

Implementasi Firebase Realtime Database sebagai Backend pada Sistem Monitoring Kapasitas dan Beban Tempat Sampah Berbasis IoT

1st Lukman Hakim

*School of Electrical Engineering
Telkom University
Jakarta, Indonesia*

kohlukman@student.telkomuniversity.a
c.id

2nd Kamelia Quzwain

*School of Electrical Engineering
Telkom University
Jakarta, Indonesia*

kquzwain@telkomuniversity.ac.id

3rd Sevierda Raniprima

*School of Electrical Engineering
Telkom University
Jakarta, Indonesia*

sevierdar@telkomuniversity.ac.id

Abstrak - Pengelolaan data yang cepat, akurat, dan terhubung secara menyeluruh menjadi aspek penting dalam penerapan Internet of Things (IoT), terutama pada sistem pemantauan kapasitas dan beban tempat sampah. Pada penelitian ini, Firebase Realtime Database dimanfaatkan sebagai backend utama untuk menyimpan, mengelola, dan menyinkronkan data sensor secara langsung antara perangkat IoT dan aplikasi Android. Sistem dibangun menggunakan NodeMCU ESP8266 yang terhubung dengan sensor ultrasonik dan sensor beban guna mengukur volume serta berat sampah. Data dari sensor dikirim melalui protokol HTTP ke Firebase, lalu secara otomatis ditampilkan pada aplikasi dengan pembaruan instan. Hasil pengujian menunjukkan bahwa Firebase mampu memproses data dengan latensi rendah, sehingga informasi yang diterima pengguna selalu up to date. Desain struktur data yang digunakan juga mempermudah pengembangan fitur lanjutan, seperti pemberitahuan otomatis saat kapasitas atau beban hampir penuh. Temuan ini membuktikan bahwa penggunaan Firebase sebagai backend dapat memberikan dukungan yang andal, responsif, dan fleksibel untuk sistem pemantauan berbasis IoT.

Kata kunci: *Internet of Things, Firebase Realtime Database, Backend, NodeMCU ESP8266, Monitoring Sampah*

I. PENDAHULUAN

Kemajuan teknologi *Internet of Things* (IoT) telah memberikan dampak besar pada berbagai bidang, termasuk pengelolaan sampah. Dengan konsep ini, perangkat fisik seperti sensor dan mikrokontroler dapat terhubung ke internet untuk mengirim, menerima, dan memproses data secara otomatis. Salah satu kunci keberhasilan penerapan IoT adalah kemampuan mengelola data secara cepat, akurat, dan dapat diakses pengguna secara real-time[1].

Dalam penelitian ini, Firebase Realtime Database digunakan sebagai backend utama untuk sistem monitoring kapasitas dan beban tempat sampah. Layanan ini memiliki kemampuan menyinkronkan data secara instan antara

perangkat IoT dan aplikasi Android, sehingga setiap perubahan data dari sensor langsung terlihat pada antarmuka pengguna. Firebase juga menawarkan kemudahan integrasi, dukungan lintas platform, dan penyimpanan data yang terstruktur, sehingga memudahkan pengembangan sistem dalam jangka panjang[2].

Sistem yang dibangun memanfaatkan NodeMCU ESP8266 yang terhubung dengan sensor ultrasonik dan sensor beban. Data dari kedua sensor dikirim ke Firebase melalui protokol HTTP dan kemudian ditampilkan di aplikasi Android. Dengan dukungan backend yang andal, sistem ini diharapkan mampu menyajikan informasi secara akurat, meminimalkan jeda pembaruan, dan membantu pengguna merespons dengan cepat saat kapasitas atau beban tempat sampah mendekati batas maksimum.

II. KAJIAN TEORI

A. Internet of Things (IoT)

Internet of Things (IoT) merupakan konsep yang menghubungkan berbagai perangkat fisik ke jaringan internet sehingga mampu saling bertukar data dan informasi secara otomatis. Dalam penerapannya, IoT memungkinkan integrasi sensor, mikrokontroler, dan layanan *cloud* untuk memantau dan mengontrol objek dari jarak jauh. Dalam konteks penelitian ini, IoT berperan sebagai penghubung antara perangkat pengukur kapasitas dan beban tempat sampah dengan aplikasi pemantauan berbasis Android[1].

