

Perancangan dan Implementasi Sistem Kontrol Sepeda Motor Berbasis Internet of Things Menggunakan ESP32 dan Firebase Realtime Database

Firman Akbar

Universitas Islam Al – Azhar Mataram– Nusa Tenggara Barat

firman12@fmanil.com

| 15

ARTICLE INFO**Article history:**

Received Maret, 2026

Revised April , 2026

Accepted Juli, 2026

Available online Juli 2026

Kata Kunci *ESP32, Firebase Realtime Database, Internet of Things, Kodular, Sistem Kontrol Sepeda Motor.*



This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.
Copyright © 2026 by Author. Published by KABAR_MOE

ABSTRAK

Perkembangan teknologi *Internet of Things* (IoT) memberikan kemudahan dalam pemantauan dan pengendalian perangkat elektronik secara jarak jauh. Pada kendaraan roda dua, sebagian besar sistem pengoperasian masih dilakukan secara manual melalui saklar mekanis. Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem kontrol sepeda motor 4 saklar berbasis mikrokontroler ESP32 terintegrasi dengan *Firebase Realtime Database* dan aplikasi Android yang dirancang via Kodular. Sistem ini mengontrol empat fungsi utama kendaraan, yaitu kelistrikan kontak, starter mesin, alarm keamanan, dan lampu *hazard*. Komunikasi data berlangsung secara *real-time* dengan memanfaatkan protokol HTTP/Web socket. Berdasarkan hasil pengujian fungsionalitas, sistem mampu mengeksekusi perintah *relay* (nilai logika '1' untuk ON dan '0' untuk OFF) dengan interval *polling* pembacaan data sebesar 200 milidetik.

Integrasi antarmuka Android juga berhasil menampilkan status koneksi sistem (*ONLINE*) dan mempersiapkan modul pembacaan tegangan baterai (*battery voltage*). Sistem ini memberikan efisiensi, praktisitas, serta keamanan tambahan bagi pemilik kendaraan.

LATAR BELAKANG

Perkembangan teknologi informasi dan komunikasi pada era Industri 4.0 mengalami akselerasi yang sangat pesat serta membawa dampak signifikan terhadap pemenuhan kebutuhan manusia. Salah satu penemuan teknologi mendasar yang menjadi penggerak utama transformasi digital saat ini adalah *Internet of Things* (IoT). Secara mendasar, IoT didefinisikan sebagai sebuah paradigma di mana berbagai objek fisik dilengkapi dengan sensor, mikrokontroler, dan modul komunikasi untuk saling terhubung dan bertukar data melalui jaringan internet. Penerapan konsep ini memungkinkan proses pemantauan (*monitoring*) serta pengontrolan (*controlling*) suatu alat dapat dilakukan secara terpusat, nirkabel, dan tanpa dibatasi oleh jarak geografis selama terhubung ke jaringan.

Sektor otomotif, khususnya kendaraan roda dua atau sepeda motor, merupakan salah satu bidang yang paling potensial untuk diintegrasikan dengan teknologi IoT. Di berbagai negara berkembang seperti Indonesia, sepeda motor merupakan moda transportasi utama masyarakat karena tingkat mobilitasnya yang tinggi dan efisiensi biayanya. Meskipun demikian, sebagian besar sistem kendali dan keamanan pada sepeda motor komersial yang beredar saat ini masih

mengandalkan mekanisme kontrol manual, seperti kunci kontak mekanis dan tombol-tombol konvensional.

Penggunaan mekanisme manual pada kendaraan roda dua memiliki beberapa keterbatasan fungsional dan tingkat kerentanan keamanan yang relatif tinggi. Kunci kontak konvensional sangat rentan terhadap tindakan kriminal pencurian kendaraan bermotor (curanmor) dengan teknik pengrusakan kunci T. Selain itu, pengoperasian saklar fisik secara berulang dalam jangka panjang sering kali menimbulkan masalah keausan komponen listrik, korosi pada konektor, serta keterbatasan fleksibilitas dalam mengontrol fitur keamanan secara fleksibel dari jarak jauh. Guna mengatasi keterbatasan tersebut, perlunya transformasi sistem kontrol konvensional menjadi sistem kontrol cerdas berbasis digital (*smart motorcycle system*). Dengan memanfaatkan *smartphone* berbasis Android sebagai antarmuka pengendali (*user interface*), pengguna dapat menyalakan atau mematikan sistem kelistrikan kendaraan dengan lebih praktis. Pendekatan ini tidak hanya meningkatkan nilai kenyamanan (*convenience*), tetapi juga menambah lapisan keamanan (*security layer*) pada kendaraan.

