

Implementasi Sistem Multitasking pada Mobil RC Berbasis ESP32 dan FreeRTOS

Faid Rama Daniy¹, Tegar Desduido Hasbing Mulia², Abdurrakhman Hamid Al-Azhari³,
Djuniadi Djuniadi⁴

^{1,2,3,4} Universitas Negeri Semarang; Sekaran, Gunungpati, Semarang 50229, Jawa Tengah, Indonesia;
024-8508091

Received: 04-07-2025
Accepted: 25-07-2025

Keywords:
FreeRTOS;
RC Car;
ESP32;
Multitasking;
real-time system;

Correspondent Email:
faidramadaniy@students.unnes.ac.id

Abstrak. *Seiring dengan meningkatnya kebutuhan terhadap sistem tertanam yang mampu merespons secara real-time, penerapan Real-Time Operating System (RTOS) pada kendaraan kendali jarak jauh (RC Car) menjadi salah satu pendekatan yang relevan. Penelitian ini mengembangkan sistem RC Car berbasis mikrokontroler ESP32 dan FreeRTOS dengan dua mode operasi utama: mode manual yang dikendalikan melalui koneksi Bluetooth, serta mode autonomous yang menggunakan sensor ultrasonik HC-SR04 yang dipasang pada unit servo untuk melakukan pemindaian area di sekitarnya. Sistem juga mencakup kendali motor DC melalui driver L298N, serta pengelolaan sejumlah task secara paralel, termasuk pembacaan sensor, penggerak motor, tampilan visual pada OLED, dan sistem notifikasi menggunakan buzzer. Hasil pengujian menunjukkan bahwa sistem mampu menjalankan kedua mode dengan performa yang responsif, serta mendistribusikan beban multitasking secara efisien tanpa gangguan blocking. Temuan ini mengindikasikan bahwa FreeRTOS pada ESP32 merupakan platform yang andal dan efektif untuk pengembangan sistem kendali tertanam yang memerlukan pengolahan multitasking secara real-time.*

Abstract. *Along with the increasing need for embedded systems capable of responding in real-time, applying Real-Time Operating System (RTOS) in remote control vehicles (RC Car) is one of the relevant approaches. This research develops an RC Car system based on ESP32 microcontroller and FreeRTOS with two main operating modes: manual mode controlled via Bluetooth connection and an autonomous mode that uses HC-SR04 ultrasonic sensors mounted on servo units to scan the surrounding area. The system also includes control of a DC motor via an L298N driver and management of several tasks in parallel, including sensor reading, motor drive, visual display on an OLED, and notification system using a buzzer. The test results show that the system can run both modes with responsive performance and distribute the multitasking load efficiently without blocking interference. These findings indicate that FreeRTOS on ESP32 is a reliable and effective platform for developing embedded control systems that require real-time multitasking processing.*

1. PENDAHULUAN

Kemajuan teknologi sistem tertanam telah mendorong integrasi perangkat keras dan lunak dalam berbagai aplikasi, termasuk RC Car yang kini digunakan tidak hanya untuk

hiburan, tetapi juga untuk pengembangan sistem kendali otomatis dan studi sistem real-time [1]. Penggunaan mikrokontroler ESP32 sebagai inti sistem kendali real time juga telah diaplikasikan secara luas, termasuk dalam

pengembangan sistem drone yang menuntut respons cepat dan akurasi tinggi dalam pengendalian gerak [2].

Tantangan utama dalam pengembangannya adalah kebutuhan untuk menangani banyak fungsi secara paralel, seperti kendali motor, sensor, dan komunikasi nirkabel. Sistem bare-metal kurang efisien untuk multitasking dan penjadwalan tugas tepat waktu [3], sehingga RTOS menjadi solusi yang relevan karena menawarkan determinisme tinggi, multitasking, serta fitur seperti semaphore, mutex, dan queue. Tanpa RTOS, RC Car rentan terhadap blocking, race condition, dan kesulitan menangani interupsi [4]. RTOS memungkinkan tiap fungsi dijalankan sebagai task mandiri sesuai prioritas, meningkatkan stabilitas sistem. Mengingat keterbatasan memori dan prosesor, penting memilih RTOS ringan seperti FreeRTOS, Zephyr, atau ChibiOS [5].

