Journal of System & Technology
ISSN: 3110-0171
Vol. 2, No. 1, June 2026

IMPLEMENTASI SISTEM KENDALI ROBOT 4WD BERBASIS KEYWORD SPOTTING DENGAN EDGE AI PADA MIKROKONTROLER ESP32-S3

Rehandra[1], Feri Candra[1]*
[1]Teknik Informatika, Universitas Riau, Indonesia
rehandra2004@gmail.com, feri@eng.unri.ac.id
*Corresponding author: feri@eng.unri.ac.id

Abstract: Meningkatnya adopsi Kecerdasan Buatan (AI) dalam dunia pendidikan telah menciptakan kesenjangan yang signifikan antara alat pembelajaran robotika konvensional dengan kebutuhan industri saat ini. Platform yang ada, seperti robot Edubit berbasis BBC Micro:bit yang digunakan di Educourse.id, belum memiliki kemampuan pemrosesan AI secara asli (native) dan memiliki biaya yang relatif tinggi, sehingga kurang memadai untuk mengajarkan konsep embedded AI modern. Penelitian ini menyajikan desain dan implementasi purwarupa robot 4WD yang dikendalikan suara menggunakan teknologi Keyword Spotting yang diimplementasikan sepenuhnya pada perangkat melalui Edge AI. Sistem ini dibangun menggunakan mikrokontroler ESP32-S3 dengan memanfaatkan arsitektur dual-core Xtensa LX7 yang mendukung instruksi vektor untuk akselerasi jaringan saraf. Input audio ditangkap melalui mikrofon MEMS INMP441 melalui protokol I2S dan diproses oleh framework ESP-SR, yang mengintegrasikan WakeNet untuk deteksi wake word ("Hi ESP") dan MultiNet7 untuk klasifikasi perintah navigasi. Lima perintah arah didukung dalam sistem ini: maju, mundur, kiri, kanan, dan berhenti. Arsitektur FreeRTOS dual-core dengan penempatan tugas (task pinning) secara eksplisit memisahkan inferensi AI pada Core 1 dari logika Finite State Machine dan transmisi nirkabel pada Core 0 untuk memastikan performa real-time yang deterministik. Perintah navigasi yang telah divalidasi dikirimkan secara nirkabel ke unit robot menggunakan protokol ESP-NOW melalui struktur paket 7-byte yang ringkas dengan verifikasi integritas XOR checksum. Evaluasi sistem dilakukan menggunakan Black Box Testing melalui 50 percobaan yang melibatkan lima penutur berbeda di lingkungan dalam ruangan dengan tingkat kebisingan sedang. Hasil penelitian menunjukkan akurasi pengenalan perintah sebesar 100% di seluruh lima perintah navigasi, dengan skor kepercayaan rata-rata 17,62%. Temuan ini mengonfirmasi bahwa sistem yang diusulkan sangat andal dan layak digunakan sebagai platform pembelajaran Edge AI yang terjangkau dan interaktif untuk pendidikan robotika.

Keywords—Edge AI, ESP32-S3, Keyword Spotting, Robotika Pendidikan, ESP-NOW

Article Information:
Received: 01-Mei-2026; Revised: 21-Jun-2026; Accepted: 26-Jun-2026; Published: 27-Jun-2026

This work is licensed under a **CC BY-SA**. Copyright ©2026 by Author. Published by Universitas Riau.

I. PENDAHULUAN

Kemajuan pesat Kecerdasan Buatan (AI) telah mengubah lanskap sistem tertanam secara signifikan, memunculkan paradigma yang dikenal sebagai *Edge AI*. Paradigma ini, inferensi AI dieksekusi secara langsung pada perangkat keras dengan sumber daya terbatas tanpa bergantung pada infrastruktur awan (*cloud*) [1]. Subbidang yang menonjol dari bidang ini, yaitu *Tiny Machine Learning* (TinyML), memungkinkan penerapan model jaringan saraf ringan pada mikrokontroler dengan memori dan kapasitas komputasi yang terbatas [2]. Dalam domain robotika pendidikan, pergeseran ini menghadirkan peluang menarik untuk memperkenalkan aplikasi AI dunia nyata kepada siswa melalui alat pembelajaran interaktif yang bersifat praktis. Interaksi manusia-mesin berbasis suara, secara khusus, telah muncul sebagai salah satu antarmuka yang paling intuitif untuk mendemonstrasikan konsep AI dalam pengaturan praktis, karena menghilangkan kebutuhan akan pengontrol fisik dan mencerminkan arah sistem cerdas modern [3].