| 16

Dalam merancang sistem kontrol sepeda motor cerdas berbasis IoT, pemilihan platform mikrokontroler merupakan faktor yang sangat krusial. Mikrokontroler ESP32 dipilih karena memiliki spesifikasi performa yang unggul, seperti prosesor *dual-core*, konsumsi daya yang efisien, serta modul Wi-Fi dan Bluetooth *built-in*. Integrasi modul Wi-Fi secara langsung pada *chip* ESP32 menghilangkan kebutuhan akan modul komunikasi nirkabel tambahan, sehingga mampu menekan biaya produksi dan memperkecil dimensi fisik perangkat. Selain faktor perangkat keras, arsitektur komunikasi data pada *cloud server* juga memegang peranan penting dalam keberhasilan sistem IoT berbasis waktu nyata (*real-time*). *Firebase Realtime Database* dari Google menyediakan layanan basis data berbasis *cloud* yang menyimpan dan mensinkronisasikan data antar-klien secara instan. Penggunaan *Firebase* memungkinkan pertukaran status antara aplikasi seluler dan mikrokontroler berlangsung dengan latensi yang sangat rendah.

Di sisi antarmuka pengguna, pemanfaatan platform *low-code* seperti Kodular memungkinkan pengembangan aplikasi Android dilakukan secara efisien. Melalui pendekatan pemrograman berbasis blok (*block programming*), logika komunikasi HTTP atau API menuju *Firebase* dapat dibangun secara terstruktur tanpa mengurangi fungsionalitas dan keandalan aplikasi. Berdasarkan uraian permasalahan di atas, penelitian ini bertujuan untuk merancang dan membuat sistem kontrol sepeda motor 4 saklar berbasis mikrokontroler ESP32 dengan memanfaatkan *Firebase Realtime Database* dan aplikasi Android. Sistem ini dirancang untuk mengendalikan empat fungsi utama kendaraan—yaitu sistem kontak kelistrikan, starter mesin, alarm keamanan, dan lampu *hazard*—serta memantau kondisi tegangan baterai secara *real-time*; (a) Merancang dan mengimplementasikan arsitektur sistem kontrol sepeda motor 4 saklar berbasis mikrokontroler ESP32, (b) Membangun integrasi dan komunikasi data secara *real-time* antara mikrokontroler ESP32, *Firebase Realtime Database*, dan aplikasi Android, dan (c) Memverifikasi keandalan fungsionalitas kendali empat *relay* (kontak, starter, alarm, dan *hazard*) serta pengujian status konektivitas sistem.

METODE PENELITIAN

Metode penelitian yang digunakan mencakup tahap perancangan perangkat keras (*hardware*), perancangan perangkat lunak (*software*), integrasi sistem, dan pengujian laboratorium.

A. Perancangan Perangkat Keras (Hardware)

Perangkat keras utama yang digunakan meliputi:

- **ESP32 Development Board:** Berfungsi sebagai unit pemroses data utama dan modul Wi-Fi.
- **Modul Relay 4-Saluran:** Dihubungkan ke pin GPIO ESP32:
 - GPIO 23: Mengendalikan Relay 1 (Kontak Utama)
 - GPIO 22: Mengendalikan Relay 2 (Motor Starter)
 - GPIO 21: Mengendalikan Relay 3 (Sistem Alarm)
 - GPIO 19: Mengendalikan Relay 4 (Lampu Hazard/Fungsi Tambahan)
- **Rangkaian Pembaca Tegangan Aki:** Sensor pembagi tegangan (*voltage divider*) untuk mengukur kondisi baterai sepeda motor.

B. Perancangan Perangkat Lunak (Software)

1. **Firestore Realtime Database Configuration:** Struktur JSON disiapkan dengan parameter relay1, relay2, relay3, relay4, battery, dan status.
2. **Pengembangan Aplikasi Android:** Dibuat menggunakan platform Kodular. Aplikasi ini mengirimkan nilai integer 1 (aktif) atau 0 (non-aktif) ke Firestore.
3. **Pemrograman ESP32:** Menggunakan Arduino IDE berbasis bahasa C++ dengan *library* WiFi.h dan *Firestore_ESP_Client.h*. Program diset untuk melakukan *polling* pembacaan data Firestore pada interval delay(200) ms.