Dalam konteks RC Car, respons real-time krusial untuk menghindari objek dan menerima perintah dengan cepat, dan RTOS membantu mengurangi latensi [6]. Selain itu, arsitektur modular RTOS mendukung ekspansi sistem ke fitur lebih kompleks seperti kamera, GPS, atau AI navigasi. RTOS bekerja dengan prinsip preemptive multitasking, memungkinkan eksekusi berbagai task secara bersamaan berdasarkan prioritas, yang diatur oleh scheduler sesuai event tertentu. RTOS akan membuat kinerja multitasking menjadi lebih efisien dan lebih hemat energi [7]. FreeRTOS sebagai salah satu RTOS populer menyediakan fitur manajemen task, komunikasi antar task, dan pengaturan waktu. Task dibuat menggunakan fungsi *xTaskCreate()*, di mana pengguna menentukan prioritas dan alokasi stack sesuai kebutuhan, sehingga mendukung pengembangan sistem yang modular.

Sinkronisasi antar task dilakukan menggunakan primitive seperti *semaphore* untuk pengendalian eksekusi dan *queue* untuk pengiriman data secara asinkron. Sistem dirancang berbasis event-driven melalui interupsi dan fungsi ISR yang mengaktifkan task terkait, sehingga lebih efisien dibanding polling terus-menerus [5]. Di sisi perangkat keras, proyek ini memanfaatkan mikrokontroler ESP32 yang kompatibel dengan RTOS. Penggunaan hardware abstraction layer (HAL) menyederhanakan integrasi dengan komponen

seperti motor driver, sensor ultrasonik, dan modul komunikasi. Proses pengujian dilakukan melalui teknik *runtime trace* dan *kernel-aware debugging* yang tersedia pada IDE seperti STM32CubeIDE atau ESP-IDF, untuk memantau status task, penggunaan stack, serta latensi interupsi selama sistem berjalan [4].

Penelitian ini bertujuan untuk merancang dan mengimplementasikan sistem RC Car real-time yang memanfaatkan RTOS untuk meningkatkan responsifitas dan modularitas sistem. Fokus utamanya adalah mengatasi tantangan multitasking dan sinkronisasi, merancang arsitektur perangkat lunak berbasis RTOS, dan mengintegrasikan perangkat keras seperti motor, sensor, dan modul komunikasi untuk evaluasi kinerja dalam skenario operasional yang sebenarnya.

2. TINJAUAN PUSTAKA

Sistem operasi waktu nyata atau *Real-Time Operating System* (RTOS) telah menjadi fondasi penting dalam pengembangan sistem tertanam modern, khususnya untuk perangkat yang memerlukan multitasking dan respons yang deterministik. FreeRTOS, sebagai salah satu RTOS sumber terbuka yang paling banyak digunakan, menawarkan efisiensi dalam pengelolaan task, penggunaan memori yang ringan, serta kemudahan dalam integrasi dengan berbagai platform mikrokontroler seperti ESP32. Penggunaan FreeRTOS terbukti memberikan peningkatan dalam hal stabilitas, skalabilitas, dan struktur program dibandingkan sistem konvensional berbasis loop tunggal, terutama pada aplikasi yang membutuhkan banyak fungsi berjalan secara paralel [8].

Dalam sistem mobil RC, multitasking menjadi kebutuhan penting, terutama saat sistem harus menangani berbagai proses seperti komunikasi nirkabel, kendali motor, serta deteksi rintangan secara simultan. FreeRTOS memungkinkan pemisahan fungsional ke dalam task independen, yang kemudian dijadwalkan berdasarkan prioritas maupun waktu tunda tertentu. Sistem berbasis FreeRTOS mampu mempertahankan performa kendali walaupun terdapat beban kerja tinggi pada task non-kritis, asalkan prioritas dan waktu tunggu dikelola dengan tepat. Hal ini penting dalam konteks kendaraan kendali jarak jauh, karena gangguan dari task sekunder seperti tampilan OLED tidak

boleh mengganggu respons sistem terhadap perintah kendali utama.

Komunikasi Bluetooth merupakan salah satu metode yang umum digunakan dalam pengendalian mobil RC secara manual. Modul HC-05 menjadi pilihan populer karena kemudahan integrasi dengan mikrokontroler serta protokol komunikasi serial yang sederhana. Selain itu, kombinasi antara ESP32 dan Bluetooth menawarkan fleksibilitas lebih tinggi berkat kemampuan bawaan ESP32 dalam menangani koneksi BLE maupun Classic Bluetooth. Sistem kontrol robotik berbasis Bluetooth memberikan latensi rendah serta kestabilan yang cukup untuk keperluan real-time control, terutama saat dikombinasikan dengan task prioritas tinggi di dalam RTOS. Sistem mobil RC dilengkapi dengan sensor jarak seperti HC-SR04 untuk mendukung mode autonomous. Sensor ini mampu mendeteksi keberadaan objek dalam jarak 2–400 cm dengan prinsip gelombang ultrasonik. HC-SR04 banyak digunakan dalam sistem robotik karena harganya terjangkau, mudah diimplementasikan, serta akurasinya cukup baik untuk navigasi dasar.