Terlepas dari potensi tersebut, banyak platform pendidikan robotika yang digunakan saat ini belum mampu mengimbangi perkembangan teknologi ini. Di Educourse.id, sebuah perusahaan teknologi pendidikan berbasis STEM yang beroperasi di berbagai kota di Indonesia, alat pembelajaran utama yang digunakan adalah robot Edubit yang dibangun di atas mikrokontroler BBC Micro:bit. Meskipun berfungsi untuk instruksi pemrograman dan elektronik dasar, arsitektur BBC Micro:bit tidak dirancang untuk menangani beban kerja *deep learning*. Kapasitas SRAM dan kecepatan pemrosesannya yang terbatas membuat pengenalan suara pada perangkat (*on-device speech recognition*) hampir tidak mungkin dilakukan tanpa modul eksternal yang mahal. Selain itu, harga unit platform yang relatif tinggi mengurangi aksesibilitasnya bagi pasar pendidikan yang lebih luas, terutama sekolah menengah dengan anggaran terbatas.

Sistem robot yang dikendalikan suara saat ini sebagian besar mengatasi masalah ini dengan melimpahkan (*offloading*) pengenalan suara ke layanan berbasis awan seperti Google Assistant atau Amazon Alexa [4]. Meskipun efektif dalam lingkungan terkendali, pendekatan ini menimbulkan latensi yang signifikan akibat keterlambatan transmisi jaringan dan menciptakan

ketergantungan kritis pada konektivitas internet yang stabil — suatu kondisi yang tidak dapat dijamin di banyak ruang kelas dan laboratorium di Indonesia. Selain itu, transmisi data audio ke server eksternal menimbulkan kekhawatiran terkait privasi pengguna dan keamanan data [5]. Keterbatasan-keterbatasan ini secara kolektif mendorong kebutuhan akan sistem pengenalan suara *on-device* yang sepenuhnya luring (*offline*) dan dapat beroperasi secara andal dalam kondisi pendidikan dunia nyata.

Makalah ini mengusulkan desain dan implementasi purwarupa robot *four-wheel-drive* (4WD) yang dikendalikan suara untuk mengatasi keterbatasan tersebut melalui integrasi teknologi *Keyword Spotting* (KWS) yang diterapkan sepenuhnya sebagai *Edge AI* pada mikrokontroler ESP32-S3. ESP32-S3 dipilih karena prosesor *dual-core* Xtensa LX7 yang dimilikinya memiliki dukungan instruksi vektor asli (SIMD), yang secara signifikan mempercepat inferensi jaringan saraf tanpa memerlukan ko-prosesor AI eksternal [6]. Sistem ini menggunakan *framework* ESP-SR yang dikembangkan oleh Espressif Systems, menggabungkan model WakeNet untuk deteksi *wake word* dan MultiNet7 untuk klasifikasi perintah navigasi, yang keduanya beroperasi sepenuhnya secara luring [7]. Arsitektur FreeRTOS *dual-core* dengan penempatan tugas (*task pinning*) secara eksplisit memastikan bahwa inferensi AI dan logika kontrol robot dieksekusi secara bersamaan tanpa interferensi satu sama lain, sementara protokol ESP-NOW menyediakan transmisi perintah nirkabel dengan latensi rendah antara unit pengontrol dan robot [8].

Sistematika penulisan makalah ini disusun sebagai berikut. Bagian II menguraikan landasan teori dan metodologi desain sistem, yang mencakup arsitektur perangkat keras, desain *firmware*, dan protokol komunikasi. Bagian III menyajikan hasil implementasi dan evaluasi kinerja sistem. Bagian IV menarik kesimpulan dari temuan tersebut serta menguraikan arah pengembangan di masa depan.

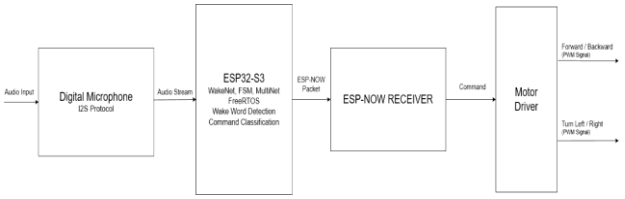
II. METODOLOGI PENELITIAN

Penelitian ini dilaksanakan melalui beberapa tahapan sistematis yang meliputi perancangan arsitektur perangkat keras, pengembangan *firmware* berbasis sistem operasi waktu nyata, serta integrasi model kecerdasan buatan pada perangkat *edge*.

A. Arsitektur Perangkatr Keras

Sistem kendali robot yang dikembangkan mengadopsi arsitektur *transmitter-receiver* yang memisahkan beban kerja pemrosesan sinyal suara dari kendali aktuator mekanis secara fisik. Pemisahan ini dirancang untuk mengisolasi proses inferensi AI yang bersifat komputasi intensif dari proses kendali motor yang menuntut responsivitas tinggi, sehingga stabilitas operasional kedua unit dapat terjaga secara independen. Gambaran menyeluruh mengenai interkoneksi antar komponen dalam sistem ini dapat dilihat pada Gambar 1.

