

SmartBin: Pemantau Sampah Berbasis IoT

Disusun Oleh:

Timotheus Ferdinand Tjondrojo (105222019)

Haekal Putra Alharis (105222028)



Laporan penelitian ini adalah sebagai bentuk
Ujian Akhir Semester (UAS)
untuk mata kuliah Mikrokontroller dan Internet of Things (IoT)

Juli 2025

Contents

<i>ABSTRAK</i>	2
<i>I. PENDAHULUAN</i>	2
1.1. Latar Belakang	2
II. Tujuan dan Manfaat.....	3
<i>III. METODE PENGEMBANGAN</i>	4
<i>IV. HASIL DAN ANALISIS</i>	10
<i>V. KESIMPULAN DAN SARAN</i>	11
<i>VI. REFERENSI</i>	12
<i>LAMPIRAN</i>	12

ABSTRAK

Permasalahan pengelolaan sampah yang masih dilakukan secara manual menjadi salah satu faktor rendahnya efisiensi dan higienitas dalam pengumpulan limbah di lingkungan publik. Petugas kebersihan harus memeriksa kondisi tempat sampah secara langsung, yang tidak hanya memakan waktu tetapi juga meningkatkan risiko kontak fisik dengan sampah. Untuk mengatasi permasalahan tersebut, dirancang sistem SmartBin, yaitu tempat sampah pintar berbasis Internet of Things (IoT) yang mampu mendeteksi keberadaan objek dan tingkat kepenruhan sampah secara otomatis. Sistem ini menggunakan dua sensor ultrasonik dan mikrokontroler ESP32. Sensor pertama berfungsi untuk membuka tutup tempat sampah secara otomatis saat mendeteksi objek di dekatnya, sementara sensor kedua memantau jarak antara tutup dan permukaan sampah. Data yang diperoleh dikirimkan melalui koneksi Wi-Fi ke platform website berbasis dashboard untuk ditampilkan dalam bentuk persentase kepenruhan, status tempat sampah, dan jumlah pembuangan. Hasil pengujian menunjukkan bahwa sistem mampu bekerja secara real-time, responsif, dan dapat mengurangi kontak langsung antara pengguna dengan tempat sampah. Dengan demikian, SmartBin mendukung pengelolaan sampah yang lebih efisien, aman, dan menjadi bagian dari implementasi sistem berbasis smart city.

Kata Kunci : SmartBin, IoT, ESP32, sensor ultrasonik, monitoring sampah, dashboard, otomatisasi, smart city.

I. PENDAHULUAN

1.1. Latar Belakang

Permasalahan pengelolaan sampah yang tidak efisien masih menjadi tantangan di berbagai lingkungan seperti perumahan, sekolah, dan fasilitas umum. Keterbatasan dalam memperoleh informasi secara real-time mengenai status kapasitas tempat sampah seringkali menyebabkan keterlambatan dalam proses pengosongan dan menurunnya kebersihan lingkungan. Situasi ini berpotensi menciptakan kondisi lingkungan yang tidak nyaman serta meningkatkan risiko penyebaran penyakit akibat tumpukan sampah yang tidak segera ditangani [1].

Petugas kebersihan umumnya dihadapkan pada sistem manual, di mana mereka harus memeriksa satu per satu kondisi tempat sampah secara langsung. Aktivitas ini selain memakan waktu, juga memperbesar kemungkinan kontak fisik dengan sampah yang terkontaminasi. Sebuah studi menunjukkan bahwa petugas kebersihan memiliki risiko kesehatan yang tinggi akibat paparan langsung dengan limbah padat, termasuk risiko infeksi saluran pernapasan dan penyakit kulit [2]. Oleh karena itu, diperlukan solusi yang mampu meminimalkan interaksi langsung ini dan mendukung lingkungan kerja yang lebih aman dan higienis.

Tidak hanya petugas kebersihan, masyarakat umum pun sering kali harus membuka tutup tempat sampah secara manual untuk membuang sampah, yang dapat meningkatkan risiko penularan penyakit melalui permukaan yang kotor. Dengan mempertimbangkan aspek kesehatan masyarakat, diperlukan sistem yang tidak hanya

dapat memantau kondisi tempat sampah secara otomatis, tetapi juga mampu mengurangi sentuhan langsung dalam proses pembuangan.

Sebagai solusi, dirancanglah SmartBin, sebuah sistem tempat sampah pintar berbasis Internet of Things (IoT) yang mengintegrasikan sensor ultrasonik dan mikrokontroler ESP32. Sistem ini bekerja dengan mendeteksi keberadaan objek (seperti tangan) untuk mengaktifkan pembukaan tutup secara otomatis tanpa perlu sentuhan fisik. Selain itu, sensor ultrasonik kedua memantau ketinggian sampah dan mengirimkan data status ke platform website melalui koneksi Wi-Fi. Ketika sampah telah mencapai ambang batas penuh, sistem akan mengirimkan notifikasi kepada petugas kebersihan secara real-time. Sistem ini secara signifikan meningkatkan efisiensi pengelolaan sampah dan memberikan perlindungan kesehatan yang lebih baik bagi pengguna maupun petugas [3]. Dengan memadukan aspek otomatisasi, pemantauan jarak jauh, serta kontrol kontak fisik, SmartBin menjadi bagian dari infrastruktur pendukung kota pintar (smart city) dalam mewujudkan lingkungan yang bersih, sehat, dan efisien.