Dalam sistem berbasis FreeRTOS, pembacaan sensor ini dapat dijadikan satu task tersendiri yang berjalan periodik, dan hasilnya digunakan oleh task pengambilan keputusan untuk menghindari rintangan. Implementasi deteksi rintangan dengan HC-SR04 pada robot autonomous berbasis RTOS memungkinkan sistem berjalan lebih stabil karena proses pengukuran sensor dan pergerakan motor berjalan paralel tanpa saling mengganggu. Dengan menggabungkan FreeRTOS, komunikasi Bluetooth, dan sensor ultrasonik HC-SR04, sistem mobil RC dapat mendukung dua mode kendali secara efisien: mode manual berbasis perintah nirkabel dan mode autonomous berbasis data sensor. Penelitian-penelitian terbaru mendukung arsitektur semacam ini untuk keperluan pembelajaran, prototipe kendaraan cerdas, hingga simulasi sistem robotik nyata. Pendekatan ini juga mencerminkan praktik rekayasa sistem tertanam modern yang mengedepankan modularitas, efisiensi waktu eksekusi, serta ketahanan terhadap gangguan antar-task.

3. METODE PENELITIAN

Proyek ini menggunakan metode pengembangan sistem tertanam berbasis RTOS pada platform ESP32, yang dikenal memiliki kemampuan dual-core dan dukungan untuk FreeRTOS secara native. Penggunaan ESP32 memungkinkan pemrosesan multitugas dengan efisiensi tinggi, menjadikannya ideal untuk aplikasi RC Car yang memerlukan kendali simultan atas aktuator, sensor, dan komunikasi nirkabel [9]. Tahapan penelitian dimulai dari studi pustaka dan analisis kebutuhan sistem, diikuti oleh desain arsitektur perangkat keras dan lunak. Pengembangan perangkat lunak dilakukan dengan membagi fungsi utama (kendali motor, pembacaan sensor, komunikasi) ke dalam task-task terpisah menggunakan FreeRTOS. Setiap task dikembangkan dan diuji secara modular sebelum diintegrasikan secara penuh pada platform ESP32.

Desain pengujian mencakup evaluasi terhadap performa sistem, khususnya dalam hal responsivitas, kestabilan eksekusi task, dan efisiensi penggunaan sumber daya. Pengukuran dilakukan dengan bantuan fitur debug dan monitoring bawaan ESP32 serta perangkat lunak seperti Espressif Monitor dan Visual Trace Diagnostics [10]. Tahapan pengembangan dimulai dari studi literatur dan pemilihan RTOS yang sesuai. Setelah itu, dilakukan desain perangkat keras dan perangkat lunak. Kode program dibangun dengan memanfaatkan pustaka FreeRTOS pada ESP-IDF, dengan konfigurasi task menggunakan `xTaskCreate`, serta penggunaan queue dan semaphore untuk koordinasi antar task. Setiap task diuji secara individual sebelum dilakukan integrasi sistem.

Desain uji sistem meliputi pengujian fungsional untuk memverifikasi eksekusi task dan pengujian performa untuk menilai waktu respons dan efisiensi sistem. Evaluasi dilakukan menggunakan fitur debugging dan tracing pada ESP-IDF serta ekstensi seperti FreeRTOS Trace untuk memantau jalannya task secara real-time.

Komponen-komponen yang digunakan dalam proyek ini meliputi sebagai berikut.

3.1. *Doit ESP32*

ESP32 Doit DevKit V1 merupakan mikrokontroler berbasis arsitektur Xtensa dual-core 32-bit dengan frekuensi kerja hingga 240 MHz. Modul ini dilengkapi dengan RAM sebesar 520 KB dan memori flash hingga 4 MB. Kelebihannya terletak pada dukungan terhadap konektivitas Wi-Fi 802.11 b/g/n dan Bluetooth v4.2, serta tersedianya berbagai periferal seperti ADC, PWM, UART, SPI, dan I2C. Dalam proyek ini, ESP32 bertindak sebagai otak sistem, menjalankan FreeRTOS untuk membagi tugas seperti pengendalian motor, pembacaan sensor, komunikasi Bluetooth, dan pembaruan antarmuka OLED ke dalam beberapa task. Kemampuan multithreading-nya memungkinkan proses berjalan simultan dan efisien [9], [11].