Unit *transmitter* berfungsi sebagai pengolah suara cerdas yang dibangun di atas mikrokontroler ESP32-S3. Chip ini dipilih karena merupakan satu-satunya mikrokontroler di kelasnya yang secara *native* mendukung komputasi Edge AI melalui prosesor dual-core Xtensa LX7 yang dilengkapi instruksi vektor (SIMD/PIE). Instruksi vektor ini dirancang khusus untuk mempercepat operasi jaringan saraf tiruan dan pemrosesan sinyal digital langsung pada tingkat perangkat keras, tanpa memerlukan ko-prosesor AI eksternal. Selain itu, ketersediaan PSRAM hingga 8MB pada ESP32-S3 memungkinkan model pengenalan suara dimuat secara penuh ke dalam memori tanpa kompromi terhadap kualitas inferensi.



Gambar 1. Blok Diagram

Di sisi unit *receiver*, sebuah mikrokontroler ESP32-S3 kedua bertugas menerima paket instruksi navigasi secara nirkabel dan menerjemahkannya menjadi sinyal kendali aktuator. Unit ini terhubung ke modul *motor driver* yang mengatur distribusi daya ke empat motor DC penggerak roda. Sinyal *Pulse Width Modulation* (PWM) dibangkitkan berdasarkan kode perintah yang diterima untuk mengatur arah putaran dan kecepatan masing-masing motor secara presisi sesuai instruksi navigasi yang diberikan pengguna.

Manajemen catu daya pada kedua unit disuplai melalui koneksi USB pada masing-masing ESP32-S3. Pin tegangan 3.3V dari regulator internal mikrokontroler digunakan untuk menyuplai daya operasional mikrofon INMP441, sedangkan pin tegangan 5V (VBUS) dari jalur USB didistribusikan untuk menyuplai komponen tambahan seperti layar LCD 16×2 dan modul *joystick* pada unit *transmitter*. Seluruh komponen elektronik pada unit *transmitter* diintegrasikan ke dalam sebuah *enclosure* berbahan plastik hitam yang kompak, dengan mikrofon INMP441 diposisikan di balik lubang ventilasi sirkular untuk memastikan gelombang suara dapat ditangkap secara optimal tanpa hambatan fisik namun tetap terlindungi dari debu dan benturan.

A. Arsitektur FreeRTOS dan Manajemen Task

Pengembangan perangkat lunak sistem ini dibangun menggunakan Espressif IoT Development Framework (ESP-IDF) dengan memanfaatkan kernel FreeRTOS sebagai sistem operasi waktu nyata. Pemilihan ESP-IDF sebagai framework pengembangan didasarkan pada dukungan penuhnya terhadap seluruh fitur perangkat keras ESP32-S3, termasuk komponen ESP-SR untuk pengenalan suara dan antarmuka I2S untuk akuisisi audio. Berbeda dengan platform pengembangan berbasis Arduino yang menyembunyikan detail tingkat rendah, ESP-IDF menyediakan Application Programming Interface (API) berbasis bahasa C yang memungkinkan pengembang mengontrol seluruh fitur perangkat keras secara optimal, termasuk manajemen memori dan interaksi antar-inti prosesor.

Strategi manajemen multitasking diterapkan melalui mekanisme *task pinning* menggunakan fungsi xTaskCreatePinnedToCore(), yang memungkinkan penetapan afinitas inti (*core affinity*) secara eksplisit untuk setiap tugas. Sistem ini dikendalikan oleh enam task FreeRTOS yang dibagi ke dalam dua kelompok berdasarkan karakteristik beban kerjanya. Pembagian beban kerja ini dirancang secara strategis untuk mencegah proses inferensi AI yang bersifat komputasi intensif memblokir atau menunda eksekusi logika kendali robot yang menuntut responsivitas tinggi.

Core 1 didedikasikan sepenuhnya untuk menjalankan audio_processing_task dengan prioritas 5 dan alokasi memori tumpukan (*stack*) sebesar 8KB. Tugas ini bertanggung jawab atas seluruh pipeline inferensi kecerdasan buatan, mulai dari akuisisi data audio melalui I2S, pemrosesan sinyal awal oleh Audio Front-End (AFE) untuk peredaman derau (*noise suppression*), deteksi kata pemicu oleh model WakeNet, hingga klasifikasi perintah navigasi oleh model MultiNet7. Alokasi stack sebesar 8KB pada task ini merupakan yang terbesar dalam sistem, mengakomodasi kebutuhan buffer audio dan parameter model jaringan saraf yang dimuat ke dalam memori[9].

