

# Perancangan Sistem Real-Time Berbasis Firebase Untuk Klasifikasi Sampah Organik dan An-Organik Dengan Machine Learning

1<sup>st</sup> Arfio Ardana

Telecommunication Engineering,  
School of Electrical Engineering  
Telkom University  
Bandung, Indonesia

arfioardana@student.telkomuniversity.ac.id

2<sup>nd</sup> Khoerunisa Alfin

Telecommunication Engineering,  
School of Electrical Engineering  
Telkom University  
Bandung, Indonesia

khoerunisaalfin@student.telkomuniversity.ac.id

3<sup>rd</sup> Yugo Pratama

Telecommunication Engineering,  
School of Electrical Engineering  
Telkom University  
Bandung, Indonesia

yugoprata@student.telkomuniversity.ac.id

4<sup>th</sup> Yulintyandra Puja Antanita

Telecommunication Engineering,  
School of Electrical Engineering  
Telkom University  
Bandung, Indonesia

yulintyandra@student.telkomuniversity.ac.id

5<sup>th</sup> Rita Purnamasari

Telecommunication Engineering,  
School of Electrical Engineering  
Telkom University  
Bandung, Indonesia

ritapurnamasari@telkomuniversity.ac.id

6<sup>th</sup> Khaerudin Saleh

Telecommunication Engineering,  
School of Electrical Engineering  
Telkom University  
Bandung, Indonesia

khaerudin@telkomuniversity.ac.id

**Abstract**—Sampah adalah benda yang tidak diinginkan atau tidak terpakai lagi yang terbuat dari bahan organik atau anorganik yang dapat menimbulkan penyakit jika tidak dikelola dengan baik. Untuk mengatasi hal tersebut, dilakukan perancangan RESIK aplikasi pengelolaan sampah berbasis *mobile* dengan integrasi teknologi *machine learning*. Aplikasi ini dirancang untuk mendeteksi jenis sampah menggunakan model YOLOv8 serta melakukan identifikasi produk melalui *object detection* dan *barcode decoder* secara *real-time*. Layanan Firebase digunakan untuk autentikasi pengguna, *dataset barcode*, dan *reward* pengguna. Penggunaan *frontend* dibangun menggunakan Flutter sehingga dapat berjalan *multi platform* dan mengakses kamera pengguna secara *real-time*. Hasil pengujian API menunjukkan bahwa sistem dapat melakukan respon kurang dari 2 detik dengan akurasi klasifikasi dan responsivitas yang baik ada pengguna. Arsitektur ini memberikan infrastruktur yang ringan dan mudah diakses untuk meningkatkan pengelolaan sampah dengan terstruktur.

**Kata kunci**—*firebase, backend, monolith*

## I. PENDAHULUAN

Permasalahan pengelolaan sampah menjadi semakin krusial, terutama di negara-negara berkembang seperti Indonesia, di mana volume sampah terus meningkat sementara kesadaran masyarakat untuk memilah sampah masih rendah. Menurut Kementerian Lingkungan Hidup dan Kehutanan (KLHK), total sampah yang dihasilkan pada tahun 2022 mencapai sekitar 68,7 juta ton, dengan lebih dari 41% berupa sampah organik, terutama dari sumber rumah tangga [1]. Penanganan sampah yang tidak tepat dan kurangnya pemilahan di tingkat sumber menyebabkan sampah tercampur, yang sulit didaur ulang dan berkontribusi terhadap degradasi lingkungan serta risiko kesehatan masyarakat.

Berbagai pendekatan digital telah dikembangkan untuk mengatasi masalah ini, termasuk aplikasi berbasis seluler yang bertujuan mendorong perilaku pemilahan sampah yang mandiri dan terstruktur. Salah satu pendekatan praktis adalah memanfaatkan pemindaian kode batang untuk

mengidentifikasi jenis produk, yang memungkinkan pengguna memindai kode produk dan menerima informasi langsung tentang kategori sampah yang sesuai. Namun, sebagian besar aplikasi kode batang yang ada terbatas pada fungsi dekoder dasar dan tidak mengintegrasikan sistem klasifikasi cerdas atau desain arsitektur perangkat lunak yang efisien [2].

Untuk permasalahan tersebut dikembangkan aplikasi RESIK, aplikasi *mobile* berbasis Flutter dan diintegrasikan teknologi *machine learning* dengan mengintegrasikan hal tersebut aplikasi ini berfungsi sebagai media edukasi, pengklasifikasian sampah dan pemberian hadiah terhadap pengguna ketika melakukan setoran sampah. Proses klasifikasi dilakukan secara *real-time* oleh kamera pengguna dengan bantuan *machine learning* model YOLOv8, arsitektur *backend* juga dibuat berbasis JavaScript yang dibangun menggunakan Node.js dengan keunggulan yang cepat, ringan dan mudah diintegrasikan dengan model *machine learning*. Model yang digunakan adalah YOLOv8 yang telah dilatih menggunakan data gambar gambar sampah.