3.2. *L298N (Motor Driver)*

L298N adalah driver motor dual-channel berbasis H-Bridge yang mampu menangani arus hingga 2A per kanal dengan tegangan operasi antara 5V–35V. Modul ini dilengkapi dengan pin enable dan input logika yang memungkinkan pengendalian arah dan kecepatan motor menggunakan sinyal PWM dari ESP32. Dalam proyek ini, L298N digunakan untuk mengontrol dua motor DC yang menggerakkan kendaraan. Fungsi utamanya adalah sebagai jembatan antara sinyal kontrol logika dari ESP32 dan kebutuhan daya tinggi dari motor, sekaligus memberikan perlindungan terhadap arus lebih dan panas berlebih [12].

3.3. *XL4015*

XL4015 merupakan modul buck converter berbasis chip XL4015 yang mampu menurunkan tegangan input hingga 36V menjadi output teratur 5V atau 3.3V dengan efisiensi konversi di atas 90%. Modul ini mendukung arus output hingga 5A dan dilengkapi dengan heatsink untuk mengatasi panas saat beban tinggi. Peran XL4015 dalam sistem adalah menstabilkan tegangan dari baterai (misalnya 2S Li-Ion 7.4V) agar sesuai dengan kebutuhan ESP32 dan L298N, sehingga

mencegah kerusakan akibat lonjakan tegangan [13].

3.4. *Motor Gearbox*

Motor DC yang digunakan adalah model mtr-0011 dengan gearbox rasio 1:48. Gearbox ini berfungsi menurunkan kecepatan rotasi motor sambil meningkatkan torsi. Dalam RC Car, motor ini dipasang pada roda penggerak dan diatur kecepatannya menggunakan sinyal PWM dari ESP32 yang diteruskan melalui L298N. Dengan rasio gear yang sesuai, kendaraan dapat melaju stabil dan responsif terhadap perubahan arah serta kecepatan yang dikontrol pengguna atau sistem otomatis [12], [14].

3.5. *HC-SR04*

HC-SR04 adalah sensor pengukur jarak berbasis ultrasonik yang bekerja dengan mengirimkan gelombang suara frekuensi tinggi (40 kHz) dan mengukur waktu pantulan (echo) dari objek di depan. Sensor ini memiliki akurasi sekitar ± 3 mm dan dapat mengukur jarak dari 2 cm hingga 400 cm. Dalam proyek ini, sensor digunakan untuk mendeteksi keberadaan rintangan di depan kendaraan. Hasil pengukuran dari sensor akan diproses oleh task khusus di ESP32 untuk mengaktifkan mode penghindaran otomatis jika objek terdeteksi dalam jarak tertentu [15], [17], [18].

3.6. *I2C OLED 128x64*

Layar OLED ini menggunakan teknologi self-emissive dengan resolusi 128x64 piksel dan komunikasi berbasis protokol I2C. Konsumsi dayanya rendah, dan ukurannya kompak (sekitar 0.96 inci). Layar OLED berfungsi menampilkan informasi real-time seperti mode operasi (manual/otomatis), jarak objek dari sensor, dan kecepatan kendaraan. Task terdedikasi dalam FreeRTOS bertanggung jawab memperbarui tampilan ini secara periodik tanpa mengganggu task lainnya, menggunakan semaphore untuk menghindari konflik akses data [16].

3.7. *Buzzer*

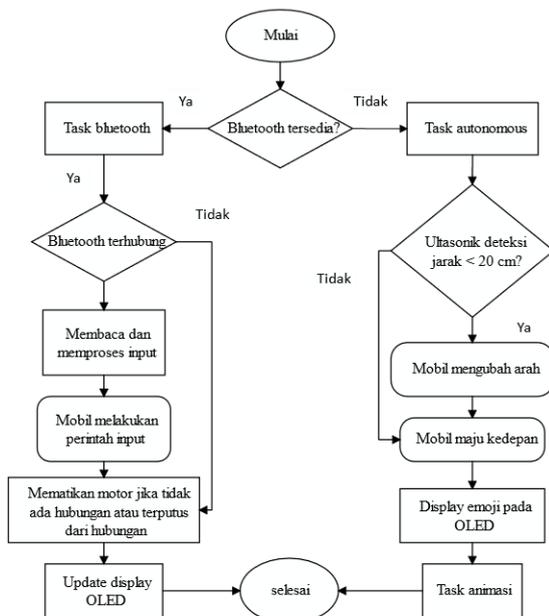
Buzzer yang digunakan dalam sistem ini merupakan buzzer aktif yang dapat dioperasikan pada tegangan 3 hingga 5 volt. Komponen ini berfungsi sebagai pemberi umpan balik suara yang menandai kondisi