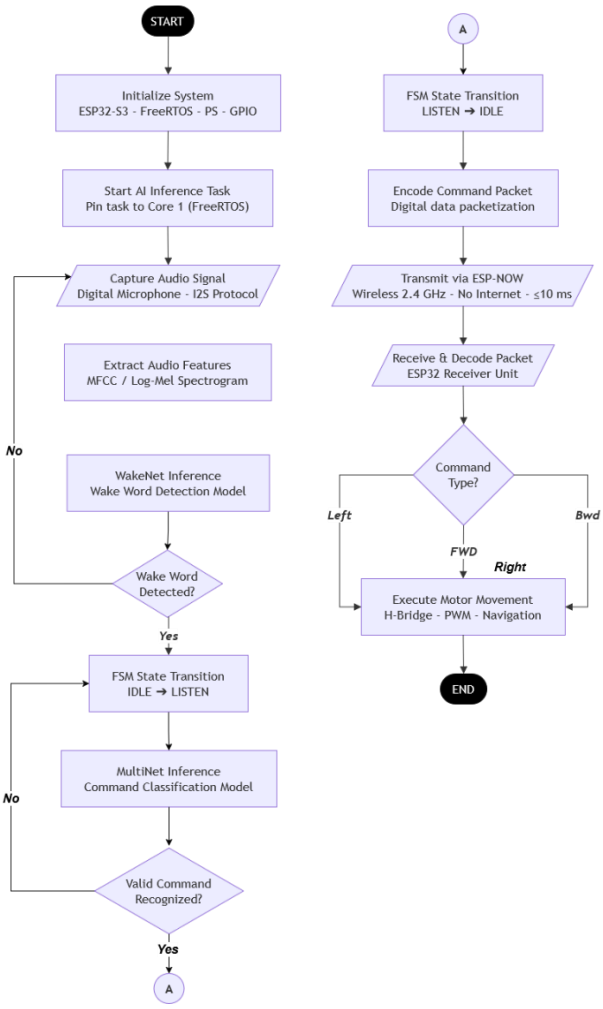
Sementara itu, Core 0 didedikasikan untuk menangani lima task yang bersifat *safety-critical* dan menuntut responsivitas tinggi. state_machine_task dengan prioritas 6 berperan sebagai pengendali logika utama sistem, menerima event dari Core 1 melalui mekanisme antrean FreeRTOS (*FreeRTOS Queue*) dan menentukan transisi status robot. button_input_task dan joystick_input_task masing-masing berjalan dengan prioritas 2 untuk menangani masukan pengguna secara manual. watchdog_feeder_task dengan prioritas 3 bertugas memantau kesehatan sistem secara berkala untuk mencegah kondisi *deadlock*, sedangkan status_led_task dengan prioritas 2 memberikan umpan balik visual kepada pengguna mengenai status operasional sistem saat ini.

B. *Pipeline Keyword Spotting dan FSM*

Inti dari sistem kendali suara ini adalah pipeline pengenalan kata kunci (*Keyword Spotting*) yang diimplementasikan menggunakan framework ESP-SR dari Espressif Systems. Pipeline ini beroperasi secara bertingkat dan terdiri dari empat tahap utama yang berjalan secara sekuensial pada Core 1. Alur kerja lengkap sistem dapat dilihat pada Gambar 2.

Tahap pertama adalah akuisisi dan pra-pemrosesan audio. Data audio mentah yang ditangkap oleh mikrofon INMP441 melalui antarmuka I2S diteruskan ke blok *Audio Front-End* (AFE) yang bertugas melakukan peredaman derau (*noise suppression*) secara real-time. Sinyal audio yang telah dibersihkan kemudian diekstraksi fiturnya menggunakan representasi *Mel-Frequency Cepstral Coefficients* (MFCC) dan *Log-Mel Spectrogram* untuk menghasilkan representasi frekuensi yang optimal bagi model jaringan saraf.

Tahap kedua adalah deteksi kata pemicu menggunakan model WakeNet. Model ini beroperasi secara kontinu dalam kondisi *idle* dengan konsumsi daya minimal, menganalisis aliran audio secara terus-menerus untuk mencari kecocokan dengan pola frekuensi kata pemicu "Hi ESP". WakeNet dirancang khusus untuk efisiensi daya sehingga dapat berjalan tanpa henti tanpa membebani sistem secara berlebihan.



Gambar 2. Diagram Alir Sistem

Tahap ketiga adalah klasifikasi perintah navigasi menggunakan model MultiNet7. Setelah kata pemicu berhasil terdeteksi, WakeNet secara otomatis dinonaktifkan dan sumber daya dialihkan sepenuhnya ke model MultiNet7. Model ini kemudian membuka jendela waktu terbatas untuk mendengarkan dan mengklasifikasikan perintah navigasi spesifik dari pengguna. Lima perintah yang didukung adalah *forward*, *backward*, *left*, *right*, dan *stop*. Validasi perintah sepenuhnya bergantung pada status internal model ESP_MN_STATE_DETECTED, bukan pada ambang batas *confidence score* mentah, sehingga menghasilkan responsivitas tinggi tanpa penyaringan ganda di sisi *firmware*.

