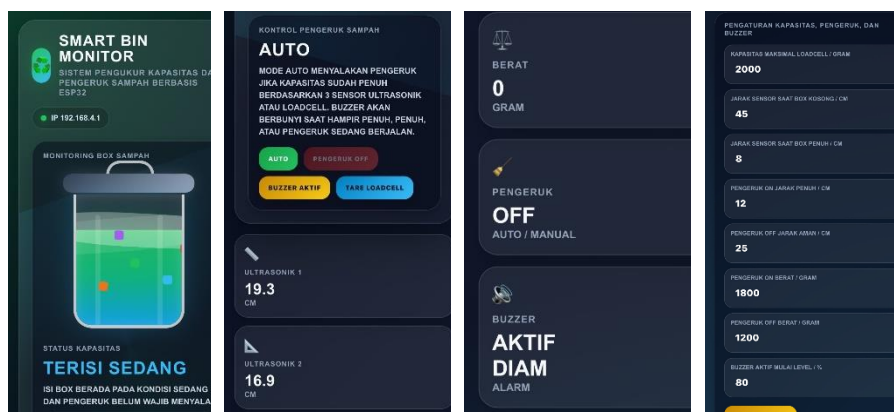
- a. *Conveyor* berjalan saat kondisi aman.
- b. *Conveyor* berhenti saat penampungan penuh atau overload.

4.2.4 Pengujian Web server dan Kendali

Pengujian ini dilakukan untuk memastikan sistem Monitoring berbasis web dapat menampilkan data dan menerima perintah dengan baik.

Langkah pengujian:

- ESP32 dihubungkan ke jaringan.
- Web dibuka melalui browser (HP/Laptop).
- Amati tampilan:
- Nilai jarak
- Nilai berat
- Status *conveyor*
- Status alarm
- Pengujian kendali



(a) (b) (c) (d)

Gambar 4. 30 Tampilan Web server dan Kendali

Sumber: Dokumentasi penelitian

Berdasarkan Gambar 4.30, *web server* menampilkan data hasil pembacaan sensor, berat sampah, persentase kapasitas, serta status Terisi Sedang sesuai dengan kondisi yang terdeteksi pada saat pengujian. Seluruh informasi berhasil ditampilkan secara real-time oleh ESP32. Detail penjelasan sesuai gambar sesuai urutan :

- (a) Tampilan Kondisi penampungan terisi sedang
- (b) Tampilan pengaturan dan data sensor Ultrasonic
- (c) Status Buzzer(alarm)
- (d) Pengaturan parameter di Kapal.

Berdasarkan pengujian web server dan kendali maka dapat terlihat bahwa web server telah berjalan sebagaimana mestinya serta parameter yang diatur dapat berfungsi dengan baik.

4.3 Pembahasan

Berdasarkan hasil pengujian parsial, komponen utama seperti ESP32, regulator tegangan, sensor ultrasonik, modul *load cell*, motor *conveyor*, relay, dan *buzzer* telah mendapatkan suplai tegangan yang stabil. Kestabilan tegangan tersebut menjadi faktor penting karena seluruh proses pembacaan sensor, pengendalian aktuator, dan pengiriman data ke *web server* bergantung pada kinerja catu daya yang baik.

Pada pengujian terintegrasi, sensor ultrasonik dan sensor berat dapat digunakan secara bersamaan untuk membaca kondisi penampungan sampah. Sensor ultrasonik berperan dalam mengetahui jarak permukaan sampah, sedangkan *load cell* digunakan untuk membaca beban sampah. Kombinasi kedua sensor tersebut membantu sistem dalam menentukan kondisi kapasitas penampungan secara lebih informatif dibandingkan hanya menggunakan satu parameter pengukuran.

Sistem alarm bekerja sebagai indikator peringatan ketika nilai hasil pembacaan sensor melewati batas yang telah ditentukan. Ketika kondisi penuh atau overload terdeteksi, *buzzer* dan indikator peringatan aktif sehingga operator dapat

segera mengetahui bahwa penampungan perlu dikosongkan atau proses *conveyor* perlu dihentikan.

Web server pada ESP32 mampu menampilkan data pengujian secara real-time melalui perangkat seperti laptop maupun smartphone. Dengan adanya tampilan *web server*, operator dapat melihat kondisi jarak, berat, status alarm, dan status *conveyor* tanpa harus melakukan pengecekan langsung pada rangkaian alat.

Secara keseluruhan, hasil pengujian menunjukkan bahwa sistem alarm berbasis sensor jarak dan sensor berat dapat mendukung proses pemantauan volume sampah pada prototipe kapal. Sistem ini masih dapat dikembangkan lebih lanjut pada tahap berikutnya melalui pengujian akurasi sensor secara lebih detail dan pengujian langsung pada kondisi operasional yang mendekati keadaan sebenarnya

(Halaman ini sengaja dikosongkan)

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem alarm berbasis sensor jarak dan sensor berat untuk pengukuran volume sampah pada prototipe kapal pengambil sampah dengan *web server*, maka dapat diambil kesimpulan sebagai berikut:

1. Sistem pengukuran volume sampah berbasis sensor jarak dan sensor berat pada prototipe kapal pengambil sampah berhasil dirancang menggunakan tiga sensor ultrasonik HC-SR04 sebagai pendeteksi volume sampah, sensor *load cell* HX711 sebagai pendeteksi berat sampah, serta *mikrokontroler* ESP32 sebagai pusat pengendali sistem. Berdasarkan hasil pengujian pada Bab IV, sensor ultrasonik memiliki selisih pengukuran sebesar 0–0,2 cm, sedangkan sensor *load cell* memiliki selisih pembacaan sebesar 2–4 gram dengan nilai galat berkisar 0,34%–0,45% dibandingkan timbangan digital. Hasil tersebut menunjukkan bahwa kedua sensor memiliki tingkat akurasi yang baik sehingga layak digunakan untuk melakukan pemantauan volume dan berat sampah pada prototipe kapal.
2. Sistem alarm sebagai indikator kapasitas muatan sampah berhasil dirancang dan bekerja sesuai dengan logika pengendalian yang telah dibuat. Berdasarkan hasil pengujian, alarm aktif secara otomatis ketika tinggi sampah mencapai batas maksimum 17 cm atau berat sampah mencapai batas operasi 2 kg, kemudian motor *conveyor* berhenti secara otomatis untuk mencegah penambahan sampah ke dalam penampung. Hasil pengujian integrasi juga menunjukkan bahwa alarm, relay, motor *conveyor*, dan sensor mampu bekerja secara sinkron tanpa mengalami kegagalan fungsi sehingga sistem mampu memberikan peringatan kepada operator ketika kapasitas penampung telah mencapai batas yang ditentukan.
3. Sistem monitoring volume sampah berbasis *web server* menggunakan ESP32 berhasil dirancang dan diimplementasikan sehingga mampu menampilkan informasi volume sampah, berat sampah, status alarm, status *conveyor*, dan

kondisi sistem secara real-time melalui browser pada laptop maupun smartphone. Berdasarkan hasil pengujian Bab IV, seluruh data pembacaan sensor berhasil diproses oleh ESP32 dan dikirim ke *web server* tanpa mengalami kendala komunikasi, sehingga operator dapat memantau kondisi kapal secara langsung. Hal ini menunjukkan bahwa sistem monitoring berbasis *web server* mampu meningkatkan efektivitas proses pemantauan dibandingkan metode manual karena informasi kondisi muatan dapat diketahui secara cepat, akurat, dan terintegrasi.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran untuk pengembangan penelitian selanjutnya, yaitu:

1. Menambahkan sensor lain, seperti sensor GPS atau sensor kemiringan kapal, sehingga sistem dapat memberikan informasi posisi dan kondisi kapal secara lebih lengkap.
2. Mengembangkan sistem monitoring menggunakan server berbasis internet atau cloud sehingga data dapat diakses dari lokasi yang lebih jauh tanpa terbatas pada jaringan lokal ESP32.
3. Melakukan pengujian langsung pada prototipe kapal di lingkungan perairan sebenarnya agar diperoleh data performa sistem yang lebih representatif terhadap kondisi operasional.
4. Menambahkan sumber daya mandiri, seperti panel surya dan sistem manajemen baterai, sehingga waktu operasi kapal menjadi lebih lama dan lebih hemat energi.
5. Mengembangkan sistem kendali otomatis untuk navigasi kapal dan proses pengambilan sampah sehingga seluruh proses dapat berlangsung secara lebih mandiri tanpa ketergantungan pada operator.

DAFTAR PUSTAKA

- Avia Semiconductor. (2018). *HX711 24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales Datasheet*. Avia Semiconductor Technology Co., Ltd. https://cdn.sparkfun.com/datasheets/Sensors/ForceFlex/hx711_english.pdf
- Arduino.cc. (2025). *Software* – Arduino. Arduino. <https://www.arduino.cc/en/software>.
- Bolton, W. (2015). *Mechatronics: Electronic Control Systems in Mechanical and Electrical Engineering* (6th ed.). Harlow: Pearson Education.
- Dwiono, W., & Fakhri. (2022). Rancang bangun miniatur sistem parkir cerdas bertingkat berbasis Internet of Things (IoT) menggunakan ESP32. *Jurnal Teknik Elektro*, Universitas Muhammadiyah Purwokerto.
- ElecFreaks. (2019). Ultrasonic Ranging Module HC-SR04 Datasheet. ElecFreaksTechnology. <https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>.
- Espressif Systems. (2022). *ESP32 series datasheet version 5.2*. Espressif Systems.
- Gunawan, T., Hernawati, E., & Aditya, B. R. (2021). *IoT-Based Waste Height and Weight Monitoring System*. *Journal of Computer Science*, 17(11), 1085–1092. <https://doi.org/10.3844/jcssp.2021.1085.1092>.
- Salamah, A., Kusumanto, R. D., & Evelina. (2023). *Sistem Monitoring Volume dan Berat Sampah pada Alat Pemilah Sampah Organik dan Anorganik Berbasis Internet of Things Menggunakan Aplikasi Blynk*. *Jurnal TELISKA*. <https://doi.org/10.5281/zenodo.8207096>.
- Şen, M., & Mutluer, M. (2025). *A Review of BLDC Motors: Types, Application, Failure Modes and Detection*. *Energies*, 18(24), 6402. <https://doi.org/10.3390/en18246402>
- Mu'afifah, F., Amiruddin, W., & Yudo, H. (2023). *Studi perancangan trash skimmer boat tipe catamaran simetris streamline untuk wilayah perairan Pantai Marina Semarang*. *Jurnal Teknik Perkapalan*, Universitas Diponegoro. <https://ejournal3.undip.ac.id/index.php/naval/article/view/41658>
- Putra, F. B. A., & Hayat, L. (2021). Rancang bangun miniatur sistem parkir cerdas bertingkat berbasis Internet of Things menggunakan ESP32. *Jurnal Riset Rekayasa Elektro (JRRE)*, 3(1), 23–34. Universitas Muhammadiyah Purwokerto. <https://jurnalnasional.ump.ac.id/index.php/JRRE>
- Yama, W. E. Z. (2023). *Monitoring Volume Tempat Sampah di Kelurahan Wosi Madu Raja Menggunakan Mikrokontroler NodeMCU ESP8266 Berbasis*

Internet of Things. RIGGS: Journal of Artificial Intelligence and Digital Business. <https://doi.org/10.31004/riggs.v4i2.159>

Yudo, H., Amiruddin, W., & Mu'afifah, F. (2023). Ping *Jurnal Teknik Perkapalan*, Universitas Diponegoro.

LAMPIRAN

```
// =====
// ESP32 - SMART BIN MONITOR
// FITUR:
// - 3 SENSOR ULTRASONIK
// - HX711 LOADCELL
// - PEMBERUK SAMPAH / CONVEYOR AUTO MANUAL
// - BUZZER ALARM KAPASITAS
// - WEB SERVER ACCESS POINT
// =====

#include <WiFi.h>
#include <WebServer.h>
#include "HX711.h"

// ===== WIFI ACCESS POINT =====
const char* ssid      = "PENGUKUR SAMPAH";
const char* password = "12345678";

WebServer server(80);

// ===== PIN HARDWARE =====
#define CONVEYOR_PIN 23

// BUZZER BARU
#define BUZZER_PIN 2

// ===== SENSOR ULTRASONIK 1 =====
#define TRIG1 18
#define ECHO1 19

// ===== SENSOR ULTRASONIK 2 =====
#define TRIG2 22
#define ECHO2 21

// ===== SENSOR ULTRASONIK 3 BARU =====
#define TRIG3 25
#define ECHO3 26

// ===== HX711 LOAD CELL =====
#define HX_DT  5
#define HX_SCK 4

HX711 scale;

// ===== SETTING OUTPUT =====
// CONVEYOR_LOGIC_DIBALIK = true ARTINYA:
// - TAMPILAN / LOGIKA ON  -> OUTPUT FISIK OFF
// - TAMPILAN / LOGIKA OFF -> OUTPUT FISIK ON
// Jika nanti ingin normal lagi, ubah CONVEYOR_LOGIC_DIBALIK
menjadi false.
const bool CONVEYOR_ACTIVE_LOW    = false;
const bool CONVEYOR_LOGIC_DIBALIK = true;
const bool BUZZER_ACTIVE_LOW      = false;

// ===== STATE SISTEM =====
```

```

bool conveyorState = false;
bool modeAuto      = true;

bool buzzerEnable   = true;
bool buzzerOutputState = false;

// ===== DATA SENSOR =====
float jarak1 = 999;
float jarak2 = 999;
float jarak3 = 999;
float jarakRata = 999;

float berat = 0;

// ===== DATA LEVEL =====
float levelKapasitas = 0;
float levelJarak = 0;
float levelBerat = 0;

// ===== SETTING KAPASITAS =====
float kapasitasMaxGram = 2000.0;
float jarakKosongCM    = 45.0;
float jarakPenuhCM     = 8.0;

// ===== BATAS AUTO Pengeruk =====
float onLimit    = 12.0;
float offLimit   = 25.0;
float beratOn    = 1800.0;
float beratOff   = 1200.0;

// ===== SETTING BUZZER =====
float buzzerLevelOn = 80.0;

// ===== TIMING =====
unsigned long lastSensor = 0;
unsigned long lastConveyor = 0;

const unsigned long sensorDelay = 180;
const unsigned long conveyorDelay = 1000;

// ===== KALIBRASI HX711 =====
float calibration_factor = 420.0;

// ===== FUNGSI BANTU =====
// =====
float constrainFloat(float x, float a, float b) {
    if (x < a) return a;
    if (x > b) return b;
    return x;
}

float mapFloat(float x, float in_min, float in_max, float out_min,
float out_max) {
    if (in_max == in_min) return out_min;
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min;
}