tertentu dalam sistem, seperti saat kendaraan mendekati rintangan, terjadi kesalahan sistem, atau ketika sistem berpindah ke mode operasi tertentu. ESP32 mengontrol buzzer melalui salah satu pin digital, dan aktivasi dilakukan berdasarkan hasil evaluasi dari task yang berjalan. Meskipun perangkat ini bersifat sederhana, buzzer sangat membantu dalam memberikan informasi instan kepada pengguna selama proses pengujian atau saat sistem sedang beroperasi di lapangan [19].

4. HASIL DAN PEMBAHASAN

4.1. Diagram Alir Kerja Sistem

Diagram alir pada gambar 1 menunjukkan proses kerja RC Car berbasis ESP32 yang memanfaatkan FreeRTOS untuk menjalankan empat fungsi utama secara bersamaan: menerima sinyal dari Bluetooth, mengaktifkan mode otomatis berbasis sensor jarak, menampilkan informasi pada layar OLED, dan menjalankan animasi emoji. Setelah sistem dinyalakan, perangkat keras diinisialisasi, dan setiap task mulai berjalan secara paralel. Task Bluetooth bertugas memproses perintah gerakan, kecepatan, buzzer, dan pengaturan mode.



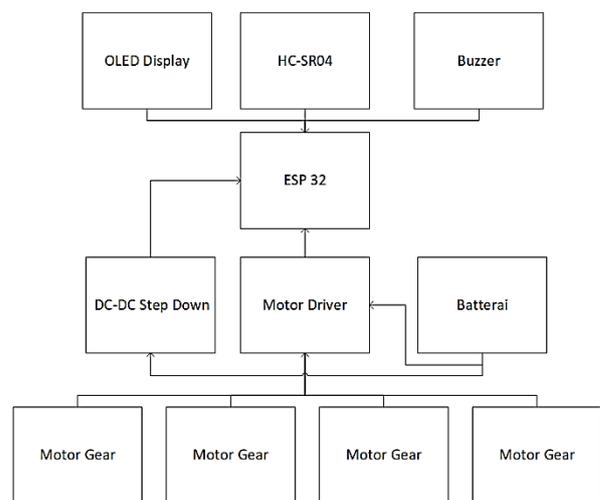
Gambar 1. Diagram Alir Sistem

Task autonomus akan mengatur arah mobil secara otomatis bila terdeteksi hambatan di depan. Informasi mode dan kecepatan

ditampilkan melalui OLED, sedangkan animasi emoji ditampilkan jika fitur tersebut diaktifkan. Seluruh proses berjalan secara simultan sehingga memungkinkan sistem merespons dengan cepat dan efisien tanpa terjadi konflik antar task.

4.2. Perancangan Diagram Alat

Gambar 2 Perancangan Diagram Alat, menunjukkan diagram dari mobil RC. Diagram blok tersebut menunjukkan keterkaitan antar komponen dalam sistem RC Car yang dikendalikan oleh ESP32.

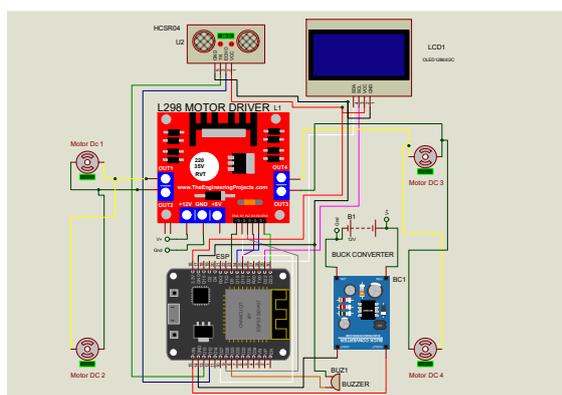


Gambar 2. Perancangan Diagram Alat

Mikrokontroler ESP32 berfungsi sebagai otak dari sistem, mengelola input dari sensor dan output ke aktuator. Sensor ultrasonik HC-SR04 digunakan untuk mengukur jarak benda di depan kendaraan, sementara layar OLED menampilkan status sistem seperti kecepatan dan mode pengoperasian. Buzzer berperan sebagai indikator suara berdasarkan perintah pengguna. Sinyal kontrol dari ESP32 diteruskan ke motor driver, yang kemudian menggerakkan empat motor gear untuk mobilitas kendaraan. Catu daya utama berasal dari baterai, dan untuk menyesuaikan tegangan yang dibutuhkan oleh ESP32 dan komponen lainnya, digunakan modul DC-DC step down. Seluruh komponen saling terintegrasi untuk memungkinkan pengoperasian RC Car secara manual maupun otomatis.