Tahap keempat adalah manajemen state melalui Finite State Machine (FSM). Sistem FSM yang diimplementasikan terdiri dari sembilan state yang terdefinisi secara eksplisit: MODE_SELECT, IDLE, LISTENING, MOVING_FWD, MOVING_BWD, TURNING_LEFT, TURNING_RIGHT, STOPPED, dan ERROR. Hasil inferensi dari Core 1 tidak langsung

mengintervensi kendali motor, melainkan dikirimkan sebagai event ke state_machine_task pada Core 0 melalui FreeRTOS Queue. Fungsi process_event_internal() kemudian menentukan transisi state berdasarkan kombinasi state saat ini dan event yang diterima, menjamin perilaku sistem yang deterministik. Mekanisme *fail-safe* berupa *watchdog timer* dan *auto-recovery* ke STATE_STOPPED juga terintegrasi dalam FSM untuk menjaga keamanan operasional robot dalam segala kondisi.

C. Protokol Komunikasi ESP-NOW

Transmisi perintah navigasi dari unit *transmitter* ke unit *receiver* dilakukan menggunakan protokol ESP-NOW yang beroperasi pada frekuensi 2.4 GHz. Protokol ini dipilih karena sifatnya yang *connectionless*, memungkinkan pengiriman data secara langsung antar-perangkat (*peer-to-peer*) tanpa memerlukan proses *handshake* WiFi konvensional maupun infrastruktur *access point* eksternal[10]. Dibandingkan dengan protokol WiFi standar, ESP-NOW menawarkan latensi transmisi yang sangat rendah (±10ms)[11] yang merupakan kebutuhan krusial untuk aplikasi kendali robot secara *real-time*.

Setiap perintah navigasi yang telah divalidasi oleh FSM dikemas ke dalam struktur paket data berukuran 7 byte menggunakan direktif kompilasi __attribute__((packed)) untuk mencegah penyisipan *padding* memori otomatis. Struktur paket terdiri dari enam field: header (0xAA) sebagai penanda validitas awal paket, msg_type untuk identifikasi jenis pesan, command yang memuat kode instruksi navigasi (0=STOP, 1=FWD, 2=BWD, 3=LEFT, 4=RIGHT), speed untuk parameter kecepatan motor dalam rentang 0–255, duration_ms untuk durasi eksekusi perintah, serta checksum untuk verifikasi integritas data.

Untuk menjamin keandalan data selama transmisi nirkabel, setiap paket dilengkapi dengan nilai *checksum* yang dihitung menggunakan algoritma *Exclusive OR* (XOR) terhadap seluruh byte data kecuali byte *checksum* itu sendiri. Mekanisme ini memungkinkan unit penerima melakukan kalkulasi ulang dan memverifikasi bahwa paket tidak mengalami korupsi data selama proses transmisi. Proses pengiriman dieksekusi oleh fungsi espnow_send_command() yang menerapkan mekanisme *semaphore* dengan batas waktu (*timeout*) 100ms untuk memastikan sinkronisasi pengiriman dan mencegah kondisi *race condition* antar-task.

D. Prosedur Evaluasi Sistem

Evaluasi sistem dilaksanakan melalui dua pendekatan pengujian yang saling melengkapi. Pendekatan pertama adalah *Black Box Testing* yang digunakan untuk memastikan bahwa setiap perintah suara menghasilkan respons gerakan motor yang tepat sesuai skenario penggunaan yang telah didefinisikan [12]. Metode ini memvalidasi fungsionalitas sistem secara menyeluruh dari perspektif pengguna akhir tanpa meninjau struktur kode internal [14].

Pendekatan kedua adalah pengujian akurasi model AI yang mengukur performa sistem pengenalan kata kunci secara statistik. Pengujian dilakukan secara berulang terhadap lima label perintah navigasi dengan masing-masing perintah diuji sebanyak sepuluh kali percobaan, sehingga total keseluruhan percobaan berjumlah 50 kali. Untuk menguji ketangguhan model terhadap variasi karakteristik vokal manusia, pengujian melibatkan lima orang penguji yang berbeda dengan artikulasi yang jelas dan konsisten pada jarak efektif 15 hingga 20 sentimeter dari mikrofon INMP441 di lingkungan dalam ruangan dengan tingkat kebisingan sedang.

