

Perancangan Sistem Backend Real-Time Berbasis FastAPI dan Firebase Untuk Klasifikasi Sampah Bahan Berbahaya dan Beracun Serta Pengambilan Lokasi Dropbox Pada Aplikasi Mobile TOXMAP

1st Gading Aurelia Nabila Lubis
Faculty of Electrical Engineering
Telkom University
Bandung, Indonesia
gadingaurelia@student.telkomuniversit
y.ac.id

2nd Rita Purnamasari
Faculty of Electrical Engineering
Telkom University
Bandung, Indonesia
ritapurnamasari@telkomuniversity.ac.i
d

3rd Khaerudin Saleh
Faculty of Electrical Engineering
Telkom University
Bandung, Indonesia
khaerudin@telkomuniversity.ac.id

Abstrak — Sampah rumah tangga yang tergolong Bahan Berbahaya dan Beracun (B3) dapat menimbulkan risiko serius terhadap lingkungan dan kesehatan jika tidak dikelola dengan baik. Untuk mengatasi hal ini, dikembangkan TOXMAP sebagai sistem berbasis mobile yang menggabungkan klasifikasi citra secara real-time dengan panduan pembuangan sampah berbasis lokasi. Penelitian ini membahas pengembangan backend TOXMAP yang menggunakan server FastAPI untuk memproses input gambar dan mengklasifikasikan jenis sampah dengan model Support Vector Machine (SVM) yang telah dilatih sebelumnya. Layanan Firebase digunakan untuk autentikasi pengguna, penyimpanan gambar, dan pengambilan data lokasi dropbox. Untuk mendukung kinerja backend ini, frontend TOXMAP dibangun menggunakan Flutter sehingga dapat berjalan lintas platform dan mendukung akses kamera serta pengiriman gambar secara real-time. Hasil pengujian integrasi dan beban menunjukkan bahwa sistem dapat merespons dalam waktu kurang dari satu detik dengan akurasi klasifikasi dan responsivitas pengguna yang baik. Arsitektur ini memberikan infrastruktur yang ringan, skalabel, dan mudah diakses untuk meningkatkan kesadaran publik dan mendukung pengelolaan sampah B3 rumah tangga secara bertanggung jawab.

Kata kunci— backend, sistem real-time, klasifikasi gambar, fastapi, firebase, sampah B3

Abstract — Hazardous household waste (B3) poses serious risks to the environment and public health when improperly managed. To address this, TOXMAP was developed as a mobile-based system that combines real-time image classification with location-based waste disposal guidance. This study presents the backend development of TOXMAP, which integrates a FastAPI server to process image inputs and classify waste types using a pre-trained Support Vector Machine (SVM) model. Firebase services support user authentication, image storage, and dropbox location retrieval. To complement this backend, the TOXMAP frontend was built using Flutter, enabling cross-

platform mobile deployment with support for camera access and real-time image submission. Integration and load testing show that the system can respond in under one second per request while maintaining classification accuracy and user responsiveness. This architecture offers a lightweight, scalable, and user-accessible infrastructure to promote public awareness and responsible household B3 waste management.

Keywords— backend, real-time system, image classification, FastAPI, Firebase, B3 waste

I. PENDAHULUAN

Sampah Bahan Berbahaya dan Beracun (B3) merupakan jenis limbah yang memiliki potensi mencemari lingkungan serta menimbulkan risiko kesehatan serius bagi manusia maupun makhluk hidup lainnya. Limbah ini meliputi berbagai produk rumah tangga seperti baterai bekas, lampu neon, kaleng aerosol, obat kedaluwarsa, dan produk pembersih kimia. Meskipun keberadaannya sangat dekat dengan aktivitas masyarakat sehari-hari, pengetahuan masyarakat mengenai identifikasi, klasifikasi, dan pengelolaan limbah B3 masih tergolong rendah. Akibatnya, limbah B3 sering tercampur dengan sampah domestik biasa, sehingga memperbesar risiko pencemaran lingkungan dan menyulitkan proses penanganan limbah di hilir[1].