4.3. Perancangan Hardware

Komponen-komponen yang akan digunakan untuk membuat mobil RC disusun sesuai dengan skematik pada gambar 3. Gambar 4 menunjukkan susunan komponen yang telah dirangkai sesuai dengan skematik hardware dan telah dibentuk menjadi mobil. Sistem RC car berhasil diimplementasikan pada mikrokontroler ESP32 dengan dukungan FreeRTOS. Arsitektur perangkat lunak terdiri dari empat task utama: TaskBluetooth, TaskAutonomous, Task OLED, dan TaskAnimasi, yang berjalan secara multitasking dengan koordinasi melalui mekanisme semaphore dan flag global. Setiap komponen diuji secara fungsional untuk memastikan integrasi sistem berjalan tanpa konflik.



Gambar 3. Perancangan Hardware

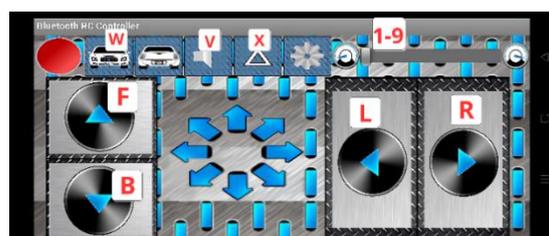


Gambar 4. Rangkaian Hardware

4.4. Kontrol RC

RC dikontrol menggunakan app dari appstore yang bernama “Bluetooth RC Control”, aplikasi ini menyediakan beberapa tombol dimana jika tombol ditekan akan mentransmisikan sinyal bluetooth, bentuk dari

sinyalnya berupa sebuah huruf alpabet sehingga mudah digunakan. Kotak teks yang berada dalam gambar 5 merepresentasikan sinyal yang ditransmisikan aplikasi. Tiap sinyal akan menjalankan memiliki peran masing masing dalam mengontrol RC, analisis tiap tombol akan dibahas dalam tabel 1.



Gambar 5. Apps Bluetooth RC controller

Tabel 1. Data Transmisi Sinyal Tiap Tombol

| No | Trasmisi sinyal | Perintah ke RC |
|----|-----------------|---------------------|
| 1 | S | Berhenti |
| 2 | F | Maju |
| 3 | B | Mundur |
| 4 | L | Kiri |
| 5 | R | Kanan |
| 6 | W | Tampil emoji LED |
| 7 | V | Mengaktifkan buzzer |
| 8 | X | Mode auto |
| 9 | 1-9 | Kontrol kecepatan |

4.5. Kecepatan dan Respon PWM

Kecepatan kendaraan dapat diatur dari level 1–9 melalui tombol 1–9, dengan masing-masing level menaikkan nilai PWM motor. Pengukuran dilakukan dengan menghitung waktu tempuh kendaraan pada lintasan lurus sepanjang 2 meter.

Tabel 2. Data Kecepatan dan Respon PWM

| NO | PWM | Waktu(s) | Kecepatan(m/s) |
|---------|-----|----------|----------------|
| default | 130 | 4 | 0,5 |
| 1 | 142 | 3 | 0,67 |
| 2 | 154 | 2,84 | 0,704 |
| 3 | 166 | 2,66 | 0,75 |
| 4 | 178 | 2,39 | 0,84 |
| 5 | 190 | 2,3 | 0,87 |
| 6 | 202 | 2,1 | 0,95 |
| 7 | 214 | 2,06 | 0,97 |
| 8 | 226 | 1,8 | 1,11 |
| 9 | 238 | 1,7 | 1,18 |

Hasil dari tabel 2 menunjukkan linearitas yang baik antara level PWM dan kecepatan kendaraan. Hal ini menandakan bahwa sistem PWM berhasil dikendalikan secara stabil.