Penelitian ini berfokus pada perancangan dan implementasi sistem *backend* RESIK sebagai solusi teknologi untuk membantu identifikasi dan pengolahan sampah organik & an-organik, dengan adanya penelitian ini diharapkan menjadi bentuk *educational movement* untuk meningkatkan kesadaran masyarakat tentang pentingnya pengelolaan sampah.

## II. KAJIAN TEORI

### A. Arsitektur Monolith

Sistem Arsitektur monolitik adalah jaringan komputasi tunggal yang besar dengan satu basis kode yang menggabungkan semua urusan bisnis [3]. arsitektur ini memudahkan tim untuk membuat pengembangan yang lebih sederhana dan cepat, proses build dan deployment juga lebih mudah karena hanya perlu membangun satu aplikasi utuh [4].

## B. Pengelolaan Sampah dan Solusi Berbasis Teknologi

Pengelolaan sampah masih menjadi isu krusial di banyak negara berkembang, termasuk Indonesia. Urbanisasi dan pertumbuhan penduduk yang pesat telah menyebabkan lonjakan produksi sampah, sementara penanganan yang tidak tepat berkontribusi terhadap masalah lingkungan dan kesehatan, serta hilangnya nilai ekonomi dari bahan daur ulang [5].

Salah satu inisiatif berbasis komunitas di Indonesia adalah program bank sampah, yang mengajak masyarakat memilah dan mengumpulkan sampah daur ulang dengan memberikan imbalan tertentu. Namun, meskipun memiliki potensi besar, program ini masih menghadapi hambatan seperti rendahnya tingkat partisipasi masyarakat dan keterbatasan infrastruktur digital [6].

## C. Firebase Authentication, Firestore and storage

Firebase menyediakan dukungan backend untuk penyimpanan data real-time dan manajemen pengguna. Dalam konteks deteksi gambar, Firebase memungkinkan pengunggahan dan pengambilan gambar yang diberi anotasi secara langsung dan memungkinkan sinkronisasi di berbagai klien, sehingga ideal untuk aplikasi [7].

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 dan Firebase Storage mengelola file gambar yang dikirim dari aplikasi mobile sebelum diproses di backend [8].

Prinsip serupa dapat diadaptasi pada sistem klasifikasi sampah dengan memanfaatkan layanan *cloud backend* seperti Firebase untuk menyimpan dan memproses data hasil klasifikasi secara langsung [9].

## D. Endpoint API

Endpoint API merupakan titik akses pada sistem backend yang berfungsi sebagai penghubung antara frontend atau klien dengan server melalui protokol HTTP [10]. Setiap endpoint dirancang untuk mewakili fungsi tertentu dalam sistem, sehingga memungkinkan pertukaran data secara efisien antara antarmuka pengguna dan logika backend. Dalam arsitektur REST API, metode POST digunakan untuk mengirim data, seperti file atau input pengguna, sedangkan metode GET digunakan untuk mengambil data dari server [10]. Perancangan endpoint yang tepat sangat penting untuk mendukung komunikasi real-time serta menjaga kinerja sistem tetap optimal.

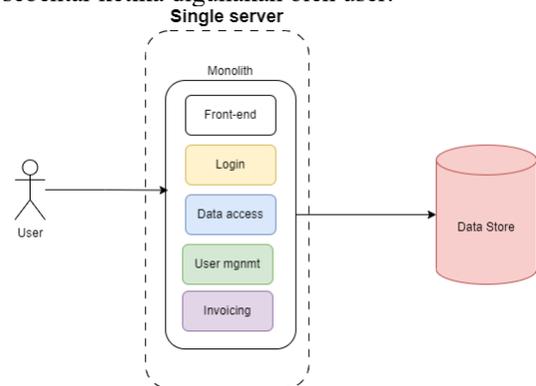
## III. METODE

### A. Arsitektur Sistem Monolith

Arsitektur monolitik merupakan pendekatan yang telah lama digunakan dalam pengembangan aplikasi. Pada model ini, seluruh komponen mulai dari antarmuka pengguna, logika bisnis, hingga lapisan akses data terintegrasi secara erat dan dijalankan sebagai satu kesatuan. Desain tersebut memudahkan proses pengembangan, pengujian, dan penerapan.