B. NodeMCU ESP8266

NodeMCU ESP8266 adalah papan pengendali berbasis mikrokontroler yang sudah dilengkapi modul Wi-Fi, sehingga dapat langsung terhubung ke internet tanpa perangkat tambahan[3]. Konsumsi dayanya rendah, harganya terjangkau, dan mendukung integrasi dengan berbagai jenis sensor. Dalam sistem ini, NodeMCU berfungsi untuk menerima data dari sensor ultrasonik dan sensor beban,

kemudian mengirimkannya ke layanan *cloud* melalui protokol HTTP[4].

C. Firebase Realtime Database

Firebase Realtime Database adalah layanan basis data berbasis *cloud* yang dikembangkan oleh Google untuk menyimpan dan menyinkronkan data secara real-time. Setiap perubahan data di server langsung diperbarui pada semua perangkat yang terhubung tanpa jeda yang signifikan. Keunggulan ini menjadikannya pilihan ideal untuk sistem monitoring yang memerlukan pembaruan data instan. Pada penelitian ini, Firebase digunakan untuk menampung data hasil pembacaan sensor dan mengirimkannya ke aplikasi Android[2].

D. Integrasi Backend dan Frontend

Backend berperan sebagai pengelola dan penyedia data yang dihasilkan oleh perangkat IoT, sedangkan frontend adalah antarmuka yang menampilkan data tersebut kepada pengguna. Integrasi yang baik antara keduanya memastikan data dari sensor dapat tersaji dengan cepat, akurat, dan mudah dipahami. Dalam sistem ini, backend berbasis Firebase mengelola data dari NodeMCU dan mengirimkannya ke aplikasi Android, sehingga pengguna dapat memantau kapasitas dan beban tempat sampah secara real-time[5].

III. METODE

A. Perancangan Sistem Backend

Backend sistem dirancang menggunakan Firebase Realtime Database untuk mengelola dan menyinkronkan data secara real-time. NodeMCU ESP8266 digunakan sebagai pengendali utama yang menerima data dari sensor ultrasonik dan sensor beban, kemudian mengirimkannya ke Firebase melalui koneksi Wi-Fi. Struktur database dibuat terorganisasi, memisahkan setiap parameter data agar mudah diakses dan diolah.

B. Konfigurasi Firebase

Tahap ini meliputi pembuatan proyek baru di Firebase, pengaturan autentikasi, serta pembuatan *node* atau *path* khusus untuk menyimpan data kapasitas dan beban. Konfigurasi ini juga mencakup pengaturan aturan keamanan (security rules) agar hanya perangkat dan aplikasi yang berwenang dapat mengakses database.

C. Implementasi Kode pada NodeMCU

Kode program ditulis untuk membaca data dari sensor ultrasonik dan sensor beban, lalu mengirimkan data tersebut ke Firebase menggunakan protokol HTTP. Setiap data yang diterima dari sensor akan dikonversi menjadi format persentase, kemudian dikirim ke path yang telah disediakan di database.

D. Pengujian dan Validasi

Pengujian dilakukan untuk memastikan data yang dikirim oleh NodeMCU tersimpan dengan benar di Firebase dan dapat ditampilkan secara instan pada aplikasi Android. Selain itu, diuji juga kemampuan sistem dalam memberikan notifikasi otomatis ketika nilai kapasitas atau beban telah melewati ambang batas tertentu. Waktu respons, akurasi data, dan kestabilan koneksi menjadi fokus utama pengujian ini.

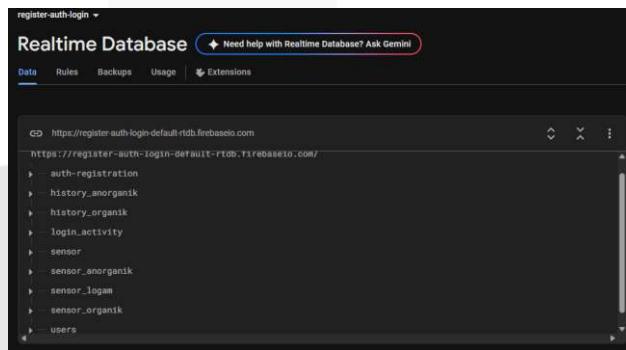
IV. HASIL DAN PEMBAHASAN

Bagian ini berisi paparan objektif peneliti terhadap hasil-hasil penelitian berupa penjelasan dan analisis terhadap penemuan-penemuan penelitian, penjelasan serta penafsiran dari data dan hubungan yang diperoleh, serta pembuatan generalisasi dari penemuan. Apabila terdapat hipotesis, maka pada bagian ini juga menjelaskan proses pengujian hipotesis beserta hasilnya.