4.6. Visualisasi OLED dan Animasi

Task TaskOLED secara periodik memperbarui informasi mode dan kecepatan pada bagian atas layar OLED, sedangkan TaskAnimasi menampilkan emoji pada bagian bawah. Task OLED diberi semaphore untuk mencegah konflik akses dari dua task yang berbeda. Meskipun pembaruan tampilan OLED membutuhkan waktu sekitar 300–400 ms, sistem tidak menunjukkan adanya gangguan pada task kendali motor maupun Bluetooth. Ini membuktikan efektivitas manajemen multitasking yang diberikan oleh FreeRTOS.

4.7. Kinerja Multitasking

Analisis multitasking dilakukan dengan mengukur durasi eksekusi loop masing-masing task. Tabel 3 menyajikan estimasi konsumsi CPU oleh masing-masing task:

Tabel 3. Data Performa Multitasking

| Task | Durasi | Delay | Estimasi load |
|---------------|--------|-------|---------------|
| taskOled | 36 ms | 500 | ~6,72 |
| taskbluetooth | 30 ms | 10 | ~61 |
| auto | 73 | 100 | ~42,2 |
| taskanimasi | 30 | 1000 | ~2,91 |

Analisis performa multitasking dilakukan dengan mengukur waktu eksekusi (durasi aktif) dan interval delay masing-masing task. Task Bluetooth menunjukkan konsumsi CPU tertinggi (~61%) karena interval delay yang sangat pendek (10 ms), diperlukan untuk menjaga responsifitas terhadap sinyal perintah dari pengguna. Task Autonomous menyusul dengan beban ~42,2%, mengindikasikan proses pembacaan sensor dan logika penghindaran rintangan cukup intensif. Task OLED dan Task Animasi menunjukkan beban CPU yang rendah, di bawah 10%. Hasil ini menunjukkan bahwa sistem multitasking FreeRTOS telah mampu menangani beban kerja secara paralel tanpa konflik signifikan, meskipun beban pada core dapat dioptimalkan lebih lanjut dengan penyesuaian delay atau penggunaan prioritas task.

Pengujian tambahan dilakukan untuk melihat dampak delay pada task non-kritis terhadap task utama. Delay besar sengaja dimasukkan ke dalam task OLED untuk menguji apakah akan terjadi gangguan pada

respon sistem terhadap perintah Bluetooth. Hasil menunjukkan bahwa task dengan delay panjang tidak mempengaruhi performa task dengan delay pendek, karena FreeRTOS menjadwalkan task sesuai prioritas dan waktu tunggu (vTaskDelay). Ini mengindikasikan bahwa sistem bebas gangguan (non-blocking) dan aman terhadap *task starvation*.

5. KESIMPULAN

Penelitian ini berhasil merancang dan mengimplementasikan sistem RC Car berbasis mikrokontroler ESP32 dengan dukungan FreeRTOS untuk manajemen multitasking secara efisien. Sistem mencakup dua mode operasi, yaitu manual melalui Bluetooth dan otomatis menggunakan sensor ultrasonik HC-SR04 yang dipasang pada servo untuk mendeteksi arah bebas hambatan. Dengan pembagian task yang jelas, masing-masing fungsi sistem dapat berjalan secara paralel dan stabil tanpa saling mengganggu. Penggunaan FreeRTOS terbukti efektif dalam menangani kebutuhan real-time dan mempercepat respons sistem terhadap masukan eksternal. Pengujian juga menunjukkan bahwa sistem dapat menangani skenario kehilangan koneksi secara andal melalui implementasi task timeout. Secara keseluruhan, integrasi FreeRTOS dalam proyek RC Car ini memberikan keunggulan dalam modularitas, skalabilitas, dan efisiensi eksekusi, menjadikannya solusi yang layak untuk sistem robotik tertanam yang kompleks.

UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada pengelola laboratorium teknik komputer UNNES yang telah memberi dukungan terhadap penelitian ini.

DAFTAR PUSTAKA

- [1] M. Yadav, "The Role of Embedded Systems in Robotics A Comprehensive Review," 2025. [Online]. Available: <http://www.ijert.org>
- [2] M. Dzaki *et al.*, "Pengembangan Mini Drone Berbasis ESP32 dengan Roll dan Pitch Control Menggunakan Algoritma PID," 2024.
- [3] A. Mohammad, R. Das, A. Islam, and F. Mahjabeen, "Real-time Operating Systems (RTOS) for Embedded Systems," *Asian Journal of Mechatronics, and Electrical*

- Engineering (AJMEE)*, vol. 2, no. 2, pp. 95–104, 2023, doi: 10.55927/ajmee.v2i2.7761.
- [4] G. , Al Azhar, S. , Sungkono, M. N. , Achmadiyah, and S. Izza, “Peningkatan Kestabilan Sistem Kontrol UGV melalui Optimalisasi Manajemen Core dan FreeRTOS pada ESP32.,” *Jurnal Elektronika Dan Otomasi Industri*, vol. 10, no. 2, pp. 256–263, 2023.
- [5] Sedat Sonko, Cosmas Dominic Daudu, Femi Osasona, Ayodeji Matthew Monebi, Emmanuel Augustine Etukudoh, and Akoh Atadoga, “The evolution of embedded systems in automotive industry: A global review,” *World Journal of Advanced Research and Reviews*, vol. 21, no. 2, pp. 096–104, Feb. 2024, doi: 10.30574/wjarr.2024.21.2.0420.
- [6] M. Bechtel, Q. Weng, and H. Yun, “DeepPicarMicro: Applying TinyML to Autonomous Cyber Physical Systems,” 2022.
- [7] F. Restuccia *et al.*, “ARTE: Providing real-time multitasking to Arduino,” *Journal of Systems and Software*, vol. 186, Apr. 2022, doi: 10.1016/j.jss.2021.111185.
- [8] D. Ramegowda and M. Lin, “Energy efficient mixed task handling on real-time embedded systems using FreeRTOS,” *Journal of Systems Architecture*, vol. 131, p. 102708, 2022, doi: <https://doi.org/10.1016/j.sysarc.2022.102708>.
- [9] J. Arm, O. Baštán, O. Mihálik, and Z. Bradác, “Measuring the Performance of FreeRTOS on ESP32 Multi-Core,” in *IFAC-PapersOnLine*, Elsevier B.V., 2022, pp. 292–297. doi: 10.1016/j.ifacol.2022.06.048.
- [10] W. Nakano, Y. Shinohara, and N. Ishiura, “Full Hardware Implementation of FreeRTOS-Based Real-Time Systems,” 2021.
- [11] Y. Tie, P. Chen, and Y. Ma, “BLE Bluetooth remote-controlled car based on ESP32,” *Applied and Computational Engineering*, vol. 81, no. 1, pp. 25–32, Nov. 2024, doi: 10.54254/2755-2721/81/20240965.
- [12] P. Peerzadaa, W. H. Larika, and A. A. Maharb, “DC Motor Speed Control Through Arduino and L298N Motor,” *International Journal of Electrical Engineering & Emerging Technology*, vol. 04, no. 02, pp. 21–24, 2021.
- [13] G. E. Moses and P. K. Ainah, “Implementation And Analysis of a 5v Rechargeable Power Supply for Microcontroller-Based Application,” *International Journal of Electrical, Electronics & Communication Engineering*, vol. 5, no. 2, pp. 12–19, 2024, doi: 10.5281/zenodo.13997511.
- [14] D. Revansa Arya Pradhana, A. Ichsan Pradhana, and D. Hartanti, “Implementasi Internet Of Things Pada Sistem Kontrol RC-car Pemantau Area Berbasis Esp32 cam,” *J Teknol*, vol. 11, no. 2, pp. 149–159, May 2024, doi: 10.31479/jtek.v11i2.306.
- [15] Z. Zakhir and H. Maulana, “Smart Blind Stick Design Using HC-SR04 Sensor and ESP 32 Based Water Level Sensor to Improve the Mobility of Blind Persons,” *Tsabit Journal of Computer Science*, vol. 1, no. 2, pp. 60–67, Jan. 2025, doi: 10.56211/tsabit32.
- [16] M. S. A. Siregar, U. K. Usman, and S. Astuti, “Pengembangan Sistem Parkir Pintar Berbasis IoT Menggunakan ESP32, Sensor Ultrasonik HC-SR04, dan OLED Display,” *eProceedings of Engineering*, vol. 11, no. 5, pp. 5553–5557, 2024.
- [17] V. S. N. Reddy, S. P. Kumar, B. Venkat, and J. S. Priyanka, “IoT based social distance checking robot using Esp32-Cam,” in *AIP Conference Proceedings*, American Institute of Physics Inc., Dec. 2021. doi: 10.1063/5.0074056.
- [18] M. R. Baldemor, “Simulation of Vehicle Distance Detection for Traffic Order,” *Data Science: Journal of Computing and Applied Informatics*, vol. 8, no. 1, pp. 37–44, Jan. 2024, doi: 10.32734/jocai.v8.i1-17139.
- [19] T. Patel, “IoT-BASED ULTRASONIC SENSORS AND BUZZER SYSTEMS IN HOME SECURITY By,” Maharastha, Dec. 2023.