```

```

float maxFloat(float a, float b) {
    if (a > b) return a;
    return b;
}

bool jarakValid(float nilai) {
    return nilai < 900;
}

float filterJarak(float lama, float baru) {
    if (!jarakValid(baru)) return 999;
    if (!jarakValid(lama)) return baru;
    return (lama * 0.70) + (baru * 0.30);
}

// =====
// OUTPUT PENGGERUK / CONVEYOR
// =====
void applyConveyor() {
    // Dibalik khusus conveyor sesuai permintaan:
    // conveyorState true = status ON di web/serial, tetapi output
    // fisik dibuat OFF
    // conveyorState false = status OFF di web/serial, tetapi output
    // fisik dibuat ON
    bool outputConveyor = CONVEYOR_LOGIC_DIBALIK ? !conveyorState :
    conveyorState;

    if (CONVEYOR_ACTIVE_LOW) {
        digitalWrite(CONVEYOR_PIN, outputConveyor ? LOW : HIGH);
    } else {
        digitalWrite(CONVEYOR_PIN, outputConveyor ? HIGH : LOW);
    }
}

// =====
// OUTPUT BUZZER
// =====
void applyBuzzer(bool on) {
    buzzerOutputState = on;

    if (BUZZER_ACTIVE_LOW) {
        digitalWrite(BUZZER_PIN, on ? LOW : HIGH);
    } else {
        digitalWrite(BUZZER_PIN, on ? HIGH : LOW);
    }
}

// =====
// CEK SENSOR JARAK
// =====
bool sensorJarakValid() {
    return jarakValid(jarakRata);
}

bool adaUltrasonikDekat(float batas) {
    if (jarakValid(jarak1) && jarak1 <= batas) return true;
    if (jarakValid(jarak2) && jarak2 <= batas) return true;
}

```

```

    if (jarakValid(jarak3) && jarak3 <= batas) return true;
    return false;
}

bool semuaUltrasonikAman(float batas) {
    bool adaData = false;

    if (jarakValid(jarak1)) {
        adaData = true;
        if (jarak1 <= batas) return false;
    }

    if (jarakValid(jarak2)) {
        adaData = true;
        if (jarak2 <= batas) return false;
    }

    if (jarakValid(jarak3)) {
        adaData = true;
        if (jarak3 <= batas) return false;
    }

    return adaData;
}

// =====
// CEK KAPASITAS PENUH
// =====
bool kapasitasPenuh() {
    bool penuhDariBerat = berat >= kapasitasMaxGram;
    bool penuhDariJarak = sensorJarakValid() && jarakRata <=
jarakPenuhCM;
    bool penuhDariLevel = levelKapasitas >= 100;

    return penuhDariBerat || penuhDariJarak || penuhDariLevel;
}

// =====
// STATUS SAMPAH
// =====
String statusSampah() {
    if (kapasitasPenuh()) return "KAPASITAS PENUH";
    if (levelKapasitas >= 80) return "HAMPIR PENUH";
    if (levelKapasitas >= 40) return "TERISI SEDANG";
    if (levelKapasitas >= 15) return "MULAI TERISI";
    return "KOSONG";
}

// =====
// BACA ULTRASONIK
// =====
float bacaUltrasonik(byte trig, byte echo) {
    digitalWrite(trig, LOW);
    delayMicroseconds(2);

    digitalWrite(trig, HIGH);
    delayMicroseconds(10);
    digitalWrite(trig, LOW);

```

```

    long durasi = pulseIn(echo, HIGH, 18000);

    if (durasi == 0) return 999;

    float cm = durasi * 0.0343 / 2.0;

    if (cm < 2 || cm > 400) return 999;

    return cm;
}

// =====
// UPDATE SENSOR
// =====
void updateSensor() {
    if (millis() - lastSensor < sensorDelay) return;
    lastSensor = millis();

    float r1 = bacaUltrasonik(TRIG1, ECHO1);
    float r2 = bacaUltrasonik(TRIG2, ECHO2);
    float r3 = bacaUltrasonik(TRIG3, ECHO3);

    jarak1 = filterJarak(jarak1, r1);
    jarak2 = filterJarak(jarak2, r2);
    jarak3 = filterJarak(jarak3, r3);

    float totalJarak = 0;
    int jumlahValid = 0;

    if (jarakValid(jarak1)) {
        totalJarak += jarak1;
        jumlahValid++;
    }

    if (jarakValid(jarak2)) {
        totalJarak += jarak2;
        jumlahValid++;
    }

    if (jarakValid(jarak3)) {
        totalJarak += jarak3;
        jumlahValid++;
    }

    if (jumlahValid > 0) {
        jarakRata = totalJarak / jumlahValid;
    } else {
        jarakRata = 999;
    }

    if (sensorJarakValid()) {
        levelJarak = mapFloat(jarakRata, jarakKosongCM, jarakPenuhCM,
0, 100);
        levelJarak = constrainFloat(levelJarak, 0, 100);
    } else {
        levelJarak = 0;
    }
}

```

```

    if (scale.is_ready()) {
        float b = scale.get_units(3);
        berat = (berat * 0.70) + (b * 0.30);
        if (berat < 0) berat = 0;
    }

    if (kapasitasMaxGram > 0) {
        levelBerat = (berat / kapasitasMaxGram) * 100.0;
        levelBerat = constrainFloat(levelBerat, 0, 100);
    } else {
        levelBerat = 0;
    }

    levelKapasitas = maxFloat(levelJarak, levelBerat);
    levelKapasitas = constrainFloat(levelKapasitas, 0, 100);
}

// =====
// UPDATE PENGGERUK / CONVEYOR AUTO
// =====
void updateConveyor() {
    if (!modeAuto) return;
    if (millis() - lastConveyor < conveyorDelay) return;

    bool ultraON = adaUltrasonikDekat(onLimit);
    bool ultraOFF = semuaUltrasonikAman(offLimit);

    bool beratONcond = berat >= beratOn;
    bool beratOFFcond = berat <= beratOff;

    if (!conveyorState && (ultraON || beratONcond ||
    kapasitasPenuh())) {
        conveyorState = true;
        lastConveyor = millis();
    }

    if (conveyorState && ultraOFF && beratOFFcond &&
    !kapasitasPenuh()) {
        conveyorState = false;
        lastConveyor = millis();
    }
}

// =====
// UPDATE BUZZER
// =====
void updateBuzzer() {
    if (!buzzerEnable) {
        applyBuzzer(false);
        return;
    }

    bool penuh = kapasitasPenuh();
    bool hampirPenuh = levelKapasitas >= buzzerLevelOn;

    if (!penuh && !hampirPenuh && !conveyorState) {
        applyBuzzer(false);
    }
}