II. Tujuan dan Manfaat

2.1. Tujuan:

1. Penerapan teknologi IoT dalam sistem pengelolaan sampah

Memungkinkan tempat sampah memonitor kondisi internal secara otomatis melalui sensor ultrasonik dan ESP32 sebagai pengontrol sistem.

2. Peningkatan efisiensi operasional petugas kebersihan

Menyediakan informasi real-time terkait status kepenruhan sampah, sehingga petugas dapat menentukan waktu pengosongan dengan tepat tanpa inspeksi manual.

3. Pengurangan interaksi langsung antara manusia dan tempat sampah

Meningkatkan higienitas dengan sistem pembukaan otomatis yang dikendalikan sensor, tanpa perlu menyentuh tutup tempat sampah secara fisik.

4. Penyediaan sistem monitoring berbasis website

Menyajikan tampilan informasi status tempat sampah yang mudah diakses oleh petugas kebersihan maupun pihak pengelola lingkungan.

5. Salah satu Langkah untuk Implementasi terhadap pengembangan kawasan berbasis smart city

Menjadi salah satu implementasi nyata dari sistem perkotaan pintar dalam sektor kebersihan dan manajemen lingkungan.

2.2. Manfaat:

1. Efisiensi dan kebersihan operasional

Mempermudah tugas petugas kebersihan dengan data real-time dan sistem untuk meminimalisir sentuhan, serta mencegah penularan penyakit akibat kontak langsung.

2. Pengelolaan sampah yang terstruktur

Mengurangi potensi sampah menumpuk karena keterlambatan pengosongan, serta menjaga kebersihan di area publik dan fasilitas umum.

3. Penghematan biaya dan tenaga

Mengurangi kebutuhan pengecekan manual dan pengangkutan sampah yang belum penuh, sehingga biaya operasional dapat dioptimalkan.

4. Fondasi sistem pengelolaan lingkungan pintar

Dapat diintegrasikan dengan sistem lain seperti pelaporan otomatis, sistem rute pengumpulan, atau analitik perilaku buang sampah, sebagai bagian dari ekosistem smart city.

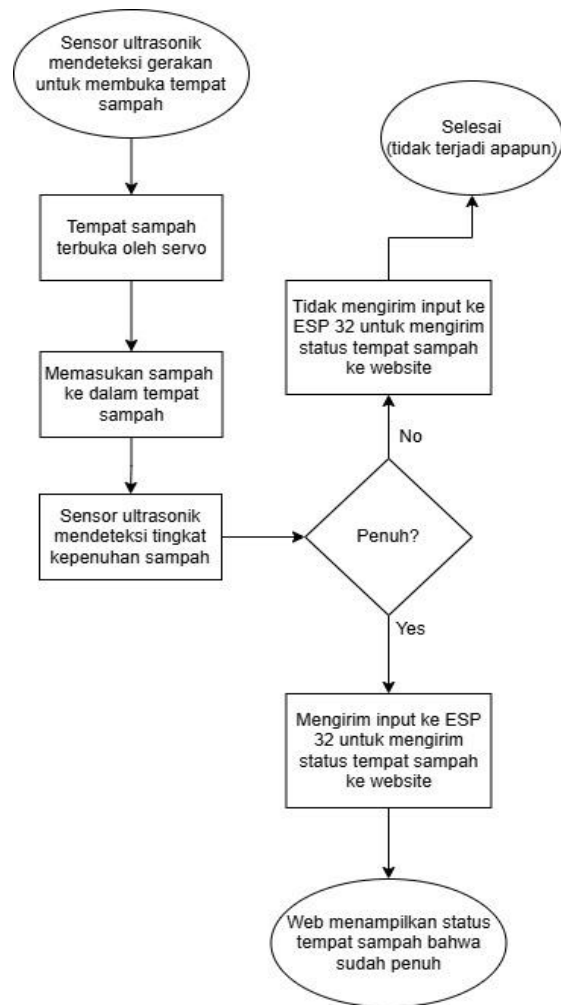
5. Mengurangi risiko paparan terhadap sampah yang terkontaminasi

Mengurangi kontak langsung antara pengguna maupun petugas kebersihan dengan sampah, sehingga menurunkan potensi penyebaran penyakit akibat kuman, bakteri, atau virus dari tempat sampah yang tidak higienis.

III. METODE PENGEMBANGAN

3.1. Blok Diagram Sistem

Sistem *SmartBin* bekerja secara otomatis dengan memanfaatkan sensor ultrasonik, mikrokontroler ESP32, dan koneksi Wi-Fi untuk mengirimkan notifikasi ke platform website. Agar memahami cara proses alur system kita dapat melihat dari blok diagram system pada gambar dibawah ini.



Gambar 3.1 Flowchart Blok Diagram



Gambar 3.2 Diagram Rangkaian

Proses alur kerja system:

1. Deteksi Gerakan Tangan oleh Sensor Ultrasonik

Proses dimulai ketika sensor ultrasonik mendeteksi adanya gerakan atau objek (seperti tangan pengguna) di dekat permukaan sensor yang terpasang di bagian luar tempat sampah. Deteksi ini menjadi pemicu pembukaan tutup tempat sampah.

2. Servo Motor Membuka Tutup Tempat Sampah

Setelah objek terdeteksi, ESP32 mengaktifkan servo motor yang akan membuka tutup tempat sampah secara otomatis. Proses ini memungkinkan pengguna membuang sampah tanpa menyentuh tutup secara langsung, sehingga lebih higienis.