HASIL DAN PEMBAHASAN

A. Pembahasan Hasil Perancangan Aplikasi Android dan Firestore

Aplikasi *Smart Motorcycle* berhasil dibangun menggunakan antarmuka Kodular. Tampilan aplikasi dirancang sederhana namun fungsional, terdiri dari label indikator *Status: ONLINE*, empat baris tombol kontrol (*Kontak*, *Starter*, *Alarm*, *Hazard*) yang masing-masing dilengkapi tombol *ON* dan *OFF*, serta indikator *Battery Voltage*. Struktur basis data waktu nyata pada Firestore telah berhasil dikonfigurasi dan diuji. Pada pengujian basis data, node <https://smart-motorcycle-7ce00-default-rtdb.asia-southeast1.firebaseio.com/> mampu menyimpan data secara *real-time*. Nilai awal dari parameter relay1 hingga relay4 dikondisikan pada logika '0', yang merepresentasikan seluruh sistem kendaraan dalam kondisi mati (*idle*).

B. Analisis Pengujian Fungsionalitas Sistem

Pengujian fungsionalitas secara menyeluruh dilakukan untuk mengevaluasi tingkat keberhasilan integrasi antara aplikasi Android, *Firestore Realtime Database*, mikrokontroler ESP32, dan modul *relay*. Pengujian pertama berfokus pada keandalan proses *handshake* jaringan nirkabel oleh ESP32. Ketika mikrokontroler diberi daya, *firmware* menginisialisasi modul Wi-Fi dan melakukan pemindaian SSID yang telah ditentukan. Berdasarkan data pengujian, ESP32 dapat terhubung ke titik akses Wi-Fi secara konsisten dan mendapatkan alamat IP lokal dalam waktu kurang dari 3 detik. Setelah koneksi Wi-Fi terstabilkan, ESP32 melanjutkan proses autentikasi dan pembuatan sesi komunikasi dengan server *Firestore Realtime Database*. Menggunakan token autentikasi anonim, sistem berhasil membangun saluran enkripsi data. Kehadiran variabel status

bertulis ONLINE pada Firebase dan aplikasi Android mengonfirmasi bahwa jalur komunikasi dua arah antara perangkat keras dan antarmuka pengguna telah siap beroperasi secara sempurna.

Pada pengujian eksekusi perintah *Relay 1* (Kontak Utama), ketika tombol *ON* pada aplikasi ditekan, aplikasi mengirimkan parameter nilai 1 ke node */relay1* di Firebase. ESP32 yang menjalankan pemindaian berulang (*loop*) pada fungsi *Firestore.RTDB.getInt()* mendeteksi perubahan logika data tersebut dalam rentang waktu terukur. Mikrokontroler kemudian memicu *pin* GPIO 23 menjadi bernilai HIGH, yang menyebabkan *relay* kontak terhubung dan menyalakan seluruh kelistrikan utama sepeda motor. Selanjutnya, pengujian dilakukan pada *Relay 2* yang berfungsi mengendalikan tombol *Starter* mesin. Berbeda dengan fungsi kontak, *relay starter* membutuhkan durasi pengaktifan sementara (*pulsed signal*). Pengguna menekan tombol *ON* untuk mengirim nilai 1 ke node */relay2* yang langsung mengaktifkan solenoid *starter* melalui GPIO 22. Setelah mesin menyala, pengguna menekan tombol *OFF* untuk mengembalikan data ke nilai 0, memutus arus ke motor *starter* guna mencegah kerusakan komponen.

Pengujian *Relay 3* (Sistem Alarm) dan *Relay 4* (Lampu *Hazard*) menunjukkan respons eksekusi yang serupa. Pengiriman data bernilai 1 pada node */relay3* berhasil menyalakan klakson/sirene alarm via GPIO 21. Sementara itu, pengiriman nilai 1 ke node */relay4* mengaktifkan GPIO 19 yang menginstruksikan modul pendedip lampu sein (*flasher*) untuk menyalakan kedua lampu sein kiri dan kanan secara bersamaan (*hazard*). Pengujian ini membuktikan bahwa keempat saluran *relay* dapat berjalan secara independen tanpa adanya kendala transmisi.