```

```

    return;
}

unsigned long now = millis();
unsigned long periodBuzzer = 1000;
unsigned long durasiOn = 80;

if (penuh) {
    periodBuzzer = 250;
    durasiOn = 120;
} else if (hampirPenuh) {
    periodBuzzer = 700;
    durasiOn = 100;
} else if (conveyorState) {
    periodBuzzer = 1300;
    durasiOn = 70;
}

bool bunyi = (now % periodBuzzer) < durasiOn;
applyBuzzer(bunyi);
}

// =====
// HALAMAN WEB SERVER
// =====
const char PAGE[] PROGMEM = R"rawliteral(
<!DOCTYPE html>
<html lang="id">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-
scale=1">

<title>SMART BIN MONITOR</title>

<style>
:root{
    --text:#f8fafc;
    --muted:#94a3b8;
    --line:rgba(255,255,255,.14);
    --card:rgba(255,255,255,.075);
    --green:#22c55e;
    --red:#ef4444;
    --blue:#38bdf8;
    --yellow:#facc15;
    --orange:#f97316;
}

*{
    box-sizing:border-box;
    text-transform:uppercase;
}

body{
    margin:0;
    min-height:100vh;
    font-family:Arial, Helvetica, sans-serif;
    color:var(--text);

```

```

    background:
      radial-gradient(circle at top left, rgba(34,197,94,.28),
transparent 30%),
      radial-gradient(circle at bottom right, rgba(56,189,248,.25),
transparent 36%),
      linear-gradient(135deg,#020617,#0f172a,#111827);
  }

.wrap{
  width:100%;
  max-width:1180px;
  margin:auto;
  padding:18px;
}

.hero{
  position:relative;
  overflow:hidden;
  border:1px solid var(--line);
  border-radius:30px;
  padding:22px;
  background:linear-
gradient(135deg,rgba(255,255,255,.11),rgba(255,255,255,.04));
  box-shadow:0 28px 70px rgba(0,0,0,.38);
}

.hero:before{
  content:"";
  position:absolute;
  inset:-90px;
  background:conic-gradient(from 180deg, transparent,
rgba(34,197,94,.16), transparent, rgba(56,189,248,.18),
transparent);
  animation:spin 10s linear infinite;
  opacity:.8;
}

@keyframes spin{
  to{transform:rotate(360deg);}
}

.heroContent{
  position:relative;
  z-index:2;
}

.top{
  display:flex;
  align-items:center;
  justify-content:space-between;
  gap:16px;
  flex-wrap:wrap;
}

.brand{
  display:flex;
  align-items:center;
  gap:14px;

```

```

}

.logo{
  width:60px;
  height:60px;
  display:flex;
  align-items:center;
  justify-content:center;
  border-radius:20px;
  background:linear-gradient(135deg,#22c55e,#38bdf8);
  color:#001018;
  font-size:32px;
  box-shadow:0 0 35px rgba(34,197,94,.35);
}

h1{
  margin:0;
  font-size:clamp(26px,4vw,48px);
  letter-spacing:1px;
  line-height:1.05;
}

.subtitle{
  color:var(--muted);
  font-size:14px;
  margin-top:6px;
  letter-spacing:.8px;
  font-weight:700;
}

.pill{
  display:inline-flex;
  align-items:center;
  gap:8px;
  border:1px solid var(--line);
  padding:10px 14px;
  border-radius:999px;
  background:rgba(0,0,0,.20);
  color:#dbeafe;
  font-size:13px;
  font-weight:900;
}

.dot{
  display:inline-block;
  width:10px;
  height:10px;
  border-radius:50%;
  background:var(--green);
  box-shadow:0 0 20px var(--green);
}

.mainGrid{
  margin-top:20px;
  display:grid;
  grid-template-columns:1.1fr .9fr;
  gap:18px;
}

```

```

.panel{
  border:1px solid var(--line);
  border-radius:26px;
  background:rgba(2,6,23,.50);
  backdrop-filter:blur(14px);
  padding:18px;
  box-shadow:0 14px 35px rgba(0,0,0,.25);
}

.label{
  color:var(--muted);
  font-size:12px;
  text-transform:uppercase;
  letter-spacing:1px;
  font-weight:900;
}

.statusText{
  margin-top:8px;
  font-size:clamp(30px,5vw,50px);
  font-weight:900;
  letter-spacing:1px;
}

.desc{
  color:#cbd5e1;
  line-height:1.55;
  font-size:14px;
  margin-top:8px;
  font-weight:700;
}

.boxArea{
  display:flex;
  align-items:center;
  justify-content:center;
  gap:28px;
  flex-wrap:wrap;
  min-height:350px;
}

.binWrap{
  position:relative;
  margin-top:58px;
}

.binHandle{
  position:absolute;
  top:-58px;
  left:72px;
  width:82px;
  height:26px;
  border:5px solid rgba(255,255,255,.72);
  border-bottom:none;
  border-radius:24px 24px 0 0;
}

```

```

.binLid{
  position:absolute;
  top:-35px;
  left:-18px;
  width:265px;
  height:32px;
  border-radius:15px;
  background:linear-gradient(135deg,#334155,#94a3b8);
  box-shadow:0 0 25px rgba(148,163,184,.25);
}

.bin{
  position:relative;
  width:230px;
  height:270px;
  border:5px solid rgba(255,255,255,.74);
  border-top:none;
  border-radius:0 0 32px 32px;
  background:linear-
gradient(180deg,rgba(255,255,255,.08),rgba(255,255,255,.02));
  box-shadow:
    inset 0 0 25px rgba(255,255,255,.08),
    0 0 45px rgba(56,189,248,.22);
  overflow:hidden;
}

.bin:before{
  content:"";
  position:absolute;
  left:32px;
  top:35px;
  width:9px;
  height:190px;
  border-radius:99px;
  background:rgba(255,255,255,.13);
  z-index:5;
}

.bin:after{
  content:"";
  position:absolute;
  right:32px;
  top:35px;
  width:9px;
  height:190px;
  border-radius:99px;
  background:rgba(255,255,255,.13);
  z-index:5;
}

.liquid{
  position:absolute;
  left:0;
  bottom:0;
  width:100%;
  height:0%;
  transition:height .55s ease;
  background:

```

```

        linear-
gradient(180deg, rgba(34,197,94,.92), rgba(16,185,129,.95), rgba(14,1
65,233,.95));
    box-shadow:
        0 -14px 45px rgba(34,197,94,.40),
        inset 0 0 22px rgba(255,255,255,.18);
}

.wave{
    position:absolute;
    left:-50%;
    top:-28px;
    width:200%;
    height:58px;
    border-radius:43%;
    background:rgba(255,255,255,.30);
    animation:waveMove 2.1s ease-in-out infinite;
}

.wave.two{
    top:-20px;
    opacity:.35;
    animation:waveMove2 2.9s ease-in-out infinite;
}

@keyframes waveMove{
    0%,100%{transform:translateX(0) rotate(0deg);}
    50%{transform:translateX(30px) rotate(4deg);}
}

@keyframes waveMove2{
    0%,100%{transform:translateX(20px) rotate(2deg);}
    50%{transform:translateX(-25px) rotate(-5deg);}
}

.trashBits{
    position:absolute;
    inset:0;
    pointer-events:none;
    z-index:6;
}

.bit{
    position:absolute;
    width:16px;
    height:16px;
    border-radius:4px;
    background:#facc15;
    opacity:.92;
    animation:floaty 3s ease-in-out infinite;
}

.bit.b1{left:38px; bottom:24%; background:#f97316; animation-
delay:.1s;}
.bit.b2{left:92px; bottom:43%; background:#22c55e; animation-
delay:.4s;}
.bit.b3{left:148px; bottom:31%; background:#38bdf8; animation-
delay:.7s;}