3. Pengguna Memasukkan Sampah ke dalam Tempat Sampah

Setelah tutup terbuka, pengguna memasukkan sampah ke dalam tempat sampah. Tutup akan kembali tertutup dalam 10 detik seperti yang sudah ditetapkan dalam program, setelah tidak ada lagi deteksi gerakan.

4. Sensor Ultrasonik Mendeteksi Tingkat Kepenuhan Sampah

Setelah sampah dimasukkan, sensor ultrasonik kedua (yang diletakkan di bagian dalam tempat sampah) akan secara berkala membaca jarak antara tutup dan permukaan sampah. Jika jarak ini menyempit di bawah ambang batas yang ditentukan, artinya tempat sampah telah penuh.

5. Logika Keputusan – Apakah Tempat Sampah Sudah Penuh?

- Jika belum penuh: sistem tidak melakukan apa-apa. Tidak ada input yang dikirim ke ESP32.
- Jika penuh: sensor akan mengirimkan sinyal ke ESP32 sebagai tanda bahwa kapasitas sudah maksimal.

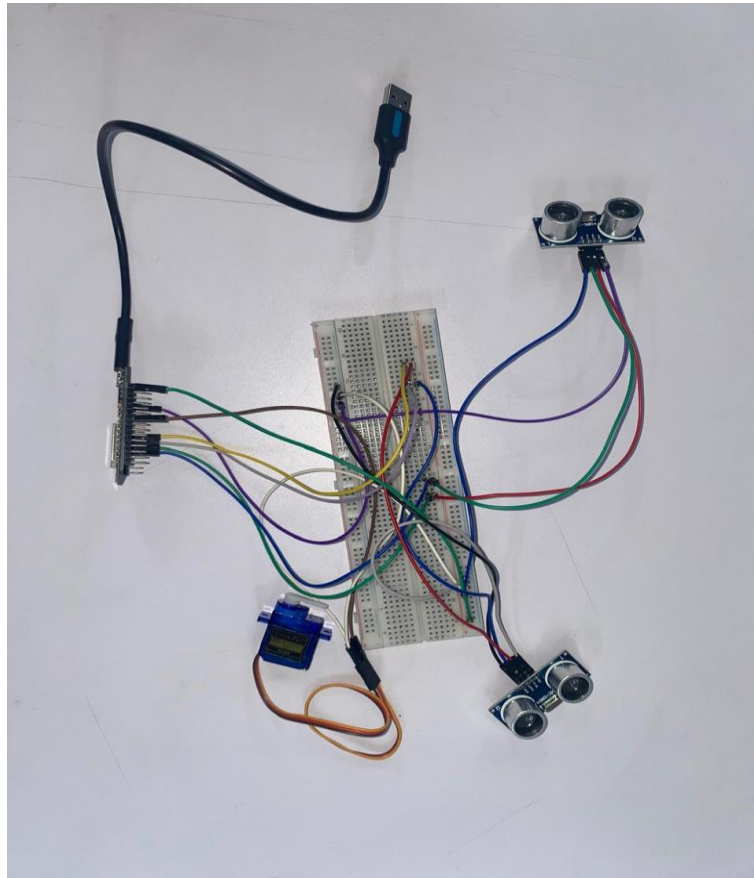
6. Pengiriman Notifikasi Melalui ESP32

ESP32 akan mengolah data tersebut dan mengirimkan input status “tempat sampah penuh” ke website melalui koneksi Wi-Fi.

7. Website Menampilkan Status Tempat Sampah

Website akan menerima informasi dari ESP32 dan menampilkan notifikasi atau peringatan bahwa **tempat sampah sudah penuh**, sehingga petugas kebersihan dapat segera melakukan pengosongan.

3.2. Skema Rangkaian Elektronik



Gambar 3.3 Skema rangkaian elektronik

A. Komponen – komponen dalam rangkaian elektronik:

1. ESP 32

Mikrokontroler utama yang mengatur semua proses logika dan komunikasi.

- **Port GND** ESP32 disambungkan ke jalur negatif pada breadboard sebagai ground bersama.
- **Port VIN** digunakan sebagai sumber tegangan (pengganti 5V) untuk menyuplai daya ke servo dan sensor ultrasonik melalui jalur positif breadboard.

2. Sensor Ultrasonik 1 (HC-SR04 – Deteksi Gerakan)

Berfungsi untuk mendeteksi keberadaan tangan atau objek di dekat tutup tempat sampah agar servo bisa membuka tutup otomatis, dengan port sebagai berikut:

- Trig: GPIO 21

- Echo: GPIO 19
- VCC: VIN (dari jalur positif breadboard)
- GND: GND (jalur negatif)

3. Sensor Ultrasonik 2 (HC-SR04 – Deteksi Kepenuhan Sampah)

Ditempatkan di dalam tempat sampah, sensor ini mendeteksi jarak antara tutup dan permukaan sampah untuk menentukan status penuh atau tidak, dengan port sebagai berikut:

- Trig: GPIO 5
- Echo: GPIO 18
- VCC: VIN
- GND: GND

4. Servo Motor (Penggerak Tutup Tempat Sampah)

Menggerakkan tutup untuk membuka dan menutup secara otomatis berdasarkan input dari sensor Gerak, dengan port sebagai berikut:

- GND: jalur negatif breadboard
- VCC: VIN (jalur positif breadboard)
- PWM (Signal): GPIO 26