Permasalahan ini tidak hanya berkaitan dengan aspek lingkungan, tetapi juga menyangkut kurangnya media bantu yang praktis dan mudah diakses oleh masyarakat umum. Informasi terkait jenis-jenis limbah B3, karakteristiknya, serta prosedur penanganannya biasanya disajikan melalui media cetak atau kampanye edukasi satu arah, yang dalam praktiknya tidak cukup menjawab kebutuhan masyarakat modern yang mengandalkan teknologi digital dalam aktivitas sehari-hari[2]. Oleh karena itu, diperlukan pendekatan yang lebih interaktif, responsif, dan terintegrasi untuk mendukung

kesadaran dan perilaku pengelolaan limbah yang lebih bertanggung jawab.

Untuk menjawab tantangan tersebut, dikembangkanlah TOXMAP, sebuah aplikasi mobile berbasis Flutter yang mengintegrasikan teknologi machine learning dan pemetaan digital dalam satu sistem yang komprehensif. Aplikasi ini tidak hanya berfungsi sebagai media edukasi, tetapi juga memungkinkan pengguna melakukan klasifikasi jenis sampah secara langsung melalui pemindaian gambar menggunakan kamera ponsel. Proses klasifikasi dilakukan secara real-time oleh sistem backend yang dibangun menggunakan FastAPI, sebuah framework backend modern berbasis Python yang cepat, ringan, dan mudah diintegrasikan dengan model machine learning. Dalam hal ini, model yang digunakan adalah Support Vector Machine (SVM) yang telah dilatih sebelumnya menggunakan data gambar sampah B3.

Sistem backend TOXMAP tidak hanya bertanggung jawab terhadap klasifikasi sampah, tetapi juga menangani autentikasi pengguna, penyimpanan metadata klasifikasi, dan pengambilan data lokasi dropbox dari basis data Cloud Firestore. Peran backend dalam sistem ini menjadi krusial karena seluruh proses pengambilan Keputusan mulai dari klasifikasi hingga pencarian lokasi dropbox dilakukan melalui endpoint API yang didefinisikan dalam server FastAPI.

Penelitian ini berfokus pada perancangan dan implementasi sistem backend TOXMAP sebagai solusi teknologi untuk membantu identifikasi dan pengolahan limbah B3 secara otomatis. Dengan adanya sistem yang ringan, terintegrasi, dan dapat diakses melalui perangkat mobile, TOXMAP diharapkan mampu menjadi sarana yang efektif dalam meningkatkan kesadaran lingkungan masyarakat serta mendorong praktik pengelolaan limbah yang lebih terstruktur dan berkelanjutan.

II. KAJIAN TEORI

A. Sistem Terdistribusi dan Arsitektur Client-Server

Sistem terdistribusi terdiri dari beberapa komponen yang berjalan secara paralel pada perangkat atau lokasi yang berbeda, namun tetap terintegrasi melalui jaringan komunikasi. Pendekatan yang umum digunakan adalah arsitektur *client-server*, di mana klien mengirim permintaan layanan dan server memproses permintaan tersebut lalu mengembalikan hasilnya[3]. Arsitektur ini memungkinkan pemisahan yang jelas antara antarmuka pengguna (*frontend*) dan logika pemrosesan (*backend*), sehingga memberikan fleksibilitas dalam pengembangan aplikasi.

B. FastAPI

FastAPI merupakan kerangka kerja web berbasis Python yang digunakan untuk membangun REST API[4]. Framework ini mendukung operasi asinkron, dokumentasi otomatis melalui OpenAPI (Swagger), serta integrasi yang mudah dengan model *machine learning* berbasis Python seperti file *.pkl*. FastAPI dikenal karena performa tinggi, kemudahan pengembangan, dan kompatibilitasnya dengan infrastruktur cloud modern[5].

C. Firebase: Authentication, Firestore, and Storage

Firebase merupakan platform *Backend-as-a-Service* (BaaS) yang menyediakan layanan penting seperti autentikasi pengguna, penyimpanan data, dan manajemen file berbasis cloud. Firebase Authentication digunakan untuk mengelola login dan registrasi pengguna secara aman. Cloud Firestore menyimpan metadata seperti lokasi dropbox dan riwayat klasifikasi, sedangkan Firebase Storage mengelola file gambar yang dikirim dari aplikasi mobile sebelum diproses di backend[6]. Kombinasi layanan ini memastikan sinkronisasi yang andal dan dukungan lintas platform yang efisien.