```

```

.bit.b4{left:178px; bottom:55%; background:#ef4444; animation-
delay:1s;}
.bit.b5{left:65px; bottom:65%; background:#a855f7; animation-
delay:1.3s;}

@keyframes floaty{
  0%,100%{transform:translateY(0) rotate(0deg);}
  50%{transform:translateY(-8px) rotate(16deg);}
}

.bin.full{
  animation:shake .55s infinite;
  border-color:rgba(239,68,68,.95);
  box-shadow:0 0 60px rgba(239,68,68,.55);
}

.bin.full .liquid{
  background:linear-gradient(180deg,#ef4444,#f97316,#facc15);
  box-shadow:0 -14px 50px rgba(239,68,68,.55);
}

@keyframes shake{
  0%,100%{transform:translateX(0);}
  25%{transform:translateX(-4px);}
  75%{transform:translateX(4px);}
}

.infoBig{
  min-width:270px;
  max-width:390px;
}

.capacityText{
  margin-top:10px;
  padding:13px;
  border-radius:18px;
  border:1px solid var(--line);
  background:rgba(255,255,255,.07);
  font-weight:900;
  color:#e2e8f0;
}

.cards{
  margin-top:18px;
  display:grid;
  grid-template-columns:repeat(6,1fr);
  gap:14px;
}

.card{
  border:1px solid var(--line);
  border-radius:24px;
  background:var(--card);
  padding:16px;
  min-height:122px;
  box-shadow:0 12px 30px rgba(0,0,0,.20);
}

```

```

.icon{
  font-size:25px;
  margin-bottom:8px;
}

.value{
  font-size:27px;
  font-weight:900;
  margin-top:5px;
}

.unit{
  color:var(--muted);
  font-size:13px;
  font-weight:700;
}

.btns{
  display:flex;
  flex-wrap:wrap;
  gap:10px;
  margin-top:16px;
}

button{
  border:none;
  outline:none;
  cursor:pointer;
  color:white;
  font-weight:900;
  letter-spacing:.5px;
  padding:13px 16px;
  border-radius:16px;
  background:#334155;
  box-shadow:0 10px 24px rgba(0,0,0,.25);
}

button:active{
  transform:scale(.98);
}

.btnGreen{
  background:linear-gradient(135deg,#16a34a,#22c55e);
}

.btnRed{
  background:linear-gradient(135deg,#b91c1c,#ef4444);
}

.btnBlue{
  background:linear-gradient(135deg,#0284c7,#38bdf8);
  color:#001018;
}

.btnYellow{
  background:linear-gradient(135deg,#ca8a04,#facc15);
  color:#111827;
}

```

```

button.disabled{
  opacity:.38;
  pointer-events:none;
}

.setting{
  display:grid;
  grid-template-columns:repeat(4,1fr);
  gap:10px;
  margin-top:14px;
}

.inputBox{
  background:rgba(255,255,255,.07);
  border:1px solid var(--line);
  border-radius:16px;
  padding:11px;
}

.inputBox label{
  display:block;
  font-size:11px;
  color:var(--muted);
  text-transform:uppercase;
  font-weight:900;
  margin-bottom:6px;
}

input{
  width:100%;
  border:none;
  outline:none;
  background:transparent;
  color:white;
  font-size:18px;
  font-weight:900;
  text-transform:uppercase;
}

.footer{
  margin-top:16px;
  color:var(--muted);
  font-size:12px;
  text-align:center;
  font-weight:800;
}

@media(max-width:1050px){
  .cards{
    grid-template-columns:repeat(3,1fr);
  }
}

@media(max-width:900px){
  .mainGrid{
    grid-template-columns:1fr;
  }
}

```

```

        .setting{
            grid-template-columns:1fr 1fr;
        }
    }

    @media (max-width:560px) {
        .cards{
            grid-template-columns:1fr;
        }

        .setting{
            grid-template-columns:1fr;
        }

        .bin{
            width:190px;
            height:230px;
        }

        .binLid{
            width:225px;
        }

        .binHandle{
            left:56px;
        }
    }
}
</style>
</head>

<body>
<div class="wrap">

    <section class="hero">
        <div class="heroContent">

            <div class="top">
                <div class="brand">
                    <div class="logo"><img alt="Logo" data-bbox="475 645 495 660"/></div>
                    <div>
                        <h1>SMART BIN MONITOR</h1>
                        <div class="subtitle">SISTEM PENGUKUR KAPASITAS DAN
PENGERUK SAMPAH BERBASIS ESP32</div>
                    </div>
                </div>

                <div class="pill">
                    <span class="dot"></span>
                    <span>IP 192.168.4.1</span>
                </div>
            </div>

            <div class="mainGrid">

                <div class="panel">
                    <div class="label">MONITORING BOX SAMPAH</div>

```

```

<div class="boxArea">
  <div class="binWrap">
    <div class="binHandle"></div>
    <div class="binLid"></div>

    <div class="bin" id="binBox">
      <div class="liquid" id="liquidFill">
        <div class="wave"></div>
        <div class="wave two"></div>
      </div>

      <div class="trashBits" id="trashBits">
        <div class="bit b1"></div>
        <div class="bit b2"></div>
        <div class="bit b3"></div>
        <div class="bit b4"></div>
        <div class="bit b5"></div>
      </div>
    </div>
  </div>

  <div class="infoBig">
    <div class="label">STATUS KAPASITAS</div>
    <div class="statusText"
id="statusSampah">KOSONG</div>
    <div class="desc" id="desc">
      ISI BOX AKAN BERGERAK MENGIKUTI PEMBACAAN 3 SENSOR
      ULTRASONIK DAN LOADCELL.
    </div>

    <div class="capacityText" id="capacityInfo">
      KAPASITAS MASIH AMAN
    </div>
  </div>
</div>

<div class="panel">
  <div class="label">KONTROL PENGGERUK SAMPAH</div>
  <div class="statusText" id="modeText">AUTO</div>

  <div class="desc">
    MODE AUTO MENYALAKAN PENGGERUK JIKA KAPASITAS SUDAH
    PENUH BERDASARKAN 3 SENSOR ULTRASONIK ATAU LOADCELL.
    BUZZER AKAN BERBUNYI SAAT HAMPIR PENUH, PENUH, ATAU
    PENGGERUK SEDANG BERJALAN.
  </div>

  <div class="btns">
    <button id="modeBtn" class="btnGreen"
onclick="toggleMode()">AUTO</button>
    <button id="conveyorBtn" class="btnRed disabled"
onclick="toggleConveyor()">PENGGERUK OFF</button>
    <button id="buzzerBtn" class="btnYellow"
onclick="toggleBuzzer()">BUZZER AKTIF</button>
    <button class="btnBlue" onclick="tareLoadcell()">TARE
    LOADCELL</button>
  </div>