5. Breadboard dan Power Supply

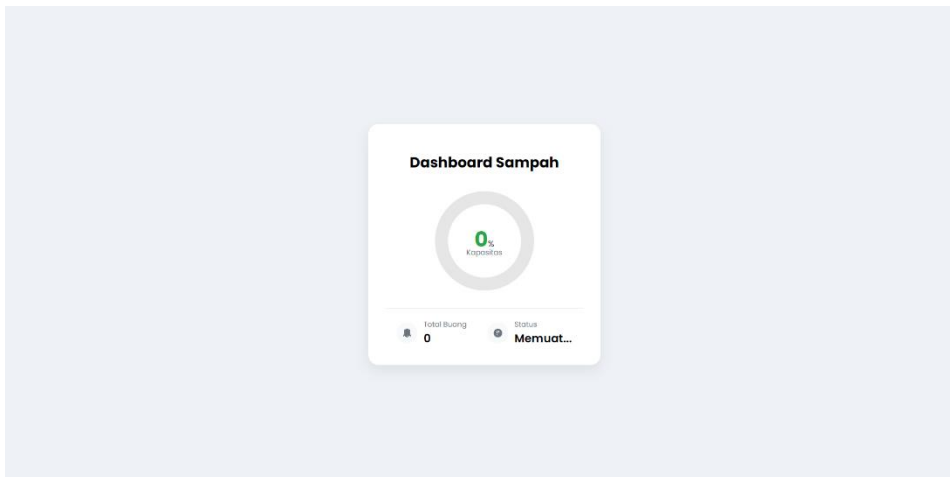
Semua komponen dihubungkan melalui breadboard untuk distribusi daya dan sinyal. Power supply disalurkan dari kabel USB ke ESP32, dan selanjutnya disebar ke seluruh komponen melalui pin VIN dan GND.

B. Fungsi Sistem Secara Keseluruhan:

1. Sensor ultrasonik 1: Mendeteksi objek di dekatnya, ESP32 akan menginstruksikan servo untuk membuka tutup tempat sampah.
2. Sensor ultrasonik 2 akan memantau ketinggian sampah di dalam wadah.
3. Jika sensor mendeteksi bahwa sampah sudah mencapai ambang batas penuh, ESP32 akan mengirimkan **notifikasi** ke website melalui koneksi Wi-Fi.
4. Sistem ini beroperasi **otomatis dan real-time**, serta mengurangi kontak fisik langsung dengan tempat sampah.

3.3. Tampilan Website

Tampilan antarmuka website ini berfungsi sebagai dashboard pemantauan SmartBin yang memberikan informasi real-time mengenai status tempat sampah. Website ini dirancang agar sederhana, responsif, dan mudah dipahami oleh petugas kebersihan maupun pengguna umum.



Gambar 3.4 Tampilan website

Penjelasan tampilan website:

1. Judul “Dashboard Sampah”

Merupakan bagian paling atas dari antarmuka sebagai identitas halaman utama monitoring sistem SmartBin.

2. Grafik Lingkaran Kapasitas

- Di tengah tampilan terdapat indikator berbentuk lingkaran yang menunjukkan persentase kapasitas tempat sampah.
- Pada gambar ini tertulis “0% Kapasitas”, artinya tempat sampah masih kosong.
- Nilai ini akan berubah secara dinamis berdasarkan pembacaan sensor ultrasonik yang mendeteksi tingkat ketinggian sampah.

3. Informasi Tambahan di Bagian Bawah

- Total Buang (0):
Menunjukkan jumlah aktivitas buka-tutup tutup tempat sampah, sebagai representasi jumlah orang yang telah membuang sampah pada hari itu.

IV. HASIL DAN ANALISIS

Rancangan sistem Smart TrashBin telah berhasil diimplementasikan dan diuji secara fungsional. Pengujian dilakukan untuk memverifikasi setiap komponen dan alur kerja sistem, mulai dari deteksi objek hingga pengiriman notifikasi ke dashboard website.

Sistem yang telah dirancang mampu beroperasi sesuai dengan tujuan yang ditetapkan. Hasil fungsional dari sistem ini adalah sebagai berikut:

1. **Otomatisasi Tutup Tempat Sampah:** Sensor ultrasonik 1 (deteksi gerakan) berhasil mendeteksi keberadaan tangan pada jarak kurang dari 20 cm. Setelah deteksi, mikrokontroler ESP32 secara sigap menginstruksikan motor servo untuk membuka tutup tempat sampah ke sudut 90 derajat. Tutup akan tetap terbuka selama 10 detik sebelum menutup kembali secara otomatis, memberikan waktu yang cukup bagi pengguna untuk membuang sampah.
2. **Pemantauan Kapasitas Real-time:** Sensor ultrasonik 2 (deteksi kepenuhan) secara kontinu mengukur jarak permukaan sampah dari tutup. Data jarak ini diolah oleh ESP32 untuk menghitung persentase kepenuhan berdasarkan tinggi total tempat sampah (10.7 cm).
3. **Notifikasi dan Dashboard Website:** Data persentase kepenuhan, status (Penuh/Tidak Penuh), dan jumlah pembuangan berhasil dikirimkan ke web server yang berjalan pada ESP32. Dashboard website yang diakses melalui alamat IP lokal mampu menampilkan informasi ini secara *real-time* dengan interval pembaruan setiap 2 detik. Ketika kapasitas mencapai 100% (jarak sampah ≤ 2.5 cm dari sensor), dashboard menampilkan status "Penuh", memicu notifikasi visual berupa *pop-up* peringatan, dan warna indikator berubah menjadi merah.
4. **Fitur Pengaman (Anti-Buka saat Penuh):** Sistem secara cerdas mencegah motor servo membuka tutup apabila status tempat sampah sudah penuh. Hal ini bertujuan untuk menghindari tumpahan sampah dan memberi sinyal visual kepada pengguna bahwa tempat sampah perlu segera dikosongkan.