Pada Gambar. 1 dapat ditunjukkan arsitektur monolith secara general, dimana pengaplikasian ini diterapkan pada

model YOLOv8 yang langsung diintegrasikan dengan model *frontend* sedang untuk *barcode scanner* diletakkan pada backend sehingga pemomresan hasil gambar hanya memakan waktu sebentar ketika digunakan oleh user.



Gambar. 1. Arsitektur Monolith

### B. Firebase

Sistem Firebase pada aplikasi RESIK menggunakan 4 fitur yaitu, *Setor*, *Rewards*, *Notifications*, dan *users* untuk proses autentikasi pengguna dalam proses *login* dan *register* pengguna, *Setor* untuk data-data tentang hasil setoran pengguna, *reward* adalah hasil akumulasi uang dari hasil setoran pengguna, dan *notifications* adalah layanan Firebase Cloud Messaging (FCM) untuk melakukan pengiriman pesan lintas platform

Struktur Koleksi Firebase :

TABLE I. DATABASE : USERS

Field	Type Data	Deskripsi
UID	String	Identitas Unik
Username	String	Nama Pengguna
Email	String	Email Pengguna
Phone	String	Nomor Telepon Pengguna
CreatedAt?	Time Stamp	Tanggal dan Waktu saat dibuat

TABLE II. DATABASE : SETORAN INPUT

Field	Type Data	Deskripsi
Order ID	String	ID Unik
Nama	String	Nama Pengguna
Alamat	String	Alamat Pengguna
Tanggal	String	Waktu Trankasi
Catatan?	Time Stamp	Informasi Tambahan Terkait Setoran
Total harga	Number	Total nilai trankasi
Status	String	Proses setoran

TABLE III. DATABASE : SAMPAHITEM

Field	Type Data	Deskripsi
Jenis	String	Klasifikasi Sampah
Berat	Number	Berat Sampah
Hargat	Number	Total Harga Sampah

### C. Integrasi Model Machine Learning

Model *machine learning* pada aplikasi RESIK dirancang menggunakan YOLOv8 dan Barcode Decoder. Model YOLOv8 disimpan pada bagian *frontend*. Model disimpan dengan format (.tflite) dan dilakukan proses langsung pada bagian *frontend*, sedangkan pada barcode decoder dibuatkan fungsi pemanggilan *LookupBarcode* untuk mengakses file excel serta menerima request dan validasi.

Proses integrasi dilakukan dengan inialisasi model YOLO dengan memuat path "model.tflite", pengguna melakukan scan gambar, gambar dibaca dan diubah menjadi byte array, YOLO menghasilkan prediksi dan error ditangani.

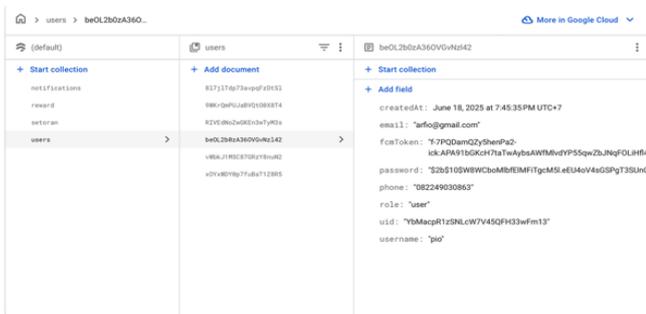
#### IV. HASIL DAN PEMBAHASAN

Pada pengujian *backend* dalam aplikasi RESIK difokuskan pada kecepatan *response time* API, keamanan data, dan memastikan aplikasi berjalan dengan baik dan responsif, pada bagian ini dibahas bagaimana masing-masing komponen backend direalisasikan baik secara fungsional, hingga uji coba *response time*

##### A. Integrasi Firebase

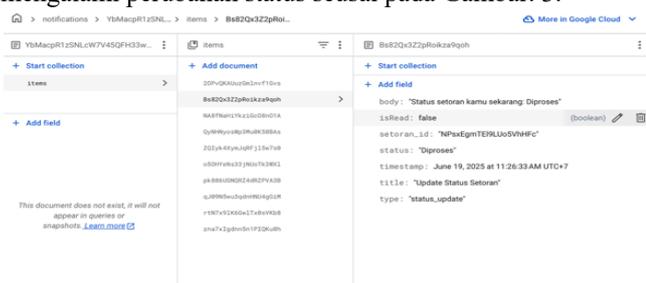
Pengintegrasian *backend* dengan menggunakan Bahasa javascript untuk layanan Firebase Console Messaging, Firebase Authentication dan Firebase Storage, masing masing mempunyai perannya sendiri.