Keandalan sistem diukur berdasarkan tiga metrik evaluasi standar klasifikasi AI. *True Positive* (TP) didefinisikan sebagai kondisi sistem berhasil mendeteksi dan mengeksekusi perintah suara dengan benar. *False Positive* (FP) terjadi ketika sistem mendeteksi perintah padahal tidak ada ucapan valid yang diberikan. *False Negative* (FN) merujuk pada kegagalan sistem dalam mendeteksi perintah valid yang diucapkan pengguna. Berdasarkan ketiga variabel tersebut, tingkat akurasi sistem dihitung menggunakan persamaan 1.

$$Akurasi = \frac{TP}{TP + FP + FN} \; X \; 100\% \qquad (1)$$

Parameter *Confidence Score* turut dicatat sebagai indikator numerik dalam rentang 0.0 hingga 1.0 yang merepresentasikan tingkat keyakinan model terhadap hasil prediksi yang dibuat, namun tidak digunakan sebagai filter keputusan dalam arsitektur *firmware* yang dikembangkan.

III. HASIL DAN PEMBAHASAN

A. Fault Management

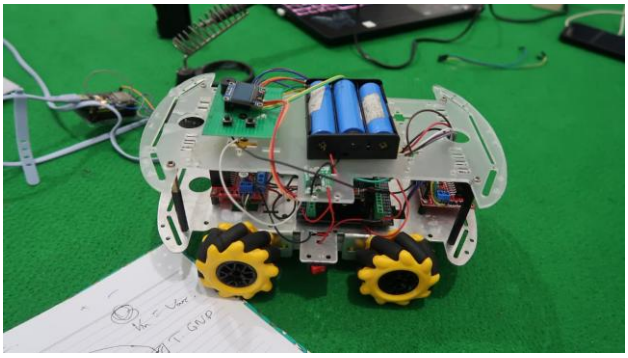
Implementasi sistem menghasilkan sebuah purwarupa robot 4WD yang dikendalikan oleh unit pengontrol cerdas berbasis ESP32-S3. Unit pengontrol tersebut berhasil mengintegrasikan seluruh komponen elektronik ke dalam sebuah *enclosure* berbahan plastik hitam yang kompak dan ergonomis. Mikrofon MEMS INMP441 diposisikan di balik lubang ventilasi sirkular pada bagian tengah *casing* untuk memastikan gelombang suara dari pengguna dapat ditangkap secara optimal tanpa hambatan fisik, namun tetap terlindungi dari debu dan benturan [13]. Hasil realisasi fisik unit *transmitter* dapat dilihat pada Gambar 3.

Sementara itu, realisasi fisik unit receiver berupa robot 4WD dengan empat roda dan motor driver terintegrasi dapat dilihat pada Gambar 4.

Pada sisi perangkat lunak, arsitektur dual-core FreeRTOS berhasil memisahkan beban kerja secara stabil sesuai rancangan. Core 1 menangani seluruh pipeline inferensi kecerdasan buatan tanpa gangguan, sementara Core 0 mengelola logika Finite State Machine dan komunikasi ESP-NOW secara bersamaan. Pemantauan sistem melalui IDF Monitor menunjukkan bahwa transisi status sistem, mulai dari deteksi wake word hingga klasifikasi perintah navigasi, berjalan sesuai dengan rancangan logika FSM tanpa adanya galat fatal (*fatal error*) atau *crash* selama operasional berlangsung. Proses interaksi yang terekam dalam log menunjukkan sistem

berhasil mencatat peristiwa WAKEWORD_DETECTED dan secara otomatis menonaktifkan WakeNet untuk mengalihkan sumber daya ke model MultiNet7 guna memproses instruksi selanjutnya.

Gambar 3. Unit Transmitter



Gambar 4. Realisasi Fisik Unit Receiver Robot 4WD

B. Analisa Performa Pengenalan Suara (Keyword Spotting)

Evaluasi kinerja model Edge AI dilakukan untuk mengukur tingkat keandalan sistem dalam mengenali instruksi navigasi secara luring. Hasil pengujian terhadap seluruh lima perintah navigasi dirangkum dalam Tabel 1.

Tabel 1. Rekapitulasi Pengujian Seluruh Perintah

Command	TP	FP	FN	Akurasi	Avg.confindece
Forward	10	0	0	100%	16.67%
Backward	10	0	0	100%	16.91%
Left	10	0	0	100%	17.59%
Right	10	0	0	100%	18.05%
Stop	10	0	0	100%	18.88%
Total	**50**	**0**	**0**	**100%**	**17.62%**

Berdasarkan Tabel I, sistem menunjukkan tingkat akurasi absolut sebesar 100% dari total 50 percobaan untuk seluruh variasi perintah. Tidak ditemukan satu pun kejadian *False Positive* maupun *False Negative* pada seluruh skenario pengujian. Keberhasilan pengenalan ini dipengaruhi oleh dua faktor utama. Pertama, pemilihan kosakata perintah navigasi dalam Bahasa Inggris yang memiliki struktur fonem dan silabel yang sangat berbeda satu sama lain, sehingga memudahkan model MultiNet7 dalam mengklasifikasikan fitur ekstraksi frekuensi suara berbasis MFCC. Kedua, kondisi pengujian yang dilakukan di lingkungan terkontrol dengan tingkat kebisingan sedang pada jarak optimal 15–20 sentimeter dari mikrofon.