Analisis Performa Hasil Pengujian

- **Responsivitas Sensor dan Aktuator:** Pengujian menunjukkan bahwa sensor ultrasonik HC-SR04 memiliki respons yang sangat cepat dan akurat untuk kedua fungsinya. Waktu tunda antara deteksi tangan dan aktivasi servo hampir tidak terasa. Demikian pula, motor servo bekerja dengan presisi sesuai sudut yang diprogramkan (0° untuk tutup, 90° untuk buka).
- **Akurasi Pemantauan Kapasitas:** Perhitungan persentase kepenuhan terbukti akurat selama permukaan sampah relatif datar. Variabel `TINGGI_TEMPAT_SAMPAH_CM` dan `AMBANG_BATAS_PENUH_CM` dalam kode program `main.py` menjadi parameter kunci yang menentukan akurasi ini. Dengan kalibrasi yang tepat pada tempat sampah yang sebenarnya, sistem ini dapat diandalkan.
- **Stabilitas Koneksi dan Web Server:** ESP32 mampu menjaga koneksi Wi-Fi yang stabil selama pengujian. Web server internal yang dibuat menggunakan *socket programming* dan dijalankan pada *thread* terpisah (`_thread`) terbukti efisien. Pendekatan ini memastikan bahwa proses pemantauan sensor dan aktuator di *main loop* tidak terganggu oleh permintaan (request) dari klien yang mengakses website, sehingga sistem tetap responsif.

- **Antarmuka Pengguna (UI/UX):** Tampilan dashboard dirancang dengan sederhana dan informatif. Penggunaan grafik lingkaran, kode warna (hijau, kuning, merah), dan notifikasi *pop-up* membuat informasi mudah dipahami oleh petugas kebersihan. Indikator "Total Buang" juga memberikan data tambahan yang berguna untuk menganalisis frekuensi penggunaan tempat sampah.

Secara keseluruhan, sistem Smart TrashBin menunjukkan performa yang andal dan sesuai dengan konsep yang diusulkan. Sistem ini berhasil membuktikan bahwa integrasi teknologi IoT dapat menciptakan solusi yang efektif untuk masalah pengelolaan sampah sehari-hari.

V. KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan dan analisis, dapat disimpulkan bahwa:

1. Proyek ini berhasil mengimplementasikan sistem tempat sampah pintar (Smart TrashBin) berbasis mikrokontroler ESP32 yang mengintegrasikan sensor ultrasonik dan motor servo secara fungsional.
2. Sistem mampu meningkatkan efisiensi pengelolaan sampah dengan menyediakan pemantauan kapasitas secara *real-time* melalui dashboard website, sehingga petugas kebersihan dapat bekerja berdasarkan data yang akurat.
3. Aspek higienitas berhasil ditingkatkan melalui fitur buka-tutup otomatis yang meminimalkan kontak fisik langsung antara pengguna dengan tempat sampah, sehingga mengurangi risiko penyebaran penyakit.
4. Smart TrashBin ini menjadi sebuah purwarupa yang valid dan aplikatif dari penerapan teknologi

Internet of Things (IoT) dalam mendukung infrastruktur kebersihan di lingkungan perumahan, perkantoran, atau sebagai fondasi menuju *smart city*. Bagi pembaca, laporan ini memberikan pengetahuan praktis mengenai cara merancang dan membangun sebuah perangkat IoT dari awal, mulai dari pemilihan komponen, perakitan sirkuit elektronik, pemrograman mikrokontroler (MicroPython), hingga pembuatan antarmuka monitoring sederhana.

Untuk pengembangan lebih lanjut, beberapa peningkatan dapat dilakukan untuk menyempurnakan performa dan fungsionalitas sistem:

1. **Manajemen Daya:** Mengganti sumber daya dari kabel USB ke sistem catu daya mandiri seperti baterai *rechargeable* (misalnya, Li-Po) yang dilengkapi dengan panel surya kecil. Ini akan membuat penempatan Smart TrashBin menjadi lebih fleksibel dan ramah lingkungan.
2. **Skalabilitas Sistem:** Untuk implementasi skala besar (misalnya, satu gedung atau satu kompleks), sebaiknya data dari setiap Smart TrashBin tidak di-hosting pada ESP32 masing-masing, melainkan dikirim ke sebuah server pusat (cloud). Hal ini memungkinkan pemantauan terpusat untuk puluhan atau ratusan tempat sampah dan dapat diintegrasikan dengan sistem optimasi rute pengangkutan sampah.
3. **Notifikasi Lanjutan:** Mengembangkan sistem notifikasi agar tidak hanya berupa *pop-up* di website, tetapi juga dapat mengirimkan peringatan melalui

email atau aplikasi pesan instan (seperti Telegram Bot) kepada petugas yang bersangkutan.

4. **Desain Fisik dan Durabilitas:** Merancang sebuah *casing* atau wadah khusus (misalnya dengan pencetakan 3D) untuk melindungi seluruh rangkaian elektronik dari potensi kerusakan akibat sampah atau cuaca, sehingga meningkatkan durabilitas dan estetika produk.

VI. REFERENSI

[1] S. N. Sirajuddin, S. Nasir, and Nurdin, "Occupational health risk analysis of waste pickers in the final disposal sites (TPA) of Makassar City," *Jurnal Ilmu Kesehatan Masyarakat*, vol. 10, no. 2, pp. 89–97, 2019. [Online]. Available: <https://core.ac.uk/download/pdf/235663842.pdf>

[2] A. A. Yudha and R. Azizah, "The Incidence of Skin Disorders in Garbage Officers in Indonesia and the Risk Factors Affecting It: A 2016–2021 Meta-Analysis Study," *Media Gizi Kemas*, vol. 12, no. 1, pp. 503–508, 2023, doi: 10.20473/mgk.v12i1.2023.503-508.

[3] P. R. Fadillah, S. A. Perdana, and A. Wibowo, "Analysis of occupational safety and health risks on waste transportation activities at TPS in Malang City," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1055, no. 1, pp. 012078, 2021, doi: 10.1088/1757-899X/1055/1/012078.

LAMPIRAN

Lampiran A: Kode Program

Lampiran A.1 Kode Main

```
from machine import Pin, time_pulse_us
from servo_lib import Servo
import utime
import network
import _thread
import socket
import json

WIFI_SSID = "hihihihi"
WIFI_PASSWORD = "123456789"

HTML_CONTENT = """
<!DOCTYPE html>
<html lang="id">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Dashboard Tempat Sampah Pintar</title>
  <link rel="preconnect" href="https://fonts.googleapis.com">

```

```

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link
href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700&display
=swap" rel="stylesheet">
<style>
:root {
  --danger-color: #dc3545;
  --success-color: #28a745;
  --warning-color: #ffc107;
}
body { font-family: 'Poppins', sans-serif; background-color: #eef2f7; display:
flex; justify-content: center; align-items: center; height: 100vh; margin: 0; }
.container { background-color: #ffffff; padding: 35px; border-radius: 20px; box-
shadow: 0 10px 25px rgba(0,0,0,0.08); text-align: center; width: 380px; }
h1 { font-size: 28px; margin-bottom: 30px; }
.chart-container { position: relative; width: 200px; height: 200px; margin: 0 auto
30px auto; }
.chart-text { position: absolute; top: 50%; left: 50%; transform: translate(-50%, -
50%); }
#percentage { font-size: 40px; font-weight: 700; }
.chart-label { font-size: 14px; color: #6c757d; margin-top: -15px; }
svg { width: 100%; height: 100%; transform: rotate(-90deg); }
.chart-background { fill: none; stroke: #e6e6e6; stroke-width: 15; }
.chart-progress { fill: none; stroke: var(--success-color); stroke-width: 15; stroke-
linecap: round; transition: all 0.5s ease-out; }
.stats-container { display: flex; justify-content: space-around; margin-top: 20px;
border-top: 1px solid #eee; padding-top: 20px; }
.stat-block { display: flex; align-items: center; gap: 12px; }
.stat-block .icon { width: 40px; height: 40px; background-color: #f8f9fa; border-
radius: 50%; display: flex; align-items: center; justify-content: center; }
.stat-block .icon svg { width: 20px; height: 20px; transform: none; fill: #6c757d;
}
.stat-info { text-align: left; }
.stat-label { font-size: 14px; color: #6c757d; }
.stat-value { font-size: 22px; font-weight: 600; }
.modal-overlay { position: fixed; top: 0; left: 0; width: 100%; height: 100%;
background-color: rgba(0, 0, 0, 0.6); display: flex; justify-content: center; align-items:
center; z-index: 1000; opacity: 0; visibility: hidden; transition: opacity 0.3s ease,
visibility 0.3s ease; }
.modal-overlay.visible { opacity: 1; visibility: visible; }
.modal-content { background-color: white; padding: 30px; border-radius: 15px;
width: 90%; max-width: 350px; text-align: center; transform: scale(0.9); transition:
transform 0.3s ease; }
.modal-overlay.visible .modal-content { transform: scale(1); }
.modal-content .icon { width: 60px; height: 60px; background-color: #f8f9fa; border-
radius: 50%; display: flex; align-items: center; justify-content: center; margin:
0 auto 20px auto; }
.modal-content .icon svg { width: 30px; height: 30px; fill: var(--danger-color); }
.modal-content h2 { font-size: 24px; margin: 0 0 10px 0; color: #333; }
.modal-content p { font-size: 16px; color: #6c757d; margin-bottom: 25px; }