Firebase authentication dipergunakan untuk melakukan autentikasi terhadap pengguna sehingga terdapat kode unik pada setiap pengguna. Pada Gambar. 2 berisikan data yang menjadi ciri khas pengguna yaitu "fcm token, dan uid" guna pengalaman data



Gambar. 2. Firebase : users

Pada setiap users juga memiliki firestore untuk setiap pengupdatean status sehingga terdapat *push notification* kepada pengguna aplikasi ketika hasil scan sampah mereka mengalami perubahan status sesuai pada Gambar. 3.



Gambar. 3. Firebase : notifications

Firebase Authentication, pengguna yang mendaftar pada aplikasi RESIK datanya akan tersimpan pada Firebase Authentication yang berisikan data email, uid, dan tanggal dibuat, hal ini memastikan setiap pengguna memiliki kode uniknya tersendiri.

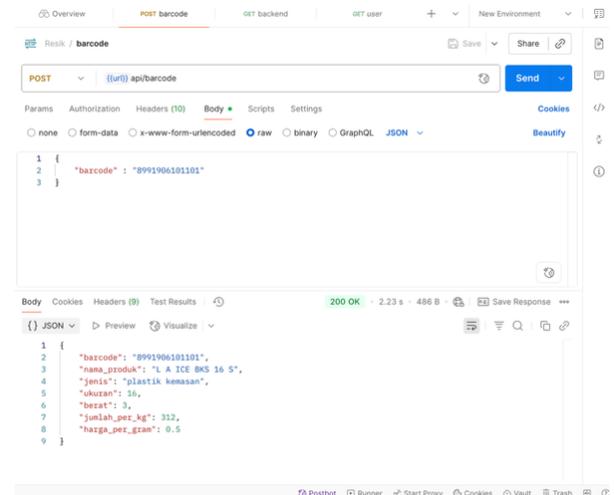
Melalui 3 layanan ini, penggunaan *backend* mampu menangani *request* dan *response* setiap pengguna melakukan aksinya pada aplikasi.

Identifier	Providers	Created	Signed In	User UID
yugoprata003@gmail.c...	📧	Jul 23, 2025	Jul 29, 2025	77oVfGxbmDv4RuOfvca4...
yuliniyandra@gmail.com	📧	Jul 18, 2025	Jul 18, 2025	nID0ZaE8ze0mN3WdHb8...
syahrizalabyyy@gmail...	📧	Jul 13, 2025	Jul 13, 2025	ANWpGkQpWda5V42DQnwg...
yugo123@gmail.com	📧	Jul 10, 2025	Jul 10, 2025	TJZPW078C9YCDJ7PyKdsgyN...
yulin8@gmail.com	📧	Jul 10, 2025	Jul 29, 2025	vvrkEAe0MgDXH54wNQY7wB...
yulin7@gmail.com	📧	Jul 10, 2025	Jul 10, 2025	Vn8KglzH2bBQbnIGW0v1W...
yulin1@gmail.com	📧	Jul 10, 2025	Jul 10, 2025	aiHsSgkMESQXbfSLg4b1mX...
yugo@gmail.com	📧	Jun 22, 2025	Jul 15, 2025	byAPUudR0Vbs7TK6FPb_PBS...
admin@gmail.com	📧	Jun 18, 2025	Jul 29, 2025	8SJA1uGCaWHNuzzW6wPmM...
erfo@gmail.com	📧	Jun 18, 2025	Jul 29, 2025	YBMacpR1zSNLcW7V45QF3wFm13

Gambar. 4. Firebase Authentications

##### B. Integrasi Backend

Model machine learning yang dilatih menggunakan YOLOv8 dan dipath pada frontend dengan format .tflite bertugas melakukan prediksi setiap pengguna melakukan scan sampah, berbeda dengan barcode yang dimuat ke dalam backend.

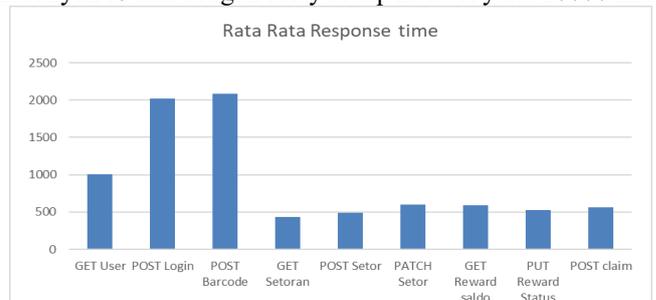


Gambar. 5. Pengujian API Barcode

Berdasarkan Gambar. 5, sistem memberikan kode 200 OK dan mengeluarkan outputnya dengan tepat sesuai dengan data pada identitas Barcode.