Meskipun sistem mencapai akurasi sempurna, analisis terhadap *Confidence Score* menunjukkan rata-rata nilai sebesar 17.62%. Secara teoritis, nilai probabilitas mentah yang rendah ini dapat mengindikasikan keraguan model. Namun dalam implementasi ini hal tersebut sama sekali tidak mendegradasi tingkat keberhasilan eksekusi. Hal ini disebabkan oleh desain *firmware* yang memprioritaskan validasi mutlak dari status internal kerangka kerja ESP-SR (ESP_MN_STATE_DETECTED) sebagai satu-satunya pemicu eksekusi perintah, tanpa penyaringan ambang batas ganda di sisi *firmware*. Pendekatan ini terbukti efektif dalam mempertahankan responsivitas tinggi sekaligus menjaga akurasi sistem secara keseluruhan.

C. Efektivitas Komunikasi Nirkabel dan Responsivitas

Integrasi protokol ESP-NOW pada sistem ini terbukti efektif dalam meminimalisir latensi kendali antara unit pengolah suara dan unit robot. Struktur paket data 7-byte yang ringkas memungkinkan transmisi perintah dilakukan secara instan setelah proses inferensi AI selesai, dengan latensi transmisi yang terukur di kisaran ±10ms. Penggunaan mekanisme *checksum* XOR pada setiap paket data menjamin integritas instruksi yang diterima oleh unit robot, sehingga risiko korupsi data selama transmisi nirkabel dapat ditekan sepenuhnya. Hasil pengujian menunjukkan tidak ada satu pun paket yang gagal tervalidasi selama keseluruhan sesi pengujian berlangsung.

Secara kualitatif, kombinasi antara akselerasi vektor pada prosesor Xtensa LX7 untuk inferensi AI dan protokol komunikasi nirkabel *connectionless* ESP-NOW memberikan pengalaman interaksi manusia-mesin yang bersifat waktu nyata (*real-time*). Proses lengkap mulai dari ucapan pengguna hingga respons pergerakan robot berlangsung dalam rentang waktu kurang dari satu detik (*end-to-end latency* <1 detik), yang merupakan standar responsivitas yang sangat memadai untuk konteks demonstrasi akademik dan alat peraga pendidikan.

D. Kelayakan Sebagai Media Pembelajaran

Berdasarkan keseluruhan hasil implementasi dan pengujian, purwarupa sistem robot kendali suara berbasis Edge AI ini dinilai sangat layak sebagai media pembelajaran robotika modern di Educourse.id. Dari aspek keterjangkauan, penggunaan ESP32-S3 sebagai platform utama menawarkan rasio performa terhadap harga (*price-to-performance ratio*) yang jauh lebih baik dibandingkan platform eksisting berbasis BBC Micro:bit, tanpa mengorbankan kapabilitas AI yang menjadi fitur utamanya.

Dari aspek relevansi teknologi, sistem ini secara langsung mengekspos peserta didik pada konsep-konsep teknologi mutakhir seperti Edge AI, TinyML, real-time operating system, dan komunikasi nirkabel *peer-to-peer* dalam satu platform terintegrasi. Tingkat keberhasilan pengenalan suara yang absolut sebesar 100% memastikan bahwa proses pembelajaran tidak akan terhambat oleh frustrasi akibat kegagalan deteksi alat peraga. Selain itu, arsitektur sistem yang modular dan terdokumentasi dengan baik melalui mekanisme *serial logging* memudahkan instruktur dalam menjelaskan alur kerja sistem kepada peserta didik secara transparan dan terukur.

IV. KESIMPULAN

Penelitian ini telah berhasil merancang dan mengimplementasikan purwarupa sistem robot 4WD yang dikendalikan suara berbasis Edge AI menggunakan mikrokontroler ESP32-S3 sebagai alternatif media pembelajaran robotika yang lebih modern dan terjangkau di Educourse.id. Penerapan arsitektur *dual-core* dengan sistem operasi FreeRTOS terbukti efektif dalam menjaga stabilitas sistem melalui pemisahan tugas secara eksplisit (*task pinning*), di mana proses inferensi kecerdasan buatan pada Core 1 tidak mengganggu jalannya logika kendali Finite State Machine dan komunikasi nirkabel pada Core 0. Integrasi framework ESP-SR yang menggabungkan model WakeNet untuk deteksi kata pemicu dan MultiNet7 untuk klasifikasi perintah navigasi memungkinkan seluruh proses pengenalan suara berjalan sepenuhnya secara luring (*offline*) tanpa ketergantungan pada infrastruktur internet, menjadikan sistem ini andal untuk dioperasikan di berbagai kondisi lapangan.

