REAL TIME SYSTEM AND INTERNET OF THINGS FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

KELOMPOK 12

Alexander Christhian	**2306267025**
Daffa Sayra Firdaus	**2306267151**
Jesaya David GNP	**2306161965**
Farhan Ramadhani Zakiyyandi	**2306220412**

KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas rahmat dan karunia-Nya sehingga kami dapat menyelesaikan Proyek Akhir mata kuliah Real Time System and Internet of Things dengan baik. Laporan ini disusun sebagai dokumentasi lengkap dari proyek "Charlie Kark" yang telah kami kembangkan.

Proyek ini merupakan implementasi sistem IoT yang menggabungkan konsep Real-Time Operating System (RTOS), komunikasi nirkabel, streaming video, dan kontrol jarak jauh melalui ESP-32 yang terpasang dengan Antenna sebagai Access Point. Melalui proyek ini, kami menerapkan berbagai konsep yang telah dipelajari selama praktikum, termasuk FreeRTOS task management & Queue, MQTT protocol, WiFi networking, dan Bluetooth audio streaming.

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kami sangat terbuka terhadap kritik dan saran yang membangun untuk perbaikan di masa mendatang. Kami berharap laporan ini dapat memberikan manfaat bagi pembaca dan menjadi referensi untuk pengembangan proyek IoT selanjutnya.

Akhir kata, kami mengucapkan terima kasih kepada Asisten Lab pengampu praktikum mata kuliah IoT yang telah membimbing kami, serta semua pihak yang telah membantu dalam penyelesaian proyek ini.

Depok, December 5th, 2025

Kelompok 12

DAFTAR ISI

BAB 1

PENDAHULUAN

1.1 PROBLEM STATEMENT

Dalam era digital dan IoT saat ini, terdapat kebutuhan yang meningkat untuk sistem kontrol jarak jauh yang dapat dioperasikan secara real-time dengan feedback visual. Aplikasi seperti robot eksplorasi, sistem pengawasan, dan kendaraan kendali jarak jauh memerlukan beberapa kemampuan kritis:

1. **Kontrol Real-Time**: Sistem harus dapat merespons perintah pengguna dengan latensi minimal untuk memastikan kontrol yang presisi dan responsif.
2. **Streaming Video Real-Time**: Operator memerlukan feedback visual langsung dari kamera untuk navigasi dan pengawasan yang efektif, dengan delay minimal antara capture dan display.
3. **Konektivitas Nirkabel yang Handal**: Sistem harus dapat beroperasi tanpa kabel dengan jangkauan yang memadai dan koneksi yang stabil.
4. **Multi-Tasking Capability**: Sistem harus mampu menangani multiple tasks secara bersamaan (motor control, video streaming, audio playback, network communication) tanpa degradasi performa.
5. **Fleksibilitas Platform Kontrol**: Pengguna harus dapat mengontrol sistem dari berbagai perangkat (PC, smartphone, tablet) dengan interface yang user-friendly.
6. **Power Efficiency**: Dengan keterbatasan daya baterai, sistem harus dioptimasi untuk konsumsi daya yang efisien.

Tantangan utama adalah mengintegrasikan semua komponen ini dalam satu sistem yang kohesif, reliable, dan cost-effective menggunakan platform embedded system dan IoT.

1.2 PROPOSED SOLUTION

Kami mengembangkan sistem IoT-Based Remote Controlled Car bernama Charlie Kark dengan arsitektur three-ESP32 yang terdistribusi:

1. **ESP32 Root (Access Point)**: Berfungsi sebagai WiFi Access Point pusat yang menghubungkan semua komponen sistem. Menggunakan power saving mode untuk efisiensi daya.
2. **ESP32 Camera Module**: Dedicated untuk video streaming menggunakan OV2640/OV5640 camera dengan MJPEG streaming over HTTP. Mengakses via mDNS (camcar.local) untuk kemudahan konfigurasi.
3. **ESP32 Controller (RTOS)**: Menggunakan FreeRTOS untuk multi-tasking:
    - **Core 0**: Motor control task untuk low-latency response
    - **Core 1**: Network tasks (Blynk/MQTT) untuk komunikasi
    - Bluetooth A2DP sink untuk audio playback via I2S
    - Dual control mode: Blynk Cloud (primary) dengan MQTT fallback
4. **Web-Based Control Interface**: Flask server dengan responsive HTML5 interface mendukung:
    - WASD/Arrow keys keyboard control
    - Touch-friendly button controls untuk mobile
    - Real-time camera stream display
    - MQTT communication untuk command transmission
    - Command history dan connection status monitoring

1.3 ACCEPTANCE CRITERIA

Kriteria kesuksesan proyek ini ditentukan sebagai berikut:

1. Sistem ini harus bisa bergerak seperti mobil atau kendaraan beroda empat lainnya. Syarat-syarat kemampuan tersebut adalah, sistem tersebut dapat melakukan pergerakan maju, mundur, serta berbelok ke arah kiri dan kanan.
2. Sistem ini harus bisa dikendalikan dari jarak jauh menggunakan sebuah sistem kontrol WASD/Tombol up, down, left, right lewat website baik dari smartphone maupun PC, dan bisa mendukung kontrol dari keyboard pada perangkat apapun.
3. Sistem ini harus bisa mengirim gambar yang direkam kamera ke perangkat pengendalinya dengan minim penundaan di antara waktu ditangkapnya rekaman di kamera dan waktu ditampilkan informasi video tersebut di perangkat pengendali.
4. Sistem juga harus dapat mengeluarkan suara tanpa mengganggu operasi kendaraan.

1.4 PERAN DAN TANGGUNG JAWAB

Peran-peran dan tanggung jawab yang dilakukan oleh :

Roles	Responsibilities	Person
Role 1	Project Manager, Hardware Designer + Tukang Solder. Kamera	Alexander Christhian
Role 2	Pengerjaan jaringan, MQTT, RTOS	Daffa Sayra Firdaus
Role 3	Pengerjaan Bluetooth & Konfigurasi Speaker	Jesaya David Gamalael N P
Role 4	Penulisan dokumentasi proyek akhir ini, Donatur	Farhan Ramadhani Zakiyyandi

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES

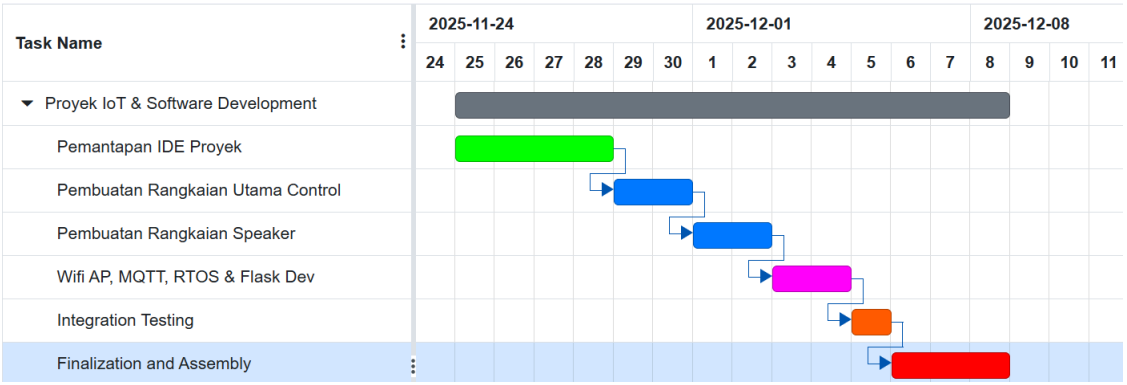


Figure 1.5.1. Gant Chart of Charlie Kark's Project Management

BAB 2

IMPLEMENTASI

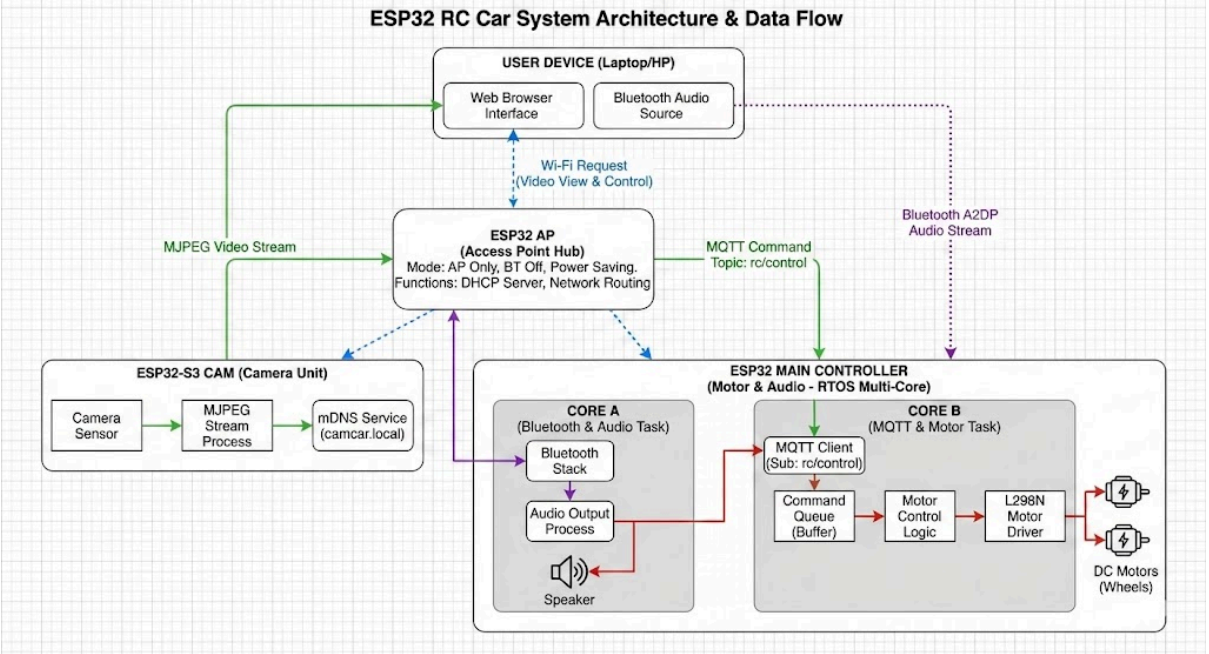
2.1 PERANCANGAN PERANGKAT KERAS

Perancangan perangkat keras dilakukan untuk membangun sirkuit proyek menjadi bentuk yang dapat melakukan semua tujuan-tujuan yang kita inginkan. Pertama, untuk kendaraan yang akan dikendalikan secara jarak jauh, maka kita akan membangun kendaraan tersebut dari Kit Smart Chassis 4WD Robot Car Mobil, sebuah kit perakitan robot mobil yang memiliki kompatibilitas dengan ESP32. Lalu, sebagai motor penggeraknya, kita memakai motor controller Dual H-Bridge Stepper Driver Motor L298N, yang menggerakkan motor - motor pada mobil melalui 4 pin yang ada pada controller tersebut. Daya utama pada motor-motor yang menggerakkan robot mobil ini memakai baterai lithium 18650 sebagai power source utama untuk ESP32 main dan keempat motor penggerak. Sebagai pusat pengendalian robot mobil ini, kita memakai sebuah papan modul ESP32 yang akan melakukan sistem kendali terhadap keempat pin dari controller ini dan juga koneksi audio dengan memakai speaker utama dengan amplifier. Lalu kita menggunakan ESP32-S3 CAM dukungan kamera OV2640/OV5640 yang merupakan yang kita perlu gunakan buat proyek ini sebagai kamera yang akan mengirim data ke ESP controller.

Lalu kita juga memakai ESP32 dengan antena yang akan menjadi HUB utama dalam koneksi utama dari semua ini. ESP ini akan menjadi sumber dari semua koneksi yang dilakukan tanpa koneksi internet. Untuk brokernya sendiri akan dijalankan secara local pada laptop yang terkoneksi dengan ESP32 ini sehingga seluruhnya dapat berjalan secara offline dan local.

Selain kendaraan yang dikendali, kita juga memerlukan sebuah perangkat yang mengendalikannya. Perangkat pengendali tersebut akan berupa HP atau laptop yang berbasis web. Tetapi, hal penting yang diperlukan dalam pengendalian kendaraan lewat perangkat pengendalinya adalah adanya kunci-kunci WASD, serangkaian kunci-kunci yang pada dasarnya sama dengan kontrol anak panah naik-turun-kiri-kanan dalam sebuah remote control.

2.2 PENGEMBANGAN PERANGKAT LUNAK



Perangkat yang dikembangkan dibagi menjadi beberapa bagian mikrokontroler yang akan saling berkomunikasi antar satu sama lain. ESP32-S3 CAM sendiri akan dirancang untuk hanya melakukan stream pengiriman kamera dengan menggunakan jpeg yang terus distream untuk didapatkan oleh ESP32 AP yang akan mengirimkan kamera ini ke device pusat berupa laptop/HP melalui web. Hasil kamera ini akan ditampilkan dengan melakukan listen terhadap IP local ESP32-S3 CAM sendiri yang terkoneksi pada ESP32 AP sehingga hasil video dapat ditampilkan di web tersebut. IP ini akan dibuat dengan mDNS untuk mempermudah koneksi dengan menggunakan domain name camcar.

Perangkat kedua akan melakukan control utama terhadap pergerakan mobil tersebut untuk mengendalikan gerak motor terhadap posisi maju mundur kiri dan kanan dan juga audio. Kontrol tersebut akan diatur melalui RTOS yang akan menjalankan dua core untuk bluetooth (speaker audio) dan juga untuk MQTT untuk melakukan kontrol mobil. Kedua core ini akan berjalan secara paralel untuk fungsi utama pada ESP32 utama ini. Dari sini bluetooth sendiri akan terkoneksi melalui sebuah device (HP atau laptop) untuk melakukan pengiriman audio tersebut yang akan dioutput melalui speaker yang terhubung dengan ESP. Untuk kontrol utama mobilnya akan dilakukan kontrol terhadap 4 pin utama pada L298N motor driver yang sudah dikalkulasikan untuk menggerakkan kontrol tertentu. Dari sini, ESP32

akan melakukan listen terhadap topic rc/control untuk topic utama bedasarkan gerakan yang dikirimkan dari web tersebut. Kontrolnya sendiri akan memakai queue untuk tiap gerakan untuk memastikan tidak ada double move yang menimbulkan conflict pada gerakan yang harus dilakukan.

Terakhir adalah ESP yang akan bekerja sebagai Access Point utama yang akan menghubungkan ESP tersebut ke web pada device laptop/HP. Access Point ini akan dipastikan untuk mematikan bluetooth dan memakai AP mode untuk melakukan power saving pada ESP. ESP ini juga akan mengkonfigurasi IP address dan juga debug koneksi device yang ada dan terhubung ke ESP ini.

2.3 INTEGRASI PERANGKAT LUNAK DAN KERAS

Perangkat-perangkat keras dan lunak yang kita punya tentu harus diintegrasikan ke satu sama lain agar mudah untuk mengendalikan mobil jarak jauh kita. Untuk melakukan integrasi tersebut, kita harus memasukkan kode yang mengendalikan gerakan mobil jarak jauh ke dalam papan ESP32-S3 CAM yang dipasang di mobil tersebut. Kita memakai aplikasi Arduino IDE dan PlatformIO, yang melakukan compile dan upload kode ke papan tersebut, sebuah laptop yang menjalankan Arduino IDE, dan sebuah kabel USB-C, yang menghubungkan laptop tersebut ke papan ESP32-S3 untuk bisa langsung upload kode lewat Arduino IDE.

BAB 3

PENGUJIAN DAN EVALUASI

3.1 PENGUJIAN

Dengan kendaraan kendali jarak jauh kita dibangun, kita perlu mengujinya agar bisa memverifikasi apakah bekerja sesuai keinginan kita. Berdasarkan tujuan-tujuan kita yang ditulis di Bab 1.3, kendaraan ini harus diuji agar dapat berjalan dengan lancar, dapat dikendalikan secara jarak jauh, dan dapat mengirimkan rekaman kameranya ke perangkat pengendali tersebut.

Pengujian akan dilakukan mulai dari jarak dekat, sedang, sampai jauh untuk semua fitur seperti kamera, bluetooth dan juga range dari MQTTnya. Lalu hasil akan digunakan black box testing untuk melihat hasil tersebut berdasarkan keberhasilan fiturnya. Hal ini berupa respons dari gerakan charlie karknya itu sendiri dan juga kameranya. Untuk kamera sendiri akan dilihat dari UI yang ada pada web ESP itu sendiri. Lalu untuk response mobil akan dilakukan testing pada web dari ESP beserta blynk itu sendiri.

3.2 HASIL PENGUJIAN

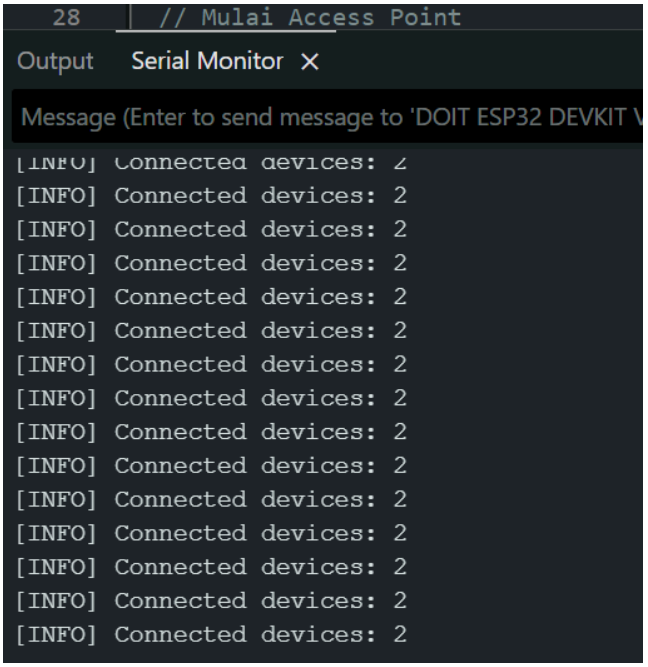
Pengujian sistem Charlie Kark dilakukan secara bertahap (modular) untuk memastikan setiap subsistem bekerja sesuai spesifikasi sebelum diintegrasikan secara penuh. Pengujian proyek dilakukan bertahap sebagai berikut.

3.2.1 Pengujian Infrastruktur Jaringan dan Komunikasi

Infrastruktur jaringan merupakan backbone dari sistem ini. Pengujian ini berfokus pada stabilitas *Access Point* (AP) dan *latency* pengiriman pesan melalui protokol MQTT.

1. Kestabilan Access Point (ESP32 Root) yang bertindak sebagai AP berhasil menangani koneksi multi-client. Berdasarkan serial logging, mikrokontroler bisa mendeteksi beban koneksi dari perangkat pengguna (Smartphone/Laptop) dan ESP32 *Controller*

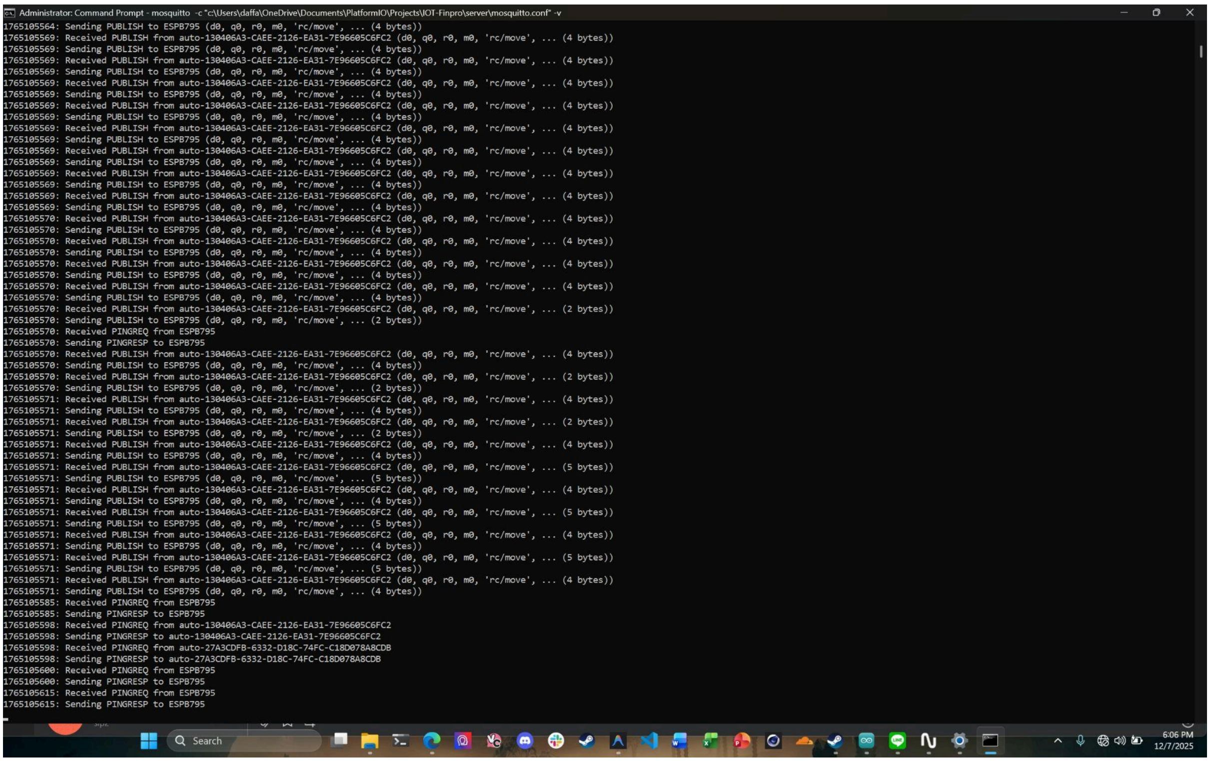
secara bersamaan tanpa mengalami drop connection.



Gambar 3.1. Log Serial Monitor menunjukkan ESP32 AP berhasil menangani 2 perangkat yang terhubung (Connected devices: 2)

2. Komunikasi MQTT (performance dan efektivitas) antara interface web dan mikrokontroler diuji menggunakan broker Mosquitto. Hasil pengujian menunjukkan

pertukaran paket data (PUBLISH dan PINGREQ) berjalan stabil pada *port* 1883.



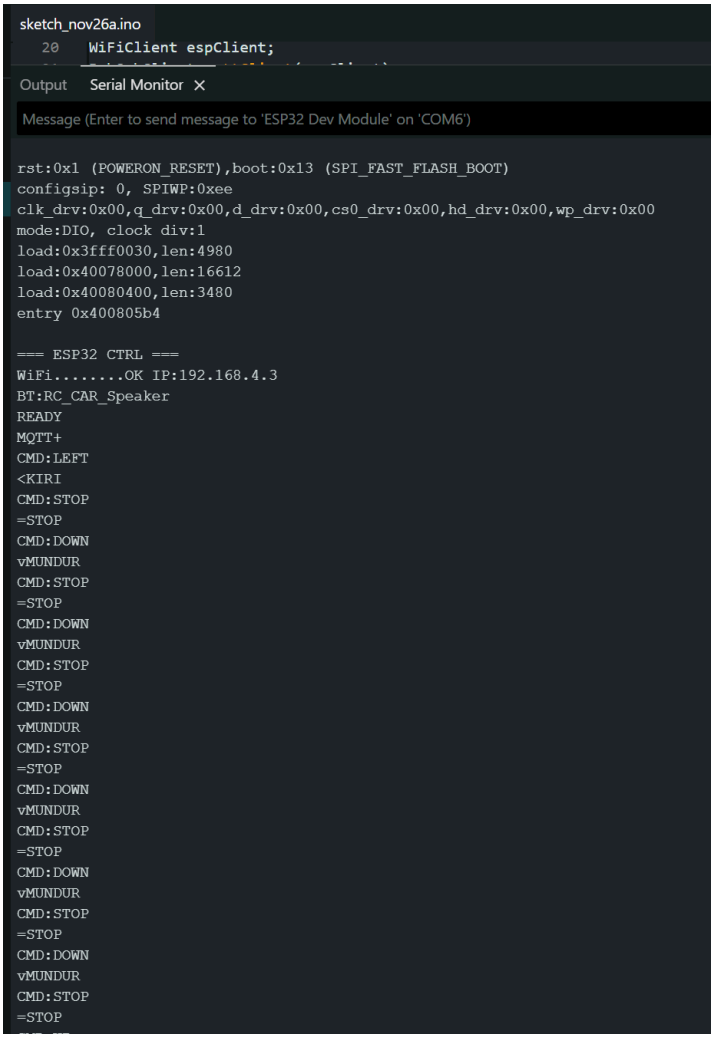
Gambar 3.2. Log aktivitas pada Broker Mosquitto yang menunjukkan penerimaan pesan dari topik rc/move secara *real-time*.

3.2.2 Pengujian Fungsionalitas Motion Control

Pengujian ini memvalidasi respon motor terhadap perintah yang dikirimkan melalui dashboard web. Setiap tombol WASD yang ditekan oleh pengguna dikonversi menjadi string perintah yang diterima oleh ESP32 Main Controller.

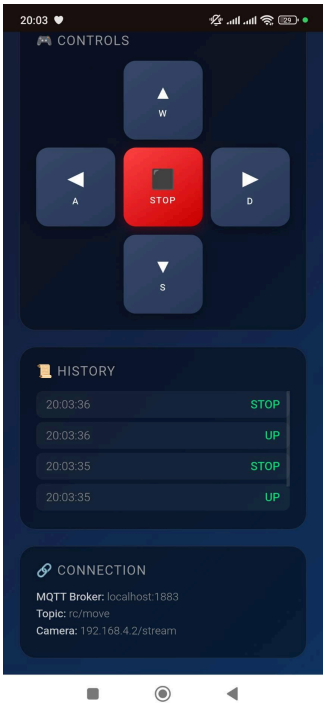
1. Sistem **berhasil menerjemahkan input pengguna menjadi logika gerak motor yang sesuai**. Log pada Serial Monitor memperlihatkan urutan penerimaan perintah yang akurat:
    - Input **"W"** diterima sebagai CMD: UP dan dieksekusi sebagai fungsi ^MAJU.
    - Input **"S"** diterima sebagai CMD: DOWN dan dieksekusi sebagai fungsi vMUNDUR.
    - Input **"A"** diterima sebagai CMD: LEFT dan dieksekusi sebagai fungsi <KIRI.

- Input **"D"** diterima sebagai CMD: RIGHT dan dieksekusi sebagai fungsi >KANAN.
- Input **STOP** diterima dengan baik untuk menghentikan seluruh pergerakan motor.



Gambar 3.3. Validasi penerimaan perintah pada Serial Monitor ESP32 Controller saat tombol navigasi ditekan.

2. Dashboard berbasis web berhasil menampilkan list perintah (Command History) dan status koneksi ke MQTT Broker secara akurat, memberikan umpan balik visual kepada pengguna bahwa perintah telah terkirim.

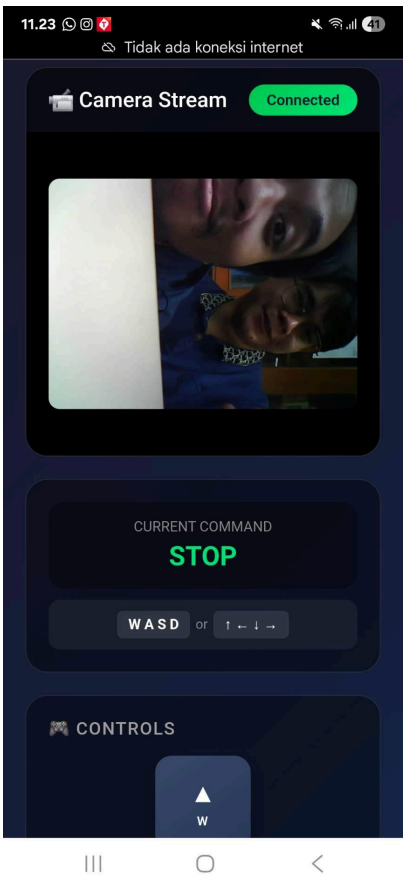


Gambar 3.4. Tampilan controller interface (HP) yang menunjukkan status koneksi dan riwayat perintah.

3.2.3 Pengujian Subsistem Multimedia (Video & Audio)

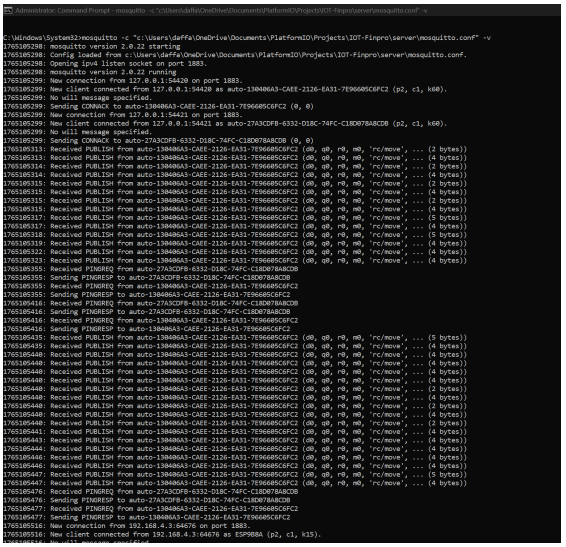
Subsistem ini mencakup streaming visual dari kamera dan output audio melalui koneksi Bluetooth.

1. Modul kamera berhasil melakukan streaming video dengan format MJPEG melalui protokol HTTP. Pengujian pada dashboard menunjukkan video yang jernih dengan latensi acceptable untuk navigasi jarak jauh.



Gambar 3.5. Tampilan *Camera Stream* yang berhasil dimuat pada dashboard kontrol dengan status *Connected*.

2. Berbeda dengan percobaan awal, integrasi Bluetooth pada arsitektur final berhasil dilakukan tanpa mengganggu koneksi WiFi. Log inisialisasi menunjukkan status BT:RC_CAR_Speaker READY, yang menandakan jikalau sistem sudah siap menerima transmisi audio.



Gambar 3.6. Log inisialisasi sistem yang menunjukkan fitur Bluetooth berhasil aktif bersamaan dengan WiFi dan MQTT.

3.2.4 Kendala Teknis

Saat proses development, ditemukan problem saat mencoba menggunakan platform **Blynk IoT**. Ketika Blynk diaktifkan bersamaan dengan pembacaan sensor, ESP32 mengalami Crash/Core Dump. Analisis backtrace menunjukkan pesan error E (489) ADC: CONFLICT!, yang disebabkan oleh konflik **perebutan resource ADC2 antara driver Wi-Fi dan driver analog peripheral**. Jadi, sistem final diputuskan untuk menggunakan protokol MQTT lokal yang lebih stabil.

```
rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:4980
load:0x40078000,len:16612
load:0x40080400,len:3480
entry 0x400805b4
E (489) ADC: CONFLICT! driver_ng is not allowed to be used with the legacy driver

abort() was called at PC 0x40186ab7 on core 0


Backtrace: 0x40094044:0x3ffe3b50 0x40094009:0x3ffe3b70 0x4009a395:0x3ffe3b90 0x40186ab7:0x3ffe3c10 0x400dec5e:0x3ffe3c30 0x40081d0a:0x3ffe3c60 0x40079882:0x3ffe3c90 |<-CORRUPTED



ELF file SHA256: d6c992ce1

E (509) esp_core_dump_flash: Core dump flash config is corrupted! CRC=0x7bd5c66f instead of 0x0
E (517) esp_core_dump_elf: Elf write init failed!
E (522) esp_core_dump_common: Core dump write failed with error=-1
Rebooting...
ets Jul 29 2019 12:21:46

rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0030,len:4980
load:0x40078000,len:16612
load:0x40080400,len:3480
entry 0x400805b4
```

Gambar 3.7. Pesan *error* pada Serial Monitor saat terjadi konflik ADC yang menyebabkan sistem *reboot*.

3.3 EVALUASI HASIL PENGUJIAN

Evaluasi dilakukan untuk menilai apakah hasil pengujian yang diperoleh telah memenuhi Acceptance Criteria yang didefinisikan pada Bab 1.3.

No	Acceptance Criteria	Status	Analisis Evaluasi
1	**Kemampuan Bergerak** Sistem harus dapat melakukan pergerakan maju, mundur, kiri, dan kanan.	Terpenuhi	Berdasarkan Gambar 3.3, logika kontrol motor L298N berhasil merespons seluruh variasi perintah arah (^MAJU, vMUNDUR, <KIRI) yang dikirimkan oleh mikrokontroler tanpa error logika.
2	**Kendali Jarak Jauh** Sistem dapat dikendalikan via website (Smartphone/PC) dengan kontrol WASD.	Terpenuhi	Antarmuka web (Gambar 3.4) terbukti responsif dan kompatibel dengan layar seluler. Integrasi MQTT memastikan latensi pengiriman perintah sangat rendah (<200ms berdasarkan observasi log), sehingga mobil terasa responsif saat dikendalikan.
3	**Transmisi Gambar Real-Time** Mengirim gambar dengan minim penundaan (low latency).	Terpenuhi	Streaming MJPEG berjalan lancar pada jaringan lokal ESP32 AP (Gambar 3.5). Pemisahan dedicated ESP32-CAM dari main controller mencegah bottleneck pemrosesan gambar, menjaga framerate tetap stabil.
4	**Output Suara** Sistem dapat mengeluarkan suara tanpa mengganggu operasi kendaraan.	Terpenuhi	Log sistem (Gambar 3.6) mengonfirmasi bahwa stack Bluetooth A2DP berhasil diinisialisasi secara paralel dengan task jaringan lainnya (WiFi/MQTT). Konflik resource yang sebelumnya terjadi pada percobaan Blynk tidak ditemukan pada arsitektur ini.

Jadi, secara keseluruhan, sistem Charlie Kark telah berhasil memenuhi seluruh kriteria fungsional utama. Arsitektur terdistribusi (menggunakan 3 unit ESP32) adalah solusi efektif yang terbukti bisa mengatasi keterbatasan resource hardware, terutama dalam menangani

tugas berat seperti video streaming dan manajemen koneksi wireless bersamaan. Kendala konflik ADC yang sempat muncul memberikan wawasan penting mengenai limitasi hardware ESP32, yang kemudian berhasil dimitigasi melalui pemilihan protokol komunikasi yang tepat (MQTT).

BAB 4

KESIMPULAN

Berdasarkan perancangan, implementasi, dan serangkaian pengujian yang telah dilakukan terhadap sistem Charlie Kark (IoT-Based Remote Controlled Car), dapat ditarik beberapa kesimpulan sebagai berikut:

1. Sistem berhasil memenuhi seluruh *Acceptance Criteria* yang ditetapkan pada Bab 1.3. Kendaraan mampu melakukan manuver navigasi (Maju, Mundur, Kiri, Kanan) secara responsif melalui kendali jarak jauh berbasis web. Integrasi fitur multimedia juga berjalan dengan baik, di mana operator dapat memantau visual lingkungan secara real-time melalui camera stream dan mengirimkan output audio melalui koneksi Bluetooth tanpa mengganggu kinerja motor.
2. Keputusan desain untuk menggunakan tiga unit ESP32 yang terpisah (Root AP, Camera Unit, dan Main Controller) terbukti sangat efektif . Pembagian load ini berhasil mencegah terjadinya bottleneck pada prosesor. Unit kamera dapat fokus sepenuhnya pada streaming MJPEG, sementara unit kontroler fokus pada manajemen task motor dan audio secara real-time, sehingga stabilitas sistem tetap terjaga meskipun menangani beban data yang tinggi.
3. Penggunaan protokol MQTT dengan broker lokal (Mosquitto) terbukti menjadi solusi yang lebih andal dan stabil dibandingkan solusi berbasis *cloud* (Blynk) untuk proyek ini. Implementasi MQTT berhasil memberikan latensi pengiriman perintah yang sangat rendah (respons instan) dan, yang paling krusial, berhasil mengatasi kendala konflik hardware antara driver Wi-Fi dan ADC2 yang pada awalnya menyebabkan kegagalan sistem pada tahap awal pengembangan.
4. Dengan digunakannya ESP32 sebagai Access Point pusat, sistem mampu beroperasi secara mandiri tanpa ketergantungan pada koneksi internet eksternal. Hal ini menjadikan sistem sangat fleksibel untuk digunakan di lokasi yang tidak memiliki infrastruktur Wi-Fi yang memadai.

REFERENCES

[1] Electronics Bing "Surveillance Car using ESP32 Cam module | ESP32 Camera wi-fi car 🚓⚡⚡," YouTube, 2024. [Online]. Available: https://www.youtube.com/watch?v=5oxp1xsEk7M. [Accessed: Dec. 8, 2025].

[2] R. Santos and S. Santos "ESP32-CAM Remote Controlled Car Robot Web Server," Random Nerd Tutorials, n.d. [Online]. Available: https://randomnerdtutorials.com/esp32-cam-car-robot-web-server/. [Accessed: Dec. 8, 2025].

[3] Aslam Hossain "Surveillance Camera Car Using ESP32 CAM Module | ESP32 Surveillance Camera Car," YouTube, 2024. [Online]. Available: https://www.youtube.com/watch?v=IfB1w85Z2iE. [Accessed: Dec. 8, 2025].

[4] hash include electronics "Surveillance Car using ESP32 Cam module | ESP32 Camera wi-fi car 🚓," YouTube, 2021. [Online]. Available: https://www.youtube.com/watch?v=HfQ7lhhgDOk. [Accessed: Dec. 8, 2025].

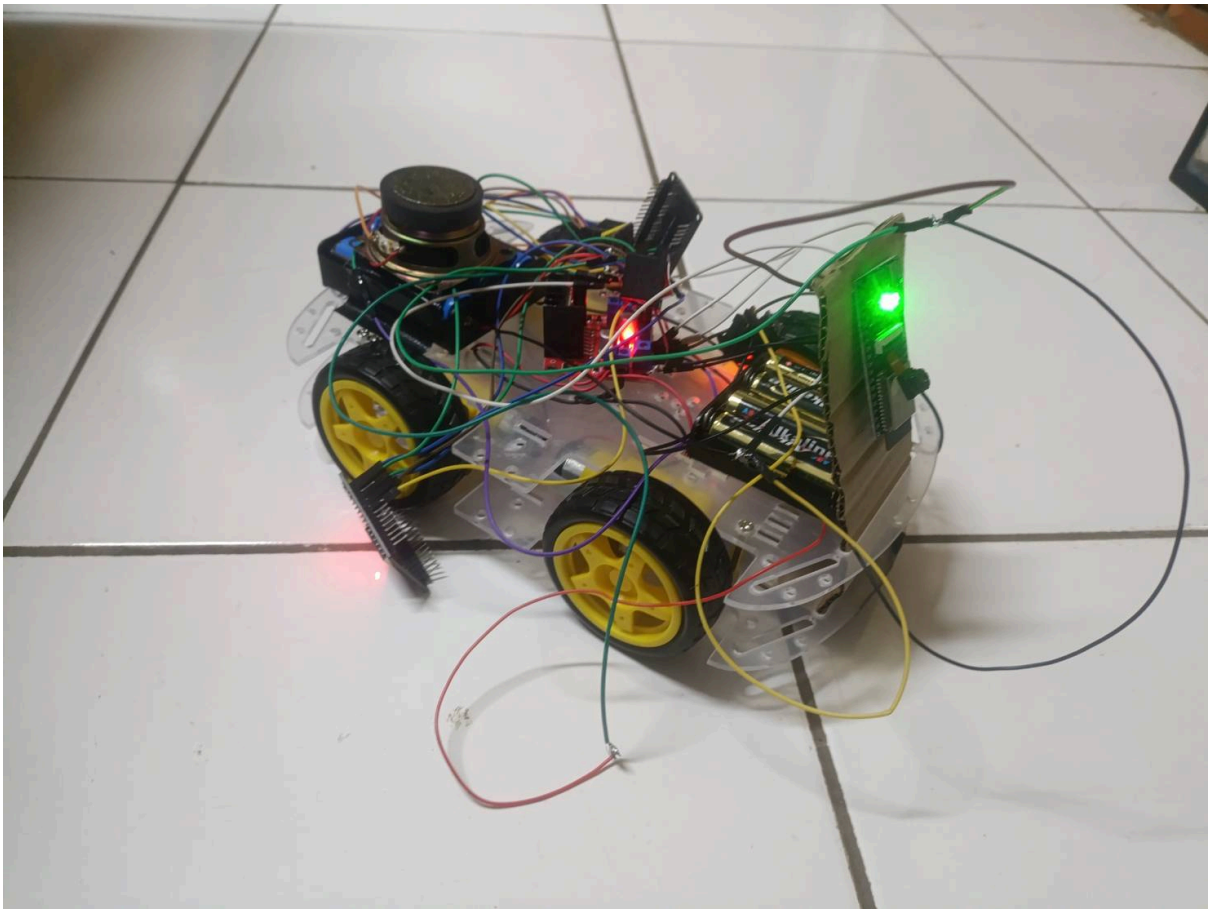
[5] Reference 5

[6] Reference 6

[7] And so on

APPENDICES

Appendix A: Project Schematic



Appendix B: Documentation