##### C. Detil Pengujian Response Time

Untuk menguji fungsionalitas setiap *endpoint* dalam aplikasi resik, digunakan pengujian postman sebagai alat bantu pengujian. Pengujian dilakukan dengan metode iterasi sebanyak 15 kali dengan delay setiap interval yaitu 10000 ms.



Gambar. 1. Rata rata response time

Pada Gambar. 6 dapat dilihat bahwasannya terdapat 2 endpoint yang memiliki reponse cukup tinggi setelah dilakukan pengujian 15 iterasi, nilai pada grafik juga adalah nilai response time rata-rata dari setiap *endpoint*.

## V. HASIL DAN PEMBAHASAN

Pengujian terhadap *endpoint* API menggunakan postman menunjukkan hasil yang baik dengan seluruh permintaan menghasilkan status *code* 200 OK. Waktu *respons* tertinggi tercatat pada *endpoint* POST Login dan POST *Barcode*, masing-masing sekitar 2000–2100 ms. Sementara itu, *endpoint* lainnya memiliki waktu respons di bawah 600 ms, yang mencerminkan performa sistem yang cepat dan stabil.

## ACKNOWLEDGEMENT

Penulis berterimakasih kepada pihak Universitas Telkom sehingga penulis dapat menyelesaikan penelitian ini.

## REFERENSI

- [1] A. Wulandari, G. W. Saraswati, and Y. F. Raihatuzzahra., "Pendampingan Manajemen Pengelolaan Sampah Rumah Tangga di Kelurahan Sambiroto," *Jurnal BUDIMAS*, vol. 5, no. 2, pp. 1-8, 2023.
- [2] R. Wudhikarn, P. Charoenkwan, and K. Malang, "Deep Learning in Barcode Recognition," *A Systematic Literature Review*, *IEEE Access*, vol. 10, no. 10.1109/ACCESS.2022.3143033., p. 8049–8072, 2022.
- [3] Chandler Harris, "Atlassian," Software, [Online]. Available: <https://www.atlassian.com/microservices/microservices-architecture/microservices-vs-monolith>. [Accessed 13 August 2025].
- [4] Kundan Gite, Ayush Gunjal, Prof. Vrushali M. Shinde., "Monolithic vs. Microservices Architecture: A Comparative Study of Software Development Paradigms," *International Scientific Journal of Engineering and Management*, vol. 04, no. 06, p. 01, 2025.
- [5] L. Julia Lingga, M. Yuana, N. Aulia Sari, H. Nur Syahida, C. Sitorus, and Shahron, "Sampah di Indonesia: Tantangan dan Solusi Menuju Perubahan Positif," *Innovative: Journal Of Social Science Research*, vol. 04, no. 04, p. 12235–12247, 2024.
- [6] D. H. A. P. Eldo et al., "Pembentukan Bank Sampah sebagai Solusi Pengelolaan Sampah di Desa.," *Jurnal Abdi Masyarakat Indonesia*, vol. 04, no. 01, p. doi: 10.54082/jamsi.1009, 2023.
- [7] Sneha Sonawane, Shraddha Rasal, Shrisha Patil, Kashish kute, Prof. V.J. Bodake, "Automatic Pothole Detection On Road Surface," *International Scientific Journal of Engineering and Management (ISJEM)*, vol. 04, no. 05, p. 2, 2025.
- [8] D. Intern, "dicoding," Pemrograman, 25 November 2020. [Online]. Available: <https://www.dicoding.com/blog/apa-itu-firebase-pengertian-jenis-jenis-dan-fungsi-kegunaannya/>. [Accessed 13 August 2025].
- [9] Irawati, I. D., Ramadan, D. N., Hadiyoso, S., Purnamasari, R., Budiman, G., Fachrurrozi, N. R., Yudiansyah, Y., Yutia, S. N., Tyas, S. H. Y., & Fajriyah, S. Z, *Workshop Implementasi Sistem Monitoring Dan Kendali Kualitas Air Pada Media Tanam Aquaponik Menggunakan Gawai Pintar*, vol. 2023, no. 10(1), pp. 44-53, Jurnal Abdi Insani.
- [10] Bogner, J., Kotstein, S., & Pfaff, T., "Do RESTful API design rules have an impact on the understandability of Web APIs," *Software Engineering*, vol. 28, no. 132, pp. <https://doi.org/10.1007/s10664-023-10367-y>, 2023.