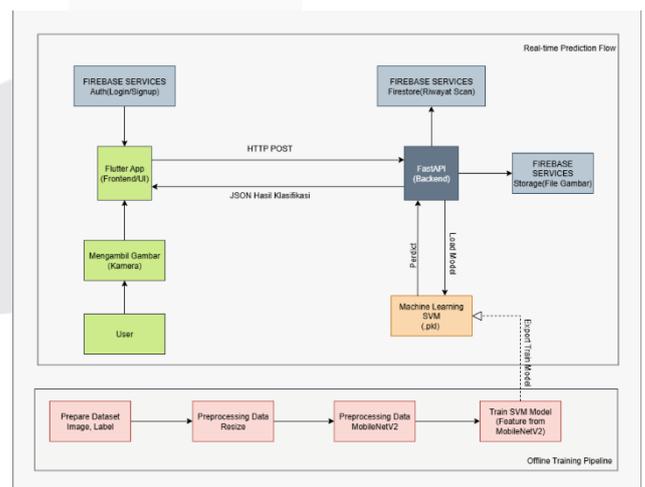
D. Endpoint API

Endpoint API adalah titik akses dalam sistem backend yang memungkinkan frontend atau klien berkomunikasi dengan server melalui protokol HTTP[7]. Desain endpoint yang baik mencerminkan fungsi spesifik dalam sistem dan memungkinkan pertukaran data secara efisien antara antarmuka pengguna dan logika backend. Dalam arsitektur RESTful, metode POST digunakan untuk mengirim data (seperti file atau input pengguna), sementara metode GET digunakan untuk mengambil data dari server[8]. Perancangan endpoint ini memegang peran penting untuk memastikan komunikasi real-time dan kinerja sistem yang optimal.

III. METODE

A. Arsitektur Sistem

Arsitektur sistem backend TOXMAP dirancang untuk mengintegrasikan antara frontend Flutter, layanan Firebase, dan model machine learning berbasis SVM (.pkl) yang dijalankan menggunakan FastAPI. Sistem ini memungkinkan proses klasifikasi sampah B3 secara real-time dari aplikasi mobile dan pengelolaan data secara efisien melalui infrastruktur cloud.



Gambar 1 Diagram Arsitektur Sistem

Pada Gambar 1 ditunjukkan arsitektur sistem secara keseluruhan, di mana data gambar dikirim dari aplikasi Flutter ke Firebase Storage. Backend FastAPI kemudian mengakses gambar tersebut melalui URL publik dan memrosesnya menggunakan model SVM yang telah dilatih sebelumnya. Setelah proses klasifikasi selesai, hasil prediksi

dan metadata terkait dikirim kembali ke frontend dan disimpan di Firebase Firestore.

B. Firebase dan Struktur Database

Sistem backend TOXMAP memanfaatkan tiga layanan utama dari Firebase, yaitu Firebase Authentication, Firebase Cloud Firestore, dan Firebase Storage. Firebase Authentication digunakan untuk proses autentikasi pengguna dalam bentuk login dan pendaftaran akun. Firebase Storage berfungsi untuk menyimpan file gambar yang dikirim dari aplikasi, sedangkan Firestore digunakan untuk menyimpan metadata klasifikasi, data pengguna, dan titik lokasi dropbox.

Struktur data pada Firestore dirancang ke dalam tiga koleksi utama:

1. **users:** Menyimpan data profil pengguna seperti ID, nama, dan email.
2. **scan_history:** Menyimpan hasil klasifikasi gambar yang diunggah oleh pengguna beserta waktu, label hasil, dan warna dropbox.
3. **dropbox_location:** Menyimpan titik lokasi dropbox terdekat beserta informasi koordinat GPS dan jenis sampah yang diterima.

Struktur masing-masing koleksi ditunjukkan pada *Tabel 1* hingga *Tabel 3* berikut ini.