```

```

        </div>
    </div>

    </div>

</div>
</section>

<section class="cards">
    <div class="card">
        <div class="icon">🔊</div>
        <div class="label">ULTRASONIK 1</div>
        <div class="value" id="j1">0</div>
        <div class="unit">CM</div>
    </div>

    <div class="card">
        <div class="icon">🔊</div>
        <div class="label">ULTRASONIK 2</div>
        <div class="value" id="j2">0</div>
        <div class="unit">CM</div>
    </div>

    <div class="card">
        <div class="icon">🔊</div>
        <div class="label">ULTRASONIK 3</div>
        <div class="value" id="j3">0</div>
        <div class="unit">CM</div>
    </div>

    <div class="card">
        <div class="icon">📊</div>
        <div class="label">BERAT</div>
        <div class="value" id="berat">0</div>
        <div class="unit">GRAM</div>
    </div>

    <div class="card">
        <div class="icon">🚚</div>
        <div class="label">PENGGERUK</div>
        <div class="value" id="conveyorText">OFF</div>
        <div class="unit">AUTO / MANUAL</div>
    </div>

    <div class="card">
        <div class="icon">🔔</div>
        <div class="label">BUZZER</div>
        <div class="value" id="buzzerText">MATI</div>
        <div class="unit">ALARM</div>
    </div>
</section>

<section class="panel" style="margin-top:18px;">
    <div class="label">PENGATURAN KAPASITAS, PENGGERUK, DAN
    BUZZER</div>

```

```

<div class="setting">
  <div class="inputBox">
    <label>KAPASITAS MAKSIMAL LOADCELL / GRAM</label>
    <input id="setKapasitasMax" type="number" step="1">
  </div>

  <div class="inputBox">
    <label>JARAK SENSOR SAAT BOX KOSONG / CM</label>
    <input id="setKosong" type="number" step="0.1">
  </div>

  <div class="inputBox">
    <label>JARAK SENSOR SAAT BOX PENUH / CM</label>
    <input id="setPenuh" type="number" step="0.1">
  </div>

  <div class="inputBox">
    <label>PENGGERUK ON JARAK PENUH / CM</label>
    <input id="setOnLimit" type="number" step="0.1">
  </div>

  <div class="inputBox">
    <label>PENGGERUK OFF JARAK AMAN / CM</label>
    <input id="setOffLimit" type="number" step="0.1">
  </div>

  <div class="inputBox">
    <label>PENGGERUK ON BERAT / GRAM</label>
    <input id="setBeratOn" type="number" step="1">
  </div>

  <div class="inputBox">
    <label>PENGGERUK OFF BERAT / GRAM</label>
    <input id="setBeratOff" type="number" step="1">
  </div>

  <div class="inputBox">
    <label>BUZZER AKTIF MULAI LEVEL / %</label>
    <input id="setBuzzerLevel" type="number" step="1">
  </div>
</div>

<div class="btns">
  <button class="btnYellow" onclick="saveSetting()">SIMPAN
SETTING</button>
</div>
</section>

<div class="footer">
  SMART BIN MONITOR | 3 ULTRASONIK | LOADCELL | PENGGERUK SAMPAH
| BUZZER
  <br>
  CLIENT: <span id="client">0</span> | UPTIME: <span
id="uptime">0</span>
</div>

</div>

```

```

<script>
let conveyorNow = "OFF";
let modeNow = "AUTO";
let buzzerEnableNow = "AKTIF";

function $(id){
    return document.getElementById(id);
}

function secToTime(t){
    let h = Math.floor(t / 3600);
    let m = Math.floor((t % 3600) / 60);
    let s = t % 60;
    return h + "J " + m + "M " + s + "D";
}

function setInputValue(id, val){
    let el = $(id);
    if(document.activeElement !== el){
        el.value = val;
    }
}

function toggleMode(){
    if(modeNow === "AUTO"){
        fetch("/manual");
    }else{
        fetch("/auto");
    }
}

function toggleConveyor(){
    if(conveyorNow === "ON"){
        fetch("/off");
    }else{
        fetch("/on");
    }
}

function toggleBuzzer(){
    if(buzzerEnableNow === "AKTIF"){
        fetch("/buzzerOff");
    }else{
        fetch("/buzzerOn");
    }
}

function tareLoadcell(){
    fetch("/tare")
    .then(r => r.text())
    .then(t => alert("LOADCELL BERHASIL DI-TARE"));
}

function saveSetting(){
    let url = "/set?";
    url += "kapasitasMax=" +
encodeURIComponent($("#setKapasitasMax").value);
    url += "&kosong=" + encodeURIComponent($("#setKosong").value);
}

```

```

url += "&penuh=" + encodeURIComponent($("#setPenuh").value);
url += "&onLimit=" + encodeURIComponent($("#setOnLimit").value);
url += "&offLimit=" +
encodeURIComponent($("#setOffLimit").value);
url += "&beratOn=" + encodeURIComponent($("#setBeratOn").value);
url += "&beratOff=" +
encodeURIComponent($("#setBeratOff").value);
url += "&buzzerLevel=" +
encodeURIComponent($("#setBuzzerLevel").value);

fetch(url)
.then(r => r.text())
.then(t => alert("SETTING TERSIMPAN"));
}

function tampilJarak(v){
return v > 900 ? "NO" : v.toFixed(1);
}

function loadData(){
fetch("/data")
.then(r => r.json())
.then(d => {
conveyorNow = d.conveyor;
modeNow = d.mode;
buzzerEnableNow = d.buzzerEnable;

$("#j1").innerHTML = tampilJarak(d.j1);
$("#j2").innerHTML = tampilJarak(d.j2);
$("#j3").innerHTML = tampilJarak(d.j3);
$("#berat").innerHTML = d.berat.toFixed(0);

$("#conveyorText").innerHTML = d.conveyor;
$("#buzzerText").innerHTML = d.buzzerEnable + "<br>" +
d.buzzer;
$("#modeText").innerHTML = d.mode;
$("#statusSampah").innerHTML = d.status;

$("#liquidFill").style.height = d.level.toFixed(0) + "%";

$("#client").innerHTML = d.client;
$("#uptime").innerHTML = secToTime(d.up);

$("#conveyorBtn").innerHTML = "PENGGERUK " + d.conveyor;
$("#conveyorBtn").className = d.conveyor === "ON" ? "btnGreen"
: "btnRed";

$("#modeBtn").innerHTML = d.mode;
$("#modeBtn").className = d.mode === "AUTO" ? "btnGreen" :
"btnBlue";

$("#buzzerBtn").innerHTML = "BUZZER " + d.buzzerEnable;
$("#buzzerBtn").className = d.buzzerEnable === "AKTIF" ?
"btnYellow" : "btnRed";

if(d.mode === "AUTO"){
$("#conveyorBtn").classList.add("disabled");
}else{