Tabel 1 merangkum hasil pengujian respons fungsionalitas saklar kendali:

Fungsi Kendali	Pin GPIO ESP32	Input Aplikasi	Data Firebase	Status Relay	Hasil Eksekusi
Kontak Utama	GPIO 23	Button Kontak ON	relay1: 1	Active (CLOSED)	Sesuai (Kelistrikan ON)
Kontak Utama	GPIO 23	Button Kontak OFF	relay1: 0	Inactive (OPEN)	Sesuai (Kelistrikan OFF)
Starter Mesin	GPIO 22	Button Starter ON	relay2: 1	Active (CLOSED)	Sesuai (Motor Starter ON)
Starter Mesin	GPIO 22	Button Starter OFF	relay2: 0	Inactive (OPEN)	Sesuai (Motor Starter OFF)
Alarm	GPIO 21	Button Alarm ON	relay3: 1	Active (CLOSED)	Sesuai (Sirene ON)
Alarm	GPIO 21	Button Alarm OFF	relay3: 0	Inactive (OPEN)	Sesuai (Sirene OFF)

Fungsi Kendali	Pin GPIO ESP32	Input Aplikasi	Data Firebase	Status Relay	Hasil Eksekusi
Lampu Hazard	GPIO 19	Button Hazard ON	relay4: 1	Active (CLOSED)	Sesuai (Hazard ON)
Lampu Hazard	GPIO 19	Button Hazard OFF	relay4: 0	Inactive (OPEN)	Sesuai (Hazard OFF)

Mengenai aspek latensi dan efisiensi sistem, penggunaan jeda waktu *delay*(200) milidetik pada *looping* utama ESP32 memberikan keseimbangan kinerja yang optimal. Jeda 200 ms ini cukup responsif bagi persepsi manusia (menghasilkan *delay* pengoperasian di bawah 0,5 detik) sekaligus mencegah terjadinya banjir permintaan (*request flooding*) ke *Firestore Realtime Database* yang berpotensi menyebabkan *socket timeout* atau peningkatan penggunaan kuota bandwidth.

Terakhir, pengujian pembacaan parameter tegangan baterai (*Battery Voltage*) menunjukkan keberhasilan integrasi data. Nilai tegangan diukur menggunakan analog-to-digital converter (ADC) ESP32 dan dikirimkan ke node */battery* pada *Firestore*. Pada saat pengujian awal tanpa pembacaan sensor aktif, sistem menampilkan data *default* sebesar 0.00 Volt. Ketika dihubungkan ke akuisisi data aktual, nilai tegangan aki ter-update secara *real-time* di aplikasi seluler, memberikan informasi kesehatan aki kepada pengguna.

SIMPULAN

1. Sistem kontrol sepeda motor 4 saklar berbasis mikrokontroler ESP32 berhasil dirancang dan dapat berfungsi sesuai spesifikasi.
2. Integrasi antara aplikasi Android (Kodular), *Firestore Realtime Database*, dan ESP32 dapat berjalan secara *real-time* dengan tingkat keandalan yang baik.
3. Pengendalian empat fungsi utama sepeda motor—yaitu kelistrikan kontak, starter mesin, alarm keamanan, dan lampu *hazard*—dapat dieksekusi secara independen menggunakan kombinasi nilai logika '1' dan '0' dengan interval respons pembacaan 200 ms.
4. Aplikasi seluler secara efektif dapat menampilkan status konektivitas sistem (*ONLINE*) serta menyajikan informasi pembacaan tegangan aki kendaraan sebagai fitur pemantauan kondisi baterai.

DAFTAR PUSTAKA

- Denton, T. (2017). *Automobile Electrical and Electronic Systems* (5th ed.). Routledge.
- Khawas, C., & Shah, P. (2018). Application of *Firestore* in Android App Development-A Review. *International Journal of Computer Applications*, 179(46), 49-53.
- Maier, A., Sharp, A., & Vagapov, Y. (2017). Comparative analysis and practical implementation of the ESP32 microcontroller module for the Internet of Things. In *2017 Internet Technologies and Applications (ITA)* (pp. 143-148). IEEE.
- Ray, P. P. (2018). A survey on Internet of Things architectures. *Journal of King Saud University-Computer and Information Sciences*, 30(3), 291-319.

Saputra, G. D. (2026). *Laporan Tugas Perancangan dan Pembuatan Sistem Kontrol Sepeda Motor 4 Saklar Berbasis ESP32 dengan Firebase*. Laporan Tugas Akhir/Proyek.

Sethi, P., & Sarangi, S. R. (2017). Internet of Things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering*, 2017, 1-25.