Hasil penelitian harus disajikan secara jelas dan sistematis supaya mudah dibaca dan dipahami. Penyajian hasil penelitian dapat dilakukan dengan cara deskriptif (naratif), menggunakan tabulasi, tabel atau grafik, atau dengan menggunakan gabungan dua atau ketiganya secara sekaligus. Penggunaan ketiga cara tersebut disesuaikan dengan jenis data dan sejauh mana deskripsi data akan dijelaskan. Misalnya, pada awal peneliti memaparkan narasi temuannya, kemudian didukung dengan sajian data dalam bentuk tabulasi, tabel atau grafik. Peneliti juga menyajikan data-data hasil penelitian, kemudian didukung grafik dilanjutkan deskripsi naratif [10 pts]. Berikan kemungkinan pengembangan atau penelitian ke depan terkait penelitian ini

A. Hasil Implementasi Backend

Backend sistem dibangun menggunakan Firebase Realtime Database yang berfungsi untuk menyimpan, mengelola, dan menyinkronkan data sensor secara real-time. Setiap node dalam database memiliki peran khusus, seperti menyimpan data hasil pengukuran sensor, riwayat pemantauan, data pengguna, dan aktivitas login. Pengelompokan struktur ini bertujuan untuk mempermudah proses pengolahan data sekaligus memastikan sistem dapat dikembangkan secara modular di masa mendatang. Tampilan struktur database yang telah diimplementasikan dapat dilihat pada Gambar 1.



Gambar 1. Struktur Database Firebase

Berdasarkan Gambar 1, database terdiri dari beberapa node utama. Node auth-registration digunakan untuk menyimpan data pendaftaran pengguna, sedangkan history_anorganik dan history_organik merekam riwayat kapasitas dan beban pada masing-masing jenis tempat sampah. Sementara itu, sensor_logam, sensor_anorganik, dan sensor_organik berfungsi menyimpan data real-time dari sensor yang terpasang. Terdapat pula node login_activity untuk mencatat aktivitas masuk pengguna dan node users yang menyimpan informasi profil. Penataan struktur ini menunjukkan bahwa backend telah dirancang secara rapi, efisien, dan siap diintegrasikan dengan frontend aplikasi.

B. Integrasi Backend-Frontend

Hasil integrasi backend dan frontend terlihat pada fitur dashboard monitoring di aplikasi Android. Data yang dikirim oleh NodeMCU ESP8266 ke Firebase secara otomatis disinkronkan dan ditampilkan pada antarmuka aplikasi tanpa jeda yang signifikan. Fitur ini memuat informasi tingkat kepuaan kapasitas dan beban tempat sampah secara real-time, lengkap dengan indikator visual untuk memudahkan pengguna memahami kondisi terkini. Tampilan fitur monitoring dapat dilihat pada Gambar 2.



Gambar 2. Tampilan Monitoring

Gambar 2 menunjukkan bahwa antarmuka aplikasi menampilkan informasi kapasitas dalam bentuk persentase dan status visual, serta informasi beban dalam satuan kilogram. Sistem juga mampu memberikan notifikasi otomatis saat kapasitas atau beban mendekati batas maksimum. Pengujian menunjukkan bahwa pembaruan data berlangsung cepat dan akurat, sehingga pengguna dapat segera mengambil tindakan untuk mencegah penumpukan sampah. Integrasi ini membuktikan bahwa backend berbasis Firebase bekerja selaras dengan frontend, memberikan kinerja yang responsif dan mendukung pengelolaan sampah secara efektif.

C. Hasil Pengujian Kinerja Backend

Pengujian backend dilakukan untuk memastikan bahwa Firebase Realtime Database mampu mengelola dan menyinkronkan data dari perangkat IoT ke aplikasi Android secara cepat, akurat, dan stabil. Fokus pengujian mencakup kecepatan pembaruan data (latensi), kesesuaian nilai antara data sensor dan data yang tersimpan, keberhasilan pembaruan tampilan di aplikasi, serta kemampuan sistem dalam memberikan notifikasi otomatis ketika nilai kapasitas atau beban melampaui ambang batas yang telah ditentukan. Hasil pengujian disajikan pada Tabel 1.