Tabel 1 Database Koleksi: Users

Field	Tipe Data	Deskripsi
email	string	Alamat email pengguna
username	string	Nama pengguna atau nama tampilan
created_at	timestamp	Waktu saat akun dibuat

Tabel 2 Database Koleksi: scan_history

Field	Tipe Data	Deskripsi
user_id	String	ID atau username dari pengguna yang melakukan scan
image_url	String	URL gambar yang diunggah ke Firebase Storage
result	String	Hasil klasifikasi (misal: "Toxic", "Non_Toxic")
dropbox_color	String	Warna dropbox yang direkomendasikan (jika B3)
timestamp	Timestamp	Waktu klasifikasi dilakukan

Tabel 3 Database Koleksi: dropbox_location

Field	Tipe Data	Deskripsi
name	String	Nama titik lokasi (misal: nama kantor atau tempat)
address	String	Alamat lengkap lokasi
lat	Float	Koordinat latitude lokasi
lng	Float	Koordinat longitude lokasi

C. Integrasi Model Machine Learning (SVM)

Model klasifikasi pada backend TOXMAP dibangun menggunakan algoritma Support Vector Machine (SVM) yang telah dilatih sebelumnya menggunakan citra sampah berlabel. Model ini disimpan dalam format (.pkl) dan diload secara langsung oleh backend FastAPI saat proses prediksi berlangsung.

Proses integrasi dilakukan dengan tahapan sebagai berikut: gambar yang telah diunggah ke Firebase Storage diakses oleh backend menggunakan URL publik, kemudian dilakukan preprocessing (seperti resize dan flattening) agar sesuai dengan format input model. Model SVM kemudian melakukan prediksi dan mengembalikan label klasifikasi (misalnya: "B3 - Merah" atau "Non-B3") dalam bentuk respons JSON.

Hasil prediksi ini digunakan oleh sistem untuk menampilkan informasi kepada pengguna serta menyimpan data klasifikasi ke dalam Firestore. Pendekatan ini memungkinkan klasifikasi gambar dilakukan secara cepat tanpa perlu pelatihan ulang model.

IV. HASIL DAN PEMBAHASAN

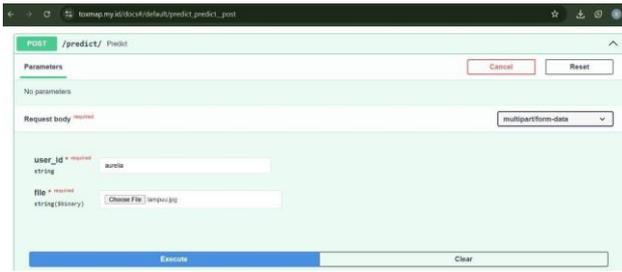
Pengembangan backend dalam sistem TOXMAP tidak hanya berfokus pada integrasi dengan model klasifikasi, tetapi juga menekankan aspek komunikasi data, efisiensi proses, dan kemudahan integrasi lintas platform. Setiap komponen dirancang agar mampu bekerja secara real-time dan mendukung kebutuhan pengguna dari sisi aplikasi mobile. Pada bagian ini, dibahas bagaimana masing-masing komponen backend direalisasikan secara fungsional, mulai dari desain endpoint API, integrasi Firebase, hingga pengelolaan model machine learning. Pembahasan juga mencakup bagaimana sistem merespons input pengguna dan menampilkan hasil secara cepat dan akurat.

A. Implementasi Endpoint API

Endpoint dirancang menggunakan framework FastAPI yang mendukung pemrosesan data secara cepat dan efisien. Endpoint utama yang digunakan dalam proses klasifikasi adalah /predict, yang bertugas menerima input berupa gambar dan ID pengguna dari aplikasi Flutter, kemudian memprosesnya menggunakan model machine learning SVM yang telah dilatih sebelumnya.

Endpoint /predict menerima dua parameter utama, yaitu user_id dan file. Backend akan mengekstrak gambar dan memprosesnya melalui model SVM, kemudian

mengembalikan hasil klasifikasi ke frontend dalam format JSON.



