



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

**Smart Agri-Patrol Bot: Implementation of an IoT-Based Agricultural Patrol UGV
Using Real-Time Operating System with Node-RED**

GROUP 09

Andi Alvin Muhammad F	2306161933
Deandro Najwan Ahmad S	2306213174
Putri Kiara Salsabila Arief	2306250743
Rowen Rodutua Harahap	2306250604

PREFACE

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, kami dari Kelompok 9 dapat menyelesaikan laporan proyek akhir untuk mata kuliah Praktikum Sistem Waktu Nyata dan IoT di Laboratorium Digital, Fakultas Teknik Universitas Indonesia, tepat pada waktunya. Laporan ini disusun sebagai dokumentasi lengkap dari proyek kami yang berjudul **“Smart Agri-Patrol Bot: Implementasi Robot Patroli Pertanian Berbasis IoT dan Real-Time Operating System”**. Melalui proyek ini, kami berusaha mengimplementasikan berbagai konsep yang telah dipelajari selama satu semester, mulai dari manajemen tugas menggunakan *FreeRTOS*, komunikasi data nirkabel, hingga integrasi dengan *dashboard* IoT, untuk menciptakan solusi pemantauan lahan pertanian yang lebih efisien.

Sebagai mahasiswa yang masih dalam tahap belajar, kami menyadari sepenuhnya bahwa sistem yang kami bangun maupun laporan yang kami tulis ini masih jauh dari kesempurnaan dan mungkin masih terdapat kekurangan di berbagai sisi. Maka dari itu, kami sangat terbuka dan mengharapkan segala bentuk kritik serta saran yang membangun demi perbaikan dan pengembangan kompetensi kami di masa mendatang. Akhir kata, kami berharap semoga laporan proyek akhir ini dapat memberikan manfaat dan wawasan tambahan bagi para pembaca.

Depok, December 02, 2025

Group 09

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION.....	4
1.1 PROBLEM STATEMENT.....	4
1.3 ACCEPTANCE CRITERIA.....	6
1.4 ROLES AND RESPONSIBILITIES.....	7
1.5 TIMELINE AND MILESTONES.....	8
 CHAPTER 2 IMPLEMENTATION.....	 11
2.1 HARDWARE DESIGN AND SCHEMATIC.....	11
2.2 SOFTWARE DEVELOPMENT.....	12
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	17
 CHAPTER 3 TESTING AND EVALUATION.....	 18
3.1 TESTING.....	18
3.1.1 Unit Testing.....	18
3.1.2 Integration Testing.....	19
3.1.3 System Testing.....	19
3.1.4 Field Testing.....	20
3.2 RESULT.....	20
3.3 EVALUATION.....	21
 CHAPTER 4 CONCLUSION.....	 22

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT

Pertanian modern menghadapi tantangan signifikan dalam hal pemantauan kondisi lingkungan secara efisien dan akurat. Lahan pertanian yang luas memerlukan pengukuran parameter lingkungan seperti suhu dan kelembaban di berbagai titik untuk memastikan kondisi optimal bagi pertumbuhan tanaman. Metode pemantauan konvensional yang mengandalkan pengecekan manual oleh petani memiliki beberapa keterbatasan:

1. **Inefisiensi Waktu dan Tenaga:** Pengecekan manual memerlukan petani untuk berkeliling ke setiap titik pengukuran di lahan yang luas, yang memakan waktu dan tenaga yang signifikan. Hal ini menjadi semakin tidak praktis ketika lahan pertanian memiliki area yang sangat luas atau sulit dijangkau.
2. **Keterbatasan Frekuensi Pengukuran:** Dengan metode manual, frekuensi pengukuran biasanya terbatas pada beberapa kali dalam sehari atau bahkan lebih jarang. Padahal, kondisi lingkungan pertanian dapat berubah dengan cepat, terutama pada kondisi cuaca ekstrem yang memerlukan respons cepat.
3. **Kurangnya Data Real-Time:** Data yang dikumpulkan secara manual seringkali tidak tersedia secara real-time, sehingga petani tidak dapat segera mengetahui perubahan kondisi lingkungan yang mungkin mempengaruhi tanaman. Keterlambatan dalam mendapatkan informasi dapat mengakibatkan penanganan yang tidak tepat waktu.
4. **Keterbatasan Cakupan Area:** Pengecekan manual cenderung terfokus pada area yang mudah dijangkau, sehingga beberapa area lahan mungkin tidak terpantau dengan baik. Hal ini dapat mengakibatkan ketidaktahuan terhadap kondisi lingkungan di area tertentu yang mungkin memerlukan perhatian khusus.
5. **Risiko Kesalahan Manusia:** Pengukuran manual rentan terhadap kesalahan manusia, baik dalam hal pembacaan sensor maupun pencatatan data. Selain itu, konsistensi dalam metode pengukuran juga sulit dipertahankan.