Tabel 1. Hasil Pengujian Firebase

No	Jenis Pengujian	Deskripsi	Hasil	Keterangan
1	Delay dan Throughput	Mengukur jeda waktu dari data sensor masuk ke Firebase hingga muncul di aplikasi	< 2 detik	Stabil dan real-time
2	Akurasi Data	Membandingkan data sensor dengan data yang tersimpan di Firebase	100% sama	Tidak ada selisih nilai
3	Keberhasilan Update	Percentase keberhasilan pembaruan data di aplikasi	100%	Selalu ter-update
4	Notifikasi Ambang Batas	Uji kirim notifikasi saat kapasitas/beban > 80%	Berhasil	Notifikasi muncul di aplikasi

Berdasarkan Tabel 1, pengujian menunjukkan bahwa sinkronisasi data antara NodeMCU, Firebase, dan aplikasi Android berjalan dengan latensi di bawah dua detik, yang berarti pembaruan data terjadi hampir seketika. Akurasi data mencapai seratus persen, membuktikan bahwa tidak ada perbedaan nilai antara hasil pembacaan sensor dan data yang tersimpan di database. Tingkat keberhasilan pembaruan di aplikasi juga mencapai seratus persen, menandakan setiap perubahan data selalu berhasil ditampilkan. Selain itu, pengujian notifikasi membuktikan bahwa sistem mampu memberikan peringatan secara otomatis saat kapasitas atau beban melebihi ambang batas yang telah ditetapkan. Hasil ini mengonfirmasi bahwa backend Firebase berfungsi secara optimal dalam mendukung sistem monitoring berbasis IoT.

V. KESIMPULAN

Implementasi Firebase Realtime Database sebagai backend pada sistem monitoring kapasitas dan beban tempat sampah berbasis IoT terbukti mampu memberikan kinerja yang cepat, akurat, dan stabil. Integrasi antara NodeMCU ESP8266, sensor ultrasonik, sensor beban, dan Firebase memungkinkan data tersimpan serta diperbarui secara real-time, dengan latensi yang sangat rendah. Struktur database yang dirancang secara terorganisir mempermudah proses pengelolaan dan pengembangan sistem di masa depan. Hasil pengujian menunjukkan bahwa pembaruan data di aplikasi selalu berhasil, akurasinya mencapai seratus persen, dan sistem dapat mengirimkan notifikasi otomatis ketika kapasitas atau beban mendekati batas maksimum. Dengan demikian, penggunaan Firebase sebagai backend terbukti efektif dalam mendukung sistem monitoring berbasis IoT, sekaligus memberikan fondasi yang kuat untuk pengembangan fitur lanjutan di kemudian hari.

REFERENSI

- [1] A. Maulida, Choirunnisa Safa Fayzalmi, Faiz Ata Choirul Anaam, Dikhi Ardiansyah, Muhamad Ariel Gunawan, and Nur Ridwan, “Design of IoT-Based Smart Trash Bin Monitoring Using ESP32 and Firebase,” *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI)*, vol. 4, no. 2, pp. 420–428, Jul. 2025, doi: 10.62712/juktisi.v4i2.420.
- [2] M. M. Ahmed, E. Hassanien, and A. E. Hassanien, “IoT-based intelligent waste management system,” *Neural Comput Appl*, vol. 35, no. 32, pp. 23551–23579, Nov. 2023, doi: 10.1007/s00521-023-08970-7.
- [3] P. O. Ayeni and O. C. Adesoba, “IoT-based home control system using NodeMCU and Firebase,” *Journal of Edge Computing*, vol. 4, no. 1, pp. 17–34, May 2025, doi: 10.55056/jec.814.
- [4] N. Shittain Mitu, V. T. Vassilev, and M. Tabany, “Low Cost, Easy-to-Use, IoT and Cloud-Based Real-Time Environment Monitoring System Using ESP8266 Microcontroller.” [Online]. Available: <http://www.iaras.org/iaras/journals/ijitws>
- [5] N. Shahida Mohd Kasim, M. Fazli Mohd Sam, N. Awtif Mohd Maliku, S. Rosmaniza Ab Rashid, S. Normi Zabri, and A. Md Yusop, “I-Bin App: An Intelligent Waste Alert Monitoring System”, doi: 10.47772/IJRRISS.