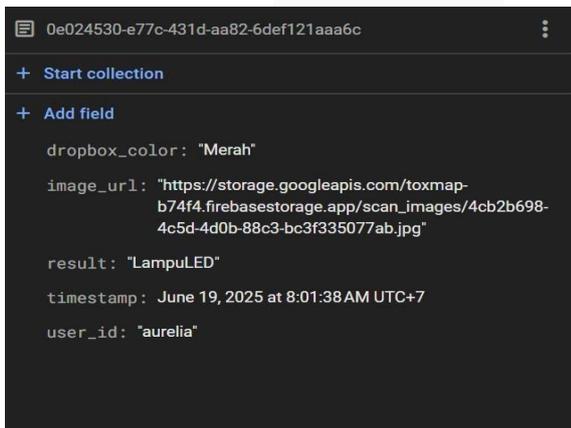
Gambar 2 Hasil Endpoint /predict

Pada Gambar 2, ditampilkan antarmuka Swagger dari FastAPI yang memperlihatkan keberhasilan dalam mengakses dan mengeksekusi endpoint /predict. Swagger sangat membantu dalam tahap pengujian dan validasi endpoint, karena memungkinkan input langsung berupa form-data tanpa perlu membangun antarmuka frontend terlebih dahulu.

B. Integrasi Firebase dan Manajemen Data

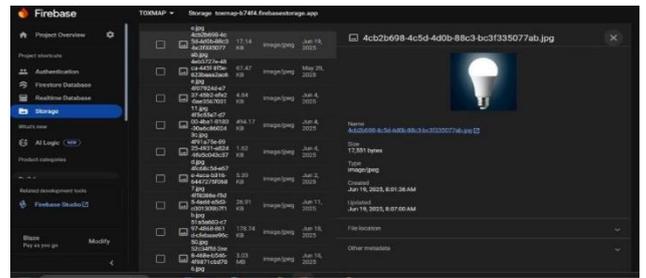
Backend yang dibangun dengan FastAPI secara langsung terhubung dengan beberapa layanan Firebase, yaitu Authentication untuk login dan registrasi pengguna, Firestore untuk menyimpan metadata hasil klasifikasi, dan Storage untuk menyimpan file gambar hasil scan.

Hasil klasifikasi disimpan dalam Cloud Firestore dengan struktur data yang menyertakan informasi seperti user_id, image_url, result, dropbox_color, dan timestamp. Sebagaimana ditunjukkan pada Gambar 3, setiap entri dalam koleksi scan_history mencatat data klasifikasi dari pengguna, termasuk URL gambar yang diunggah ke Firebase Storage dan hasil klasifikasinya. Hal ini memudahkan pelacakan data klasifikasi dan riwayat penggunaan sistem oleh setiap user.



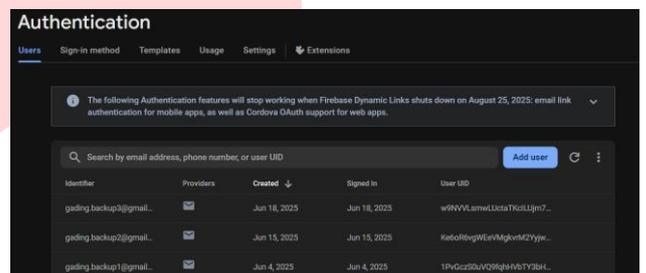
Gambar 3 Hasil di Firestore

Gambar hasil scan disimpan dalam Firebase Storage secara otomatis begitu proses prediksi selesai dijalankan. Gambar 4 menunjukkan tampilan antarmuka Firebase Storage yang berisi file gambar dengan struktur penyimpanan yang konsisten, disertai informasi nama file, ukuran, waktu upload, serta preview file. Sistem ini memisahkan data file dan metadata-nya untuk menghindari redundansi dan meningkatkan efisiensi pengelolaan data.



Gambar 4 Hasil Di Firebase Storage

Pada Firebase Authentication, pengguna yang mendaftar dari aplikasi TOXMAP akan otomatis tercatat pada daftar user seperti yang ditampilkan pada Gambar 5. Setiap pengguna memiliki identitas email dan waktu pembuatan akun yang tercatat dalam sistem, memastikan bahwa autentikasi pengguna berlangsung secara aman dan terstruktur.