Hasil evaluasi melalui *Black Box Testing* terhadap 50 percobaan yang melibatkan lima penutur berbeda menunjukkan bahwa sistem memiliki tingkat akurasi pengenalan perintah sebesar 100% untuk seluruh lima perintah navigasi (*forward*, *backward*, *left*, *right*, *stop*) di lingkungan dengan tingkat kebisingan sedang. Meskipun rata-rata *confidence score* berada pada angka 17.62%, mekanisme validasi internal berbasis status ESP_MN_STATE_DETECTED pada *firmware* terbukti mampu menjamin eksekusi perintah yang presisi dan responsif tanpa memerlukan penyaringan ambang batas ganda. Protokol ESP-NOW yang digunakan untuk transmisi perintah nirkabel berhasil memberikan latensi *end-to-end* kurang dari satu detik, memenuhi standar responsivitas yang dibutuhkan untuk aplikasi kendali robot secara *real-time* [15]. Secara keseluruhan, purwarupa ini telah memenuhi kriteria sebagai media pembelajaran robotika yang modern, terjangkau, dan relevan dengan tuntutan kurikulum berbasis kecerdasan buatan.

Untuk pengembangan di masa mendatang, beberapa aspek dapat ditingkatkan lebih lanjut. Pertama, pengujian ketahanan sistem perlu dilakukan pada lingkungan dengan tingkat kebisingan yang lebih ekstrem, seperti area luar ruangan atau lingkungan industri, untuk mengukur batas toleransi model terhadap gangguan suara. Kedua, ekspansi kosakata perintah dapat dilakukan dengan melatih model MultiNet pada bahasa Indonesia, mengingat ketersediaan model *pre-trained* saat ini masih terbatas pada Bahasa Inggris. Ketiga, dari sisi perangkat keras, perancangan *Printed Circuit Board* (PCB) kustom yang terintegrasi dapat menggantikan susunan kabel manual saat ini untuk meningkatkan keandalan mekanis, meminimalisir derau elektrik, serta mempersiapkan sistem untuk diproduksi sebagai modul alat peraga secara massal.

DAFTAR PUSTAKA

[1] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[2] P. Warden and D. Situnayake, *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. Sebastopol, CA: O'Reilly Media, 2019.

[3] M. Rusci and T. Tuytelaars, "On-Device Customization of Tiny Deep Learning Models for Keyword Spotting with Few Examples," IEEE Micro, vol. 43, no. 3, pp. 41–48, 2023.

[4] S. Gupta, U. Mamodiya, and A. J. A. Al-Gburi, "Speech Recognition-Based Wireless Control System for Mobile Robotics: Design, Implementation, and Analysis," Robotics, vol. 6, no. 3, p. 25, 2025.

[5] A. Alwarafy, K. A. Al-Thelaya, M. Abdallah, J. Schneider, and M. Hamdi, "A Survey on Security and Privacy Issues in Edge Computing-Assisted Internet of Things," IEEE Access, vol. 9, pp. 15001–15026, 2021

[6] Espressif Systems, *ESP32-S3 Technical Reference Manual*, Version 1.4, Espressif Systems, Shanghai, China, 2022. [Online].

[7] Espressif Systems, "ESP-SR Speech Recognition Framework," GitHub, 2022.

[8] R. Barry, *Mastering the FreeRTOS Real Time Kernel*. Real Time Engineers Ltd., 2016.

[9] C. R. Banbury et al., "Benchmarking TinyML Systems: Challenges and Direction," *arXiv:2003.04821*, 2020.

[10] D. Urazayev, A. Eduard, M. Ahsan, and D. Zorbas, "Indoor Performance Evaluation of ESP-NOW," in *Proc. IEEE SIST*, Astana, Kazakhstan, 2023, pp. 1–6.

[11] D. Eridani, A. F. Rochim, and F. N. Cesara, "Comparative Performance Study of ESP-NOW, Wi-Fi, Bluetooth Protocols based on Range, Transmission Speed, Latency, Energy Usage and Barrier Resistance," in Proc. IEEE ICIC, Gorontalo, Indonesia, 2021, pp. 1–6.

[12] F. Wagner, R. Schmuki, T. Wagner, and P. Wolstenholme, Modeling Software with Finite State Machines: A Practical Approach. Boca Raton, FL: Auerbach Publications, 2006..

[13] P. Horowitz and W. Hill, *The Art of Electronics*, 3rd ed. Cambridge University Press, 2015.

[14] R. S. Pressman, *Software Engineering: A Practitioner's Approach*, 8th ed. McGraw-Hill, 2015.

[15] P. Jakuš and H. Džapo, "Implementing Keyword Spotting on the MCUX947 Microcontroller with Integrated NPU," arXiv:2506.08911, 2025.