```

```

    #modal-ok-btn { background-color: var(--danger-color); color: white; border:
none; padding: 12px 30px; border-radius: 8px; font-size: 16px; font-weight: 600;
cursor: pointer; width: 100%; transition: background-color 0.2s ease; }
    #modal-ok-btn:hover { background-color: #c82333; }
</style>
</head>
<body>
    <div class="container">
        <h1>Dashboard Sampah</h1>
        <div class="chart-container"><svg viewBox="0 0 120 120"><circle class="chart-
background" cx="60" cy="60" r="50"></circle><circle class="chart-progress"
cx="60" cy="60" r="50"></circle></svg><div class="chart-text"><span
id="percentage">0</span>%<div class="chart-label">Kapasitas</div></div>
        <div class="stats-container"><div class="stat-block"><div class="icon"><svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M15
2H9v2h6V2zM9 2H9v2zm8-2h-2v-2h-2v2h-4v-2H9v2H7v-2H5v2H3v-
4h18v4h-2v-2h-2v2zM5 4h14v12H5V4z"/></svg></div><div class="stat-info"><div
class="stat-label">Total Buang</div><div class="stat-value"
id="counter">0</div></div></div><div class="stat-block"><div class="icon"><svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24"><path d="M12 2C6.48 2
2 6.48 2 12s4.48 10 10 10-4.48 10-10S17.52 2 12 2zm-1 15H9v-2h2v2zm0-
4H9V7h2v6zm4-2h-2V7h2v6z"/></svg></div><div class="stat-info"><div
class="stat-label">Status</div><div class="stat-value" id="status-
text">Memuat...</div></div></div>
        </div>
        <div class="modal-overlay" id="modal-overlay"><div class="modal-content"><div
class="icon"><svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24
24"><path d="M1 21h22L12 2 1 21zm12-3h-2v-2h2v2zm0-4h-2v-
4h2v4z"/></svg></div><h2>Peringatan</h2><p>Tempat sampah sudah penuh.
Mohon segera dikosongkan!</p><button id="modal-ok-
btn">OK</button></div></div>
    </script>
    document.addEventListener('DOMContentLoaded', function () {
        const progressCircle = document.querySelector('.chart-progress'); const radius
= progressCircle.r.baseVal.value; const circumference = 2 * Math.PI * radius;
        progressCircle.style.strokeDasharray = `${circumference} ${circumference}`;
        progressCircle.style.strokeDashoffset = circumference;
        function setProgress(percent) { const offset = circumference - (percent / 100) *
circumference; progressCircle.style.strokeDashoffset = offset; const percentageText =
document.getElementById('percentage'); percentageText.innerText =
Math.round(percent); if (percent > 85) { progressCircle.style.stroke = 'var(--danger-
color)'; percentageText.style.color = 'var(--danger-color)'; } else if (percent > 60) {
progressCircle.style.stroke = 'var(--warning-color)'; percentageText.style.color = 'var(-
-warning-color)'; } else { progressCircle.style.stroke = 'var(--success-color)';
percentageText.style.color = 'var(--success-color)'; } }
        let isPopupShown = false; const modalOverlay =
document.getElementById('modal-overlay'); const modalOkBtn =
document.getElementById('modal-ok-btn');
        function showPopup() { modalOverlay.classList.add('visible'); }
        function hidePopup() { modalOverlay.classList.remove('visible'); }

```

```

        modalOkBtn.addEventListener('click', hidePopup);
        function fetchData() {
            fetch('/data').then(response => response.json())
                .then(data => {
                    document.getElementById('counter').innerText = data.counter; const
statusElement = document.getElementById('status-text'); const statusFromServer =
data.status_text;
                    if (statusFromServer === 'Penuh' && !isPopupShown) { showPopup();
isPopupShown = true; } else if (statusFromServer !== 'Penuh') { isPopupShown =
false; }
                    if (statusFromServer === 'Penuh') { statusElement.innerText = 'Penuh';
statusElement.style.color = 'var(--danger-color)'; } else { statusElement.innerText =
'Tidak Penuh'; statusElement.style.color = 'var(--success-color)'; }
                    setProgress(data.percentage);
                }).catch(error => console.error('Error fetching data:', error));
            }
        setProgress(0); setInterval(fetchData, 2000); fetchData();
    });
</script>
</body>
</html>
""""

```

```

TRIG_PIN_1, ECHO_PIN_1 = 21, 19
TRIG_PIN_2, ECHO_PIN_2 = 5, 18
SERVO_PIN = 26
trig1 = Pin(TRIG_PIN_1, Pin.OUT)
echo1 = Pin(ECHO_PIN_1, Pin.IN)
trig2 = Pin(TRIG_PIN_2, Pin.OUT)
echo2 = Pin(ECHO_PIN_2, Pin.IN)
servo = Servo(SERVO_PIN)
SUDUT_BUKA, SUDUT_TUTUP = 90, 0
TINGGI_TEMPAT_SAMPAH_CM, AMBANG_BATAS_PENUH_CM = 10.7, 2.5

jumlah_pembuangan = 0
persentase_kepenuhan = 0
status_kepenuhan_text = "Inisialisasi..."
status_string_web = "Memuat..."

def connect_wifi():
    wlan = network.WLAN(network.STA_IF)
    wlan.active(True)
    if not wlan.isconnected():
        print(f'Menghubungkan ke jaringan {WIFI_SSID}...')
        wlan.connect(WIFI_SSID, WIFI_PASSWORD)
        while not wlan.isconnected():
            utime.sleep(1)
    ip_address = wlan.ifconfig()[0]
    print(f'Terhubung! Alamat IP: http://{ip_address}')
    return ip_address

```



```

def measure_distance(trig_pin, echo_pin):
    trig_pin.value(0)
    utime.sleep_us(2)
    trig_pin.value(1)
    utime.sleep_us(10)
    trig_pin.value(0)
    try:
        return (time_pulse_us(echo_pin, 1, 30000) / 2) / 29.1
    except OSError:
        return None

def cek_status_kepenuhan():
    global persentase_kepenuhan, status_kepenuhan_text, status_string_web
    jarak_sampah = measure_distance(trig2, echo2)
    if jarak_sampah is None:
        return False, "Status: Error Sensor"

    if jarak_sampah <= AMBANG_BATAS_PENUH_CM:
        is_full = True
        persentase_kepenuhan = 100
        status_string_web = "Penuh"
    else:
        is_full = False
        persentase = ((TINGGI_TEMPAT_SAMPAH_CM - jarak_sampah) /
TINGGI_TEMPAT_SAMPAH_CM) * 100
        persentase_kepenuhan = round(max(0, min(100, persentase)))
        status_string_web = "Tidak Penuh"

    status_kepenuhan_text = f"Status: Terisi {persentase_kepenuhan}% (Jarak:
{jarak_sampah:.1f} cm)"
    if is_full:
        status_kepenuhan_text = f"Status: PENUH (Jarak: {jarak_sampah:.1f} cm)"
    return is_full, status_kepenuhan_text

def run_web_server():
    addr = socket.getaddrinfo('0.0.0.0', 80)[0][-1]
    s = socket.socket()
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind(addr)
    s.listen(5)
    print('Web server berjalan di port 80')

    while True:
        try:
            conn, addr = s.accept()
            request = conn.recv(1024)
            request = str(request)

            if 'GET /data' in request:

```

```

        data = {'counter': jumlah_pembuangan, 'percentage': persentase_kepenuhan,
'status_text': status_string_web}
        response_body = json.dumps(data)
        conn.send('HTTP/1.0 200 OK\r\n')
        conn.send('Content-Type: application/json\r\n')
        conn.send('Connection: close\r\n\r\n')
        conn.sendall(response_body)
    elif 'GET / ' in request:
        conn.send('HTTP/1.0 200 OK\r\n')
        conn.send('Content-Type: text/html\r\n')
        conn.send('Connection: close\r\n\r\n')
        conn.sendall(HTML_CONTENT)
    else:
        conn.send('HTTP/1.0 404 Not Found\r\n')
        conn.send('Connection: close\r\n\r\n')

    conn.close()
except OSError as e:
    conn.close()

def main_program():
    global jumlah_pembuangan
    servo.move_to_angle(SUDUT_TUTUP)
    print("Program utama (sensor & servo) berjalan.")

    while True:
        is_full, status_info = cek_status_kepenuhan()
        print(status_info, end=" | ")

        jarak_tangan = measure_distance(trig1, echo1)

        if jarak_tangan is not None:
            print(f"Jarak Tangan: {jarak_tangan:.1f} cm")

            if jarak_tangan < 20 and not is_full:
                print("Objek terdeteksi! Membuka servo...")
                servo.move_to_angle(SUDUT_BUKA)
                utime.sleep(10)
                servo.move_to_angle(SUDUT_TUTUP)
                print("Servo tertutup.")
                jumlah_pembuangan += 1

                print("Menunggu objek menjauh...")
                while True:
                    dist_check = measure_distance(trig1, echo1)
                    if dist_check is None or dist_check > 25:
                        break
                    utime.sleep(0.2)

            elif jarak_tangan < 20 and is_full:

```

```

        print("GAGAL MEMBUKA: Tempat sampah sudah penuh!")
        utime.sleep(2)
    else:
        print("Jarak Tangan: -")

    utime.sleep(0.5)

try:
    connect_wifi()
    _thread.start_new_thread(run_web_server, ())
    main_program()
except KeyboardInterrupt:
    servo.deinit()
    print("\nProgram dihentikan oleh pengguna.")

```

Lampiran A.2 Kode Library Servo

```

"""
This MicroPython library is designed for any hardware platform that supports
MicroPython such as Raspberry Pi Pico, ESP32, Micro:bit... to make it easy to use
with servo motor

It is created by DIYables to work with DIYables products, but also work with products
from other brands. Please consider purchasing products from [DIYables Store on
Amazon](https://amazon.com/diyables) from to support our work.

Product Link:
- Servo Motor: https://diyables.io/products/servo-motor-sg90-180-degree
- Sensor Kit: https://diyables.io/products/sensor-kit

Copyright (c) 2024, DIYables.io. All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

- Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright
  notice, this list of conditions and the following disclaimer in the
  documentation and/or other materials provided with the distribution.

- Neither the name of the DIYables.io nor the names of its
  contributors may be used to endorse or promote products derived from
  this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY DIYABLES.IO "AS IS" AND ANY
EXPRESS OR

```

IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL DIYABLES.IO BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

"""

```
from machine import Pin, PWM
import utime
```

```
class Servo:
```

```
    def __init__(self, pin_number, min_pulse=600, max_pulse=2450, frequency=50):
```

```
        """ Initialize a servo object on a specified pin with customizable pulse widths and frequency. """
```

```
        self.servo_pin = PWM(Pin(pin_number)) # Set up PWM on the pin
```

```
        self.servo_pin.freq(frequency)      # Set frequency, default is 50Hz
```

```
        self.min_pulse = min_pulse
```

```
        self.max_pulse = max_pulse
```

```
        self.cycle = 1 / frequency * 1000000    # PWM cycle in us
```

```
    def move_to_angle(self, angle):
```

```
        """ Move the servo to a specific angle. """
```

```
        # Calculate the pulse width corresponding to the angle
```

```
        pulse_width = self.min_pulse + (self.max_pulse - self.min_pulse) * angle / 180
```

```
        # Set the PWM duty cycle
```

```
        self.servo_pin.duty_u16(int(pulse_width * 65536 / self.cycle)) # Convert to duty for Pico's 16-bit resolution
```

```
    def deinit(self):
```

```
        """ Disable the PWM signal. """
```

```
        self.servo_pin.deinit()
```

```
    def __del__(self):
```

```
        """ Clean up the servo object when it's being destroyed. """
```

```
        self.deinit()
```

--

Lampiran B : Link Video Demo

[VideoDemo.mp4 - Google Drive](#)