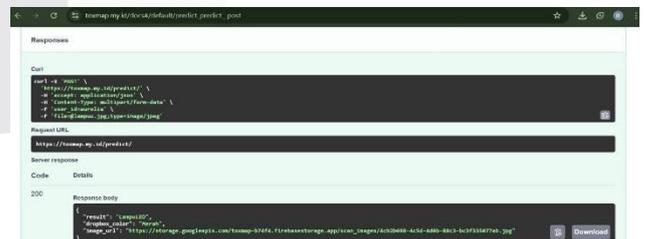


Gambar 5 Hasil di Firebase Authentication

Melalui kombinasi tiga layanan Firebase ini, sistem backend TOXMAP terbukti mampu menangani manajemen data secara terintegrasi dan efisien. Penggunaan Firebase juga memungkinkan backend untuk mendukung skala penggunaan yang lebih luas dengan kecepatan akses dan konsistensi data yang terjaga.

C. Integrasi Backend dan Model SVM

Model SVM yang telah dilatih sebelumnya diekspor dalam format .pkl dan kemudian dimuat ke dalam backend saat aplikasi dijalankan. Backend FastAPI bertugas menampung permintaan gambar dari pengguna, melakukan preprocessing jika diperlukan, dan meneruskan data gambar ke model SVM untuk diproses secara real time.



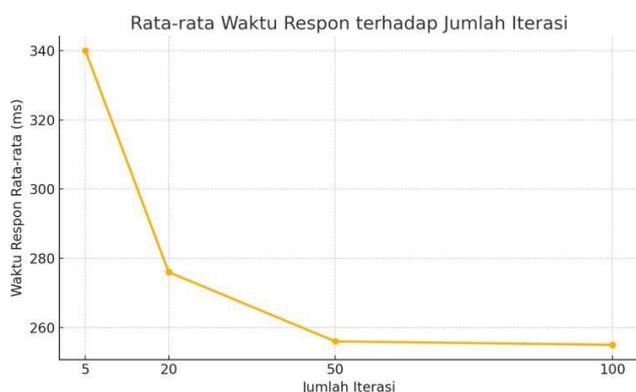
Gambar 6 Hasil Prediksi Gambar Dengan SVM

Berdasarkan Gambar 6, sistem memberikan respons dengan status kode 200 yang menandakan permintaan berhasil diproses. Di dalam respons tersebut, ditampilkan hasil klasifikasi dengan label "LampuLED", serta URL file gambar yang disimpan di Firebase Storage. Proses ini menunjukkan bahwa backend mampu menjalankan pipeline inferensi secara utuh—mulai dari menerima input gambar, memanggil

model SVM .pkl, hingga menghasilkan output berupa label klasifikasi dan metadata terkait.

D. Uji Real-Time Response

Pengujian Time Respons menjadi sangat penting karena aplikasi harus mampu melakukan klasifikasi secara real-time, terutama ketika pengguna mengunggah gambar sampah dan menunggu hasil klasifikasi dalam hitungan detik. Untuk memastikan performa tersebut, dilakukan pengujian waktu respon menggunakan Postman dengan metode iteratif sebanyak 5, 20, 50, hingga 100 kali eksekusi terhadap endpoint /predict.



Gambar 7 Hasil Uji Time Respons

Hasil pengujian ini ditampilkan pada Gambar 7, yang menunjukkan grafik rata-rata waktu respon terhadap jumlah iterasi. Terlihat bahwa pada 5 iterasi pertama, waktu respon masih cukup tinggi di atas 340 milidetik. Namun, seiring dengan meningkatnya jumlah iterasi, sistem menunjukkan kestabilan yang lebih baik. Pada 20 iterasi, rata-rata waktu turun secara signifikan ke kisaran 275 ms, dan setelah 50 iterasi, sistem menunjukkan kestabilan waktu respon di bawah 260 ms. Grafik ini menegaskan bahwa sistem backend memiliki performa yang adaptif terhadap beban dan mampu menjaga efisiensi komunikasi antara client dan server.