```

```

        $(".conveyorBtn").classList.remove("disabled");
    }

    if(d.status === "KAPASITAS PENUH"){
        $(".binBox").classList.add("full");
        $(".statusSampah").style.color = "#ef4444";
        $(".capacityInfo").innerHTML = "KAPASITAS BOX SUDAH PENUH";
        $(".capacityInfo").style.color = "#fecaca";
        $(".desc").innerHTML = "BOX SAMPAH SUDAH MENCAPAI BATAS  
MAKSIMAL. PENGINEK SAMPAH DAPAT BERJALAN OTOMATIS DAN BUZZER AKAN  
MEMBERI ALARM.";
    }else if(d.status === "HAMPIR PENUH"){
        $(".binBox").classList.remove("full");
        $(".statusSampah").style.color = "#facc15";
        $(".capacityInfo").innerHTML = "KAPASITAS HAMPIR PENUH";
        $(".capacityInfo").style.color = "#fef08a";
        $(".desc").innerHTML = "ISI BOX SUDAH MENDEKATI BATAS. BUZZER  
MEMBERI PERINGATAN AGAR KONDISI MUDAH DIPANTAU.";
    }else if(d.status === "TERISI SEDANG"){
        $(".binBox").classList.remove("full");
        $(".statusSampah").style.color = "#38bdf8";
        $(".capacityInfo").innerHTML = "KAPASITAS SEDANG";
        $(".capacityInfo").style.color = "#bae6fd";
        $(".desc").innerHTML = "ISI BOX BERADA PADA KONDISI SEDANG  
DAN PENGINEK BELUM WAJIB MENYALA.";
    }else if(d.status === "MULAI TERISI"){
        $(".binBox").classList.remove("full");
        $(".statusSampah").style.color = "#22c55e";
        $(".capacityInfo").innerHTML = "BOX MULAI TERISI";
        $(".capacityInfo").style.color = "#bbf7d0";
        $(".desc").innerHTML = "SAMPAH MULAI MASUK KE DALAM BOX.  
SISTEM MEMBACA 3 SENSOR ULTRASONIK SECARA REALTIME.";
    }else{
        $(".binBox").classList.remove("full");
        $(".statusSampah").style.color = "#22c55e";
        $(".capacityInfo").innerHTML = "KAPASITAS MASIH AMAN";
        $(".capacityInfo").style.color = "#bbf7d0";
        $(".desc").innerHTML = "BOX MASIH KOSONG ATAU ISI SAMPAH  
MASIH RENDAH.";
    }
}

setInputValue("setKapasitasMax", d.kapasitasMax);
setInputValue("setKosong", d.kosong);
setInputValue("setPenuh", d.penuh);
setInputValue("setOnLimit", d.onLimit);
setInputValue("setOffLimit", d.offLimit);
setInputValue("setBeratOn", d.beratOn);
setInputValue("setBeratOff", d.beratOff);
setInputValue("setBuzzerLevel", d.buzzerLevel);
})
.catch(err => {
    console.log("DATA BELUM TERHUBUNG");
});
}

setInterval(loadData, 700);
loadData();
</script>

```

```

</body>
</html>
)rawliteral";

// =====
// ROUTE HALAMAN UTAMA
// =====
void root() {
    server.send_P(200, "text/html", PAGE);
}

// =====
// ROUTE DATA JSON
// =====
void sendData() {
    String json = "{";

    json += "\"conveyor\":\":";
    json += (conveyorState ? "ON" : "OFF");
    json += "\", ";

    json += "\"mode\":\":";
    json += (modeAuto ? "AUTO" : "MANUAL");
    json += "\", ";

    json += "\"buzzer\":\":";
    json += (buzzerOutputState ? "BUNYI" : "DIAM");
    json += "\", ";

    json += "\"buzzerEnable\":\":";
    json += (buzzerEnable ? "AKTIF" : "MATI");
    json += "\", ";

    json += "\"status\":\":";
    json += statusSampah();
    json += "\", ";

    json += "\"j1\":";
    json += String(jarak1, 1);
    json += ", ";

    json += "\"j2\":";
    json += String(jarak2, 1);
    json += ", ";

    json += "\"j3\":";
    json += String(jarak3, 1);
    json += ", ";

    json += "\"jarakRata\":";
    json += String(jarakRata, 1);
    json += ", ";

    json += "\"berat\":";
    json += String(berat, 0);
    json += ", ";

    json += "\"level\":";

```

```

json += String(levelKapasitas, 1);
json += ",";

json += "\"kapasitasMax\":";
json += String(kapasitasMaxGram, 0);
json += ",";

json += "\"kosong\":";
json += String(jarakKosongCM, 1);
json += ",";

json += "\"penuh\":";
json += String(jarakPenuhCM, 1);
json += ",";

json += "\"onLimit\":";
json += String(onLimit, 1);
json += ",";

json += "\"offLimit\":";
json += String(offLimit, 1);
json += ",";

json += "\"beratOn\":";
json += String(beratOn, 0);
json += ",";

json += "\"beratOff\":";
json += String(beratOff, 0);
json += ",";

json += "\"buzzerLevel\":";
json += String(buzzerLevelOn, 0);
json += ",";

json += "\"client\":";
json += String(WiFi.softAPgetStationNum());
json += ",";

json += "\"up\":";
json += String(millis() / 1000);

json += "}";

server.send(200, "application/json", json);
}

// =====
// ROUTE Pengeruk ON MANUAL
// =====
void onConveyor() {
  if (!modeAuto) {
    conveyorState = true;
    lastConveyor = millis();
    applyConveyor();
  }

  server.send(200, "text/plain", "OK");
}

```

```

}

// =====
// ROUTE Pengeruk OFF MANUAL
// =====
void offConveyor() {
    if (!modeAuto) {
        conveyorState = false;
        lastConveyor = millis();
        applyConveyor();
    }

    server.send(200, "text/plain", "OK");
}

// =====
// ROUTE MODE AUTO
// =====
void autoMode() {
    modeAuto = true;
    server.send(200, "text/plain", "OK");
}

// =====
// ROUTE MODE MANUAL
// =====
void manualMode() {
    modeAuto = false;
    server.send(200, "text/plain", "OK");
}

// =====
// ROUTE BUZZER ON / OFF
// =====
void buzzerOn() {
    buzzerEnable = true;
    server.send(200, "text/plain", "BUZZER ON");
}

void buzzerOff() {
    buzzerEnable = false;
    applyBuzzer(false);
    server.send(200, "text/plain", "BUZZER OFF");
}

// =====
// ROUTE TARE LOADCELL
// =====
void tareLoadcell() {
    if (scale.is_ready()) {
        scale.tare();
        berat = 0;
        levelBerat = 0;
    }

    server.send(200, "text/plain", "TARE OK");
}

```

```

// =====
// ROUTE SIMPAN SETTING DARI WEB
// =====
void saveSetting() {
    if (server.hasArg("kapasitasMax")) {
        kapasitasMaxGram = server.arg("kapasitasMax").toFloat();
    }

    if (server.hasArg("kosong")) {
        jarakKosongCM = server.arg("kosong").toFloat();
    }

    if (server.hasArg("penuh")) {
        jarakPenuhCM = server.arg("penuh").toFloat();
    }

    if (server.hasArg("onLimit")) {
        onLimit = server.arg("onLimit").toFloat();
    }

    if (server.hasArg("offLimit")) {
        offLimit = server.arg("offLimit").toFloat();
    }

    if (server.hasArg("beratOn")) {
        beratOn = server.arg("beratOn").toFloat();
    }

    if (server.hasArg("beratOff")) {
        beratOff = server.arg("beratOff").toFloat();
    }

    if (server.hasArg("buzzerLevel")) {
        buzzerLevelOn = server.arg("buzzerLevel").toFloat();
    }

    if (kapasitasMaxGram <= 0) {
        kapasitasMaxGram = 2000.0;
    }

    if (jarakKosongCM <= jarakPenuhCM) {
        jarakKosongCM = 45.0;
        jarakPenuhCM = 8.0;
    }

    if (offLimit <= onLimit) {
        offLimit = onLimit + 8.0;
    }

    if (beratOff >= beratOn) {
        beratOff = beratOn * 0.70;
    }

    if (buzzerLevelOn < 10) {
        buzzerLevelOn = 10;
    }

    if (buzzerLevelOn > 100) {

```

```

        buzzerLevelOn = 100;
    }

    server.send(200, "text/plain", "SETTING OK");
}

// =====
// ROUTE NOT FOUND
// =====
void notFound() {
    server.send(404, "text/plain", "HALAMAN TIDAK ADA");
}

// =====
// SETUP
// =====
void setup() {
    Serial.begin(115200);
    delay(500);

    Serial.println();

    Serial.println("=====
");
    Serial.println("SMART BIN MONITOR - 3 ULTRASONIK + BUZZER");

    Serial.println("=====
");

    pinMode(CONVEYOR_PIN, OUTPUT);
    pinMode(BUZZER_PIN, OUTPUT);

    pinMode(TRIG1, OUTPUT);
    pinMode(ECHO1, INPUT);

    pinMode(TRIG2, OUTPUT);
    pinMode(ECHO2, INPUT);

    pinMode(TRIG3, OUTPUT);
    pinMode(ECHO3, INPUT);

    conveyorState = false;
    applyConveyor();

    applyBuzzer(false);

    Serial.println("MEMULAI HX711...");
    scale.begin(HX_DT, HX_SCK);
    scale.set_scale(calibration_factor);

    if (scale.is_ready()) {
        scale.tare();
        Serial.println("HX711 READY");
    } else {
        Serial.println("HX711 BELUM TERBACA");
    }

    WiFi.mode(WIFI_AP);

```

```

WiFi.softAP(ssid, password);

Serial.println();
Serial.println("WIFI ACCESS POINT AKTIF");
Serial.print("SSID      : ");
Serial.println(ssid);
Serial.print("PASSWORD : ");
Serial.println(password);
Serial.print("IP        : ");
Serial.println(WiFi.softAPIP());

server.on("/", root);
server.on("/data", sendData);

server.on("/on", onConveyor);
server.on("/off", offConveyor);

server.on("/auto", autoMode);
server.on("/manual", manualMode);

server.on("/buzzerOn", buzzerOn);
server.on("/buzzerOff", buzzerOff);

server.on("/tare", tareLoadcell);
server.on("/set", saveSetting);

server.onNotFound(notFound);

server.begin();

Serial.println("WEB SERVER READY");
Serial.println("LOGIKA CONVEYOR : DIBALIK | ON=OFF | OFF=ON");
Serial.println("BUKA DI BROWSER: http://192.168.4.1");
Serial.println("PIN ULTRASONIK 3: TRIG GPIO25 | ECHO GPIO26");
Serial.println("PIN BUZZER      : GPIO2");

Serial.println("=====");
};

// =====
// LOOP
// =====
void loop() {
    updateSensor();
    updateConveyor();
    applyConveyor();
    updateBuzzer();

    server.handleClient();

    static unsigned long lastPrint = 0;

    if (millis() - lastPrint > 1500) {
        lastPrint = millis();

        Serial.println("-----");
    }
};

```

```

Serial.print("MODE                : ");
Serial.println(modeAuto ? "AUTO" : "MANUAL");

Serial.print("PENGGERUK          : ");
Serial.println(conveyorState ? "ON" : "OFF");

Serial.print("BUZZER ENABLE       : ");
Serial.println(buzzerEnable ? "AKTIF" : "MATI");

Serial.print("BUZZER OUTPUT        : ");
Serial.println(buzzerOutputState ? "BUNYI" : "DIAM");

Serial.print("JARAK 1                  : ");
if (jarakValid(jarak1)) {
    Serial.print(jarak1, 1);
    Serial.println(" CM");
} else {
    Serial.println("NO DATA");
}

Serial.print("JARAK 2                  : ");
if (jarakValid(jarak2)) {
    Serial.print(jarak2, 1);
    Serial.println(" CM");
} else {
    Serial.println("NO DATA");
}

Serial.print("JARAK 3                  : ");
if (jarakValid(jarak3)) {
    Serial.print(jarak3, 1);
    Serial.println(" CM");
} else {
    Serial.println("NO DATA");
}

Serial.print("JARAK RATA-RATA        : ");
if (jarakValid(jarakRata)) {
    Serial.print(jarakRata, 1);
    Serial.println(" CM");
} else {
    Serial.println("NO DATA");
}

Serial.print("BERAT                    : ");
Serial.print(berat, 0);
Serial.println(" GRAM");

Serial.print("KAPASITAS MAX          : ");
Serial.print(kapasitasMaxGram, 0);
Serial.println(" GRAM");

Serial.print("LEVEL JARAK            : ");
Serial.print(levelJarak, 1);
Serial.println(" %");

Serial.print("LEVEL BERAT            : ");

```

```
Serial.print(levelBerat, 1);  
Serial.println(" %");  
  
Serial.print("LEVEL INTERNAL    : ");  
Serial.print(levelKapasitas, 1);  
Serial.println(" %");  
  
Serial.print("STATUS              : ");  
Serial.println(statusSampah());  
  
Serial.print("CLIENT WIFI          : ");  
Serial.println(WiFi.softAPgetStationNum());  
}  
}
```

BIODATA PENULIS



1. Nama Lengkap : Ahmad Valentino
2. NRP : 0423030024
3. Program Studi : D3 Teknik Kelistrikan Kapal
4. Jenis Kelamin : Laki - Laki
5. Agama : Islam
6. Tempat, Tanggal Lahir : Indramayu, 14 Februari 2003
7. Alamat : Desa Anjatan, Kec Anjatan, Kab
Indramayu, Jawa Barat
8. No. Telp/HP : 082128826667
9. Email : ahmadvalentino@student.ppns.ac.id
10. Nama Orang Tua/Wali
11. Ayah : Edi Sopandi
12. Ibu : Iti Warniti
13. Alamat Orang Tua/Wali : Desa Anjatan, Kec Anjatan, Kab
Indramayu, Jawa Barat

Riwayat Pendidikan

- 2010 – 2016 : SDN 2 Anjatan
- 2016 – 2019 : SMP Islam Al-Ishlah Boarding School
- 2019 – 2022 : MAN 2 Cirebon
- 2023 – 2026 : Politeknik Perkapalan Negeri Surabaya