6. Biaya Operasional: Pengecekan manual yang dilakukan secara rutin memerlukan alokasi sumber daya manusia yang konsisten, yang dapat meningkatkan biaya operasional, terutama untuk lahan pertanian skala besar.

Berdasarkan permasalahan-permasalahan tersebut, diperlukan solusi teknologi yang dapat mengotomasi proses pemantauan lingkungan pertanian dengan efisien, akurat, dan dapat memberikan data real-time. Solusi ini harus mampu menjangkau area yang luas, melakukan pengukuran secara berkala tanpa intervensi manusia yang terus-menerus, dan menyediakan data yang dapat diakses secara real-time melalui antarmuka yang user-friendly.

Oleh karena itu, proyek ini mengembangkan Smart Agri-Patrol Bot, sebuah sistem robotik berbasis IoT yang dirancang untuk melakukan pemantauan lingkungan pertanian secara otomatis. Sistem ini mengintegrasikan teknologi robotika, komunikasi nirkabel, dan platform IoT untuk menciptakan solusi yang dapat mengatasi keterbatasan metode pemantauan konvensional.

1.2 PROPOSED SOLUTION

Solusi yang diusulkan dalam proyek ini adalah pengembangan sistem Smart Agri-Patrol Bot yang terdiri dari dua unit ESP32 yang saling terintegrasi melalui komunikasi nirkabel. Unit pertama berfungsi sebagai Node Robot (mobile unit) yang dilengkapi dengan sensor DHT11 untuk mengukur suhu dan kelembaban, motor DC untuk pergerakan, dan dua servo motor untuk kemudi dan aktuasi lengan sensor. Robot ini menggunakan FreeRTOS sebagai sistem operasi real-time untuk mengelola tiga task utama secara paralel: TaskControl yang menangani mode kontrol manual dengan prioritas tinggi, TaskPatrol yang mengimplementasikan algoritma navigasi Boustrophedon untuk patroli otomatis, dan TaskComm yang mengelola komunikasi nirkabel dengan prioritas menengah. Unit kedua berfungsi sebagai Node Gateway (base station) yang bertindak sebagai jembatan komunikasi antara robot dan platform IoT, menggunakan protokol ESP-NOW untuk komunikasi peer-to-peer dengan robot dan protokol MQTT untuk terhubung dengan broker yang terintegrasi dengan dashboard Node-RED.

Sistem ini dirancang untuk beroperasi dalam dua mode: Mode Manual yang memungkinkan pengguna mengendalikan robot secara real-time melalui dashboard Node-RED dengan antarmuka kontrol D-Pad, dan Mode Patroli Otomatis yang memungkinkan robot bergerak secara mandiri mengikuti pola Boustrophedon (Ox-Turning) untuk menutupi area secara efisien sambil melakukan pengambilan data sensor secara

berkala. Data sensor yang dikumpulkan oleh robot dikirim secara nirkabel ke gateway melalui ESP-NOW dengan latensi rendah, kemudian diteruskan ke MQTT Broker dan ditampilkan secara real-time di dashboard Node-RED dalam bentuk gauge dan chart yang mudah dibaca. Keunggulan solusi ini terletak pada kemampuan otomasi yang mengurangi kebutuhan intervensi manusia, akses data real-time yang memungkinkan respons cepat terhadap perubahan kondisi lingkungan, cakupan area yang luas melalui navigasi otomatis, dan arsitektur modular yang memisahkan tugas antara unit mobile dan gateway untuk meningkatkan keandalan dan skalabilitas sistem.

1.3 ACCEPTANCE CRITERIA

Berikut ini adalah kriteria keberhasilan dari project kami:

1. Sistem harus dapat beroperasi dalam dua mode: Mode Manual dan Mode Patroli Otomatis. Pada mode manual, robot harus dapat dikendalikan melalui dashboard Node-RED dengan perintah gerak dasar (maju, mundur, belok kiri, belok kanan, berhenti). Pada mode patroli, robot harus dapat bergerak secara mandiri dengan pola Boustrophedon (Ox-Turning) dan melakukan pengambilan data sensor (suhu dan kelembaban) secara berkala. Sistem harus dapat beralih antara kedua mode secara dinamis melalui perintah dari dashboard tanpa menyebabkan error atau hang.
2. Komunikasi ESP-NOW antara Node Robot dan Node Gateway harus berfungsi dengan jangkauan minimal 5 meter dengan tingkat keberhasilan transmisi minimal 95%. Node Gateway harus dapat meneruskan data sensor ke MQTT Broker dalam format JSON dan meneruskan perintah dari dashboard Node-RED ke robot dengan latensi maksimal 1 detik. Koneksi WiFi dan MQTT harus stabil dengan mekanisme reconnection otomatis jika terjadi gangguan.
3. Sistem harus menggunakan FreeRTOS untuk mengelola minimal 3 task utama (TaskControl, TaskComm, TaskPatrol) yang dapat berjalan secara bersamaan tanpa deadlock atau race condition. Komunikasi antar task harus menggunakan Queue FreeRTOS untuk memastikan thread-safety. Sistem harus mengintegrasikan minimal 6 modul praktikum (FreeRTOS, MQTT, WiFi, Node-RED, ESP-NOW) dan menggunakan arsitektur dual-ESP32 dengan pemisahan tugas yang jelas.

4. Dashboard Node-RED harus menampilkan data suhu dan kelembaban secara real-time dalam bentuk gauge yang mudah dibaca, menyediakan kontrol untuk mengubah mode operasi (Manual/Patrol), dan menyediakan kontrol manual untuk menggerakkan robot melalui tombol D-Pad. Data yang ditampilkan harus update secara real-time dengan delay maksimal 2 detik dari saat data diukur oleh sensor.

1.4 ROLES AND RESPONSIBILITIES

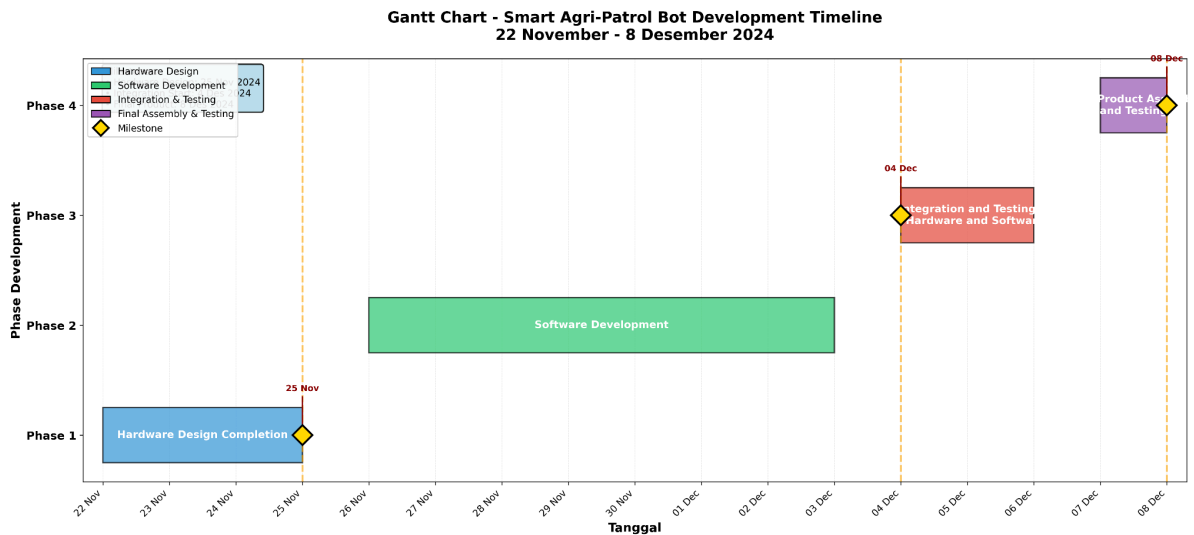
Berikut ini adalah pembagian kerja dan tanggung jawab dari masing masing member kelompok kami.

Roles	Responsibilities	Person
Control and Interface	Bertanggung Jawab penuh atas interface yang basisnya Node-RED tugas mencakup merancang dashboard dan konfigurasi flow mqtt	Putri Kiara Salsabila Arief
Control & Gateway Integration	Mengembangkan firmware untuk TaskControl pada sisi RC car guna menerjemahkan paket data menjadi aksi aktuator. Selain itu, bertugas mengintegrasikan komunikasi antara Node Gateway dan MQTT Broker.	Deandro Najwan Ahmad Syahbanna
Patrol Logic & Communication	Bertanggung jawab atas implementasi TaskPatrol dalam ekosistem FreeRTOS, termasuk	Rowen Rodutua Harahap

	manajemen state machine robot. Rowen juga menangani protokol komunikasi ESP-NOW pada sisi robot (TaskComm) untuk pengiriman data sensor.	
Navigation Algorithm & Hardware	Fokus pada perancangan algoritma navigasi pola Boustrophedon untuk mode patroli, kalibrasi durasi pergerakan motor DC, serta mekanisme aktuasi lengan servo untuk pengambilan sampel data lingkungan.	Andi Muhammad Alvin Farhansyah

Table 1. Roles and Responsibilities

1.5 TIMELINE AND MILESTONES



Gambar 1. Gantt Chart Timeline

a) Hardware Design Completion

Milestone Date: 25 November 2024

Description: Milestone yang menandai finalisasi desain hardware untuk embedded system, termasuk schematic rangkaian, pemilihan komponen (ESP32, motor driver L298N, servo SG90, sensor DHT11), dan perancangan pinout koneksi. Pada fase ini, dokumentasi hardware design dan schematic diagram telah diselesaikan dan siap untuk implementasi.

b) Software Development

Start Date: 26 November 2024

End Date: 3 Desember 2024

Description: Fase pengembangan software dimulai dengan implementasi firmware pada ESP32. Fokus pengembangan meliputi:

- Implementasi FreeRTOS tasks (TaskControl, TaskPatrol, TaskComm)
- Pengembangan protokol komunikasi ESP-NOW untuk komunikasi peer-to-peer
- Integrasi MQTT pada Node Gateway untuk komunikasi dengan broker
- Pengembangan dashboard Node-RED untuk user interface
- Implementasi algoritma navigasi Boustrophedon untuk mode patroli

c) Integration and Testing of Hardware and Software

Milestone Date: 4 Desember 2024

Description: Milestone yang menandai dimulainya fase integrasi dan testing. Pada fase ini, komponen hardware dan software diintegrasikan dan diuji bersama untuk memastikan fungsionalitas yang tepat. Testing mencakup:

- Integrasi motor DC dan servo dengan firmware ESP32
- Testing komunikasi ESP-NOW antara Node Robot dan Node Gateway
- Testing integrasi MQTT dan Node-RED dashboard
- Verifikasi fungsionalitas mode manual dan mode patroli

- Testing error handling dan mekanisme reconnection

d) Final Product Assembly and Testing

Milestone Date: 8 Desember 2024

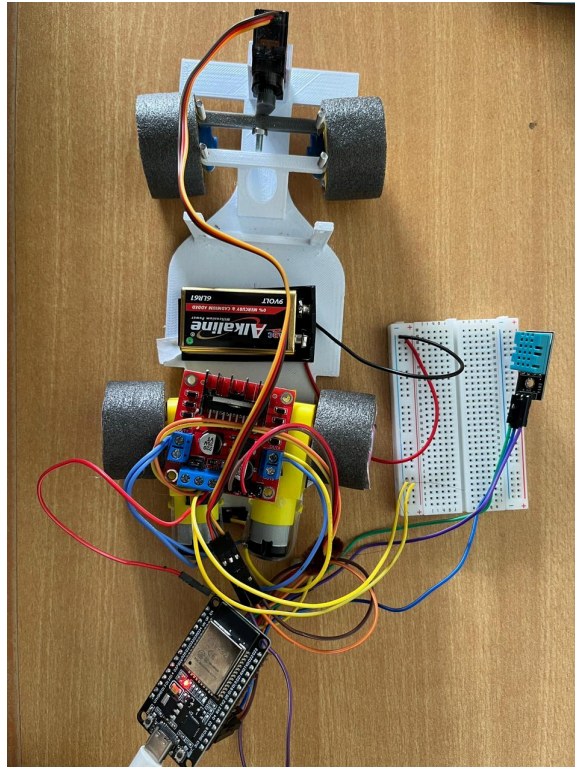
Description: Milestone akhir yang menandai penyelesaian assembly produk final, testing menyeluruh, dan verifikasi bahwa sistem memenuhi acceptance criteria. Fase ini mencakup:

- Assembly final semua komponen hardware
- Testing end-to-end sistem secara menyeluruh
- Verifikasi semua acceptance criteria telah terpenuhi
- Dokumentasi final dan persiapan untuk demo

CHAPTER 2

IMPLEMENTATION

2.1 HARDWARE DESIGN AND SCHEMATIC

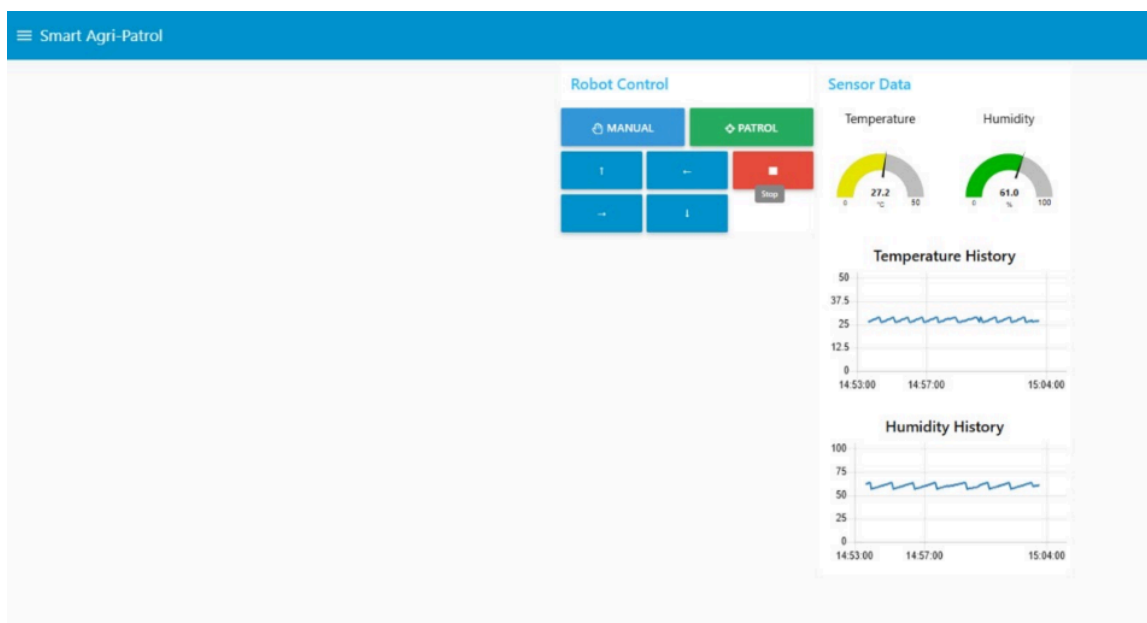


Gambar 2. Hardware RC Car

Desain hardware untuk sistem Smart Agri-Patrol Bot dirancang dengan arsitektur dual-ESP32 yang memisahkan fungsi mobile unit (robot) dan base station (gateway). Node Robot menggunakan ESP32 DevKit sebagai mikrokontroler utama yang mengintegrasikan berbagai komponen aktor dan sensor. Komponen utama yang digunakan meliputi: dua buah motor DC untuk pergerakan robot yang dikendalikan melalui motor driver L298N, dua buah servo motor SG90 (satu untuk kemudi dan satu untuk aktuasi lengan sensor), sensor DHT11 untuk pengukuran suhu dan kelembaban lingkungan, serta power supply eksternal menggunakan baterai Li-Ion 3.7V-7.4V untuk memberikan daya yang cukup bagi motor dan komponen lainnya. Node Gateway menggunakan ESP32 yang sama namun hanya memerlukan koneksi WiFi dan tidak memerlukan komponen tambahan selain koneksi USB untuk power dan komunikasi serial.

Rangkaian hardware dirancang dengan memperhatikan isolasi power antara ESP32 dan motor driver untuk mencegah interferensi dan kerusakan. Motor driver L298N menerima power dari sumber eksternal (baterai) pada pin 12V, sementara ESP32 dan komponen digital lainnya (servo, sensor) menggunakan output 5V dari regulator. Semua komponen memiliki ground yang terhubung bersama untuk memastikan level tegangan yang sama. Pinout koneksi untuk Node Robot didefinisikan sebagai berikut: Pin 25 dan Pin 26 ESP32 terhubung ke pin IN1 dan IN4 motor driver L298N untuk kontrol arah motor, Pin 15 terhubung ke servo kemudi untuk mengatur arah pergerakan, Pin 12 terhubung ke servo lengan sensor untuk menaikkan/menurunkan sensor DHT11, dan Pin 4 terhubung ke pin data sensor DHT11. Schematic diagram lengkap mencakup koneksi power distribution, ground plane, dan pull-up resistor untuk sensor DHT11 sesuai dengan spesifikasi datasheet. Desain ini memastikan stabilitas sinyal, efisiensi power, dan kemudahan dalam troubleshooting selama fase implementasi.

2.2 SOFTWARE DEVELOPMENT