V. KESIMPULAN

Penelitian ini menunjukkan bahwa perancangan sistem backend TOXMAP dengan pendekatan real-time menggunakan FastAPI dan Firebase mampu mendukung proses klasifikasi sampah B3 secara efisien serta menyediakan informasi lokasi dropbox secara akurat. Sistem berhasil mengimplementasikan arsitektur RESTful API dengan endpoint yang spesifik dan dapat diakses oleh aplikasi mobile secara responsif. Pemrosesan klasifikasi gambar berbasis model *Support Vector Machine* (SVM) telah terintegrasi dengan baik dalam alur backend, memungkinkan hasil prediksi dapat dikembalikan dalam waktu respons rata-rata di bawah 300 ms.

Firebase tidak hanya digunakan sebagai layanan autentikasi, tetapi juga sebagai penyedia basis data cloud yang menyimpan informasi pengguna dan metadata klasifikasi. Selain itu, hasil uji komunikasi menunjukkan bahwa sistem dapat merespons input dari pengguna secara konsisten dan cepat, baik dalam skenario simulasi maupun real-time.

Dengan demikian, sistem backend TOXMAP berhasil menjalankan aplikasi TOXMAP secara menyeluruh sebagai solusi teknologi yang dapat meningkatkan kesadaran masyarakat dalam memilah dan menangani sampah B3. Implementasi sistem ini diharapkan dapat mendorong perilaku yang lebih peduli lingkungan melalui kemudahan akses klasifikasi sampah langsung dari perangkat mobile.

REFERENCE

- [1] Arah Environmental. "Apa Itu Limbah B3 dan Jenis-Jenis Limbah B3 yang Sering Kita Abaikan". Tersedia: <https://arahenvironmental.com/apa-itu-limbah-b3-dan-jenis-jenis-limbah-b3-yang-sering-kita-abaikan/>, August. 15, 2020 [Jul. 15, 2025].
- [2] A. Nursabrina, T. Joko, dan O. Septiani. (2021). "Kondisi Pengelolaan Limbah B3 Industri di Indonesia dan Potensi Dampaknya: Studi Literatur." *Jurnal Riset Kesehatan*, 13(1), pp. 80–89. Tersedia: <https://jurisikes.com/index.php/jrk/article/view/1841>. [Jul. 15, 2025].
- [3] G. M. Nyabuto, V. Mony, & S. Mbugua. (2024). "Architectural Review of Client–Server Models." *International Journal of Scientific Research & Engineering Trends (IJSRET)*, 10(1), pp. 139–143. Tersedia: https://ijsret.com/wp-content/uploads/2024/01/IJSRET_V10_issue1_125.pdf. [Jul. 15, 2025]
- [4] N. Barochiya, "Unlocking the Power of Firebase: A Comprehensive Guide to Google's BaaS Platform," *Medium*. Tersedia: <https://medium.com/@nandeepbarochiya/unlocking-the-power-of-firebase-a-comprehensive-guide-to-googles-baas-platform-827ed88567f2>, 2023. [Jul. 15, 2025].
- [5] FastAPI "Concurrency and async/await," *FastAPI Documentation*. Tersedia: <https://fastapi.tiangolo.com/async/>, 2025 [Jul. 15, 2025].
- [6] C. T. Ho, "How to Use FastAPI for Machine Learning," *JETBrains*. Tersedia: <https://blog.jetbrains.com/pycharm/2024/09/how-to-use-fastapi-for-machine-learning/>, 2024 [Jul. 15, 2025].
- [7] Gusti, A. "API Endpoint: Pengertian, Manfaat dan Cara Menemukannya." *IDwebhost*. Tersedia: <https://idwebhost.com/blog/api-endpoint/>, 2025 [Jul. 15, 2025].
- [8] Bogner, J., Kotstein, S., & Pfaff, T. (2023). "Do RESTful API design rules have an impact on the understandability of Web APIs?" *Empirical Software Engineering*. Vol. 28(132). Tersedia: <https://link.springer.com/article/10.1007/s10664-023-10367-y> [Jul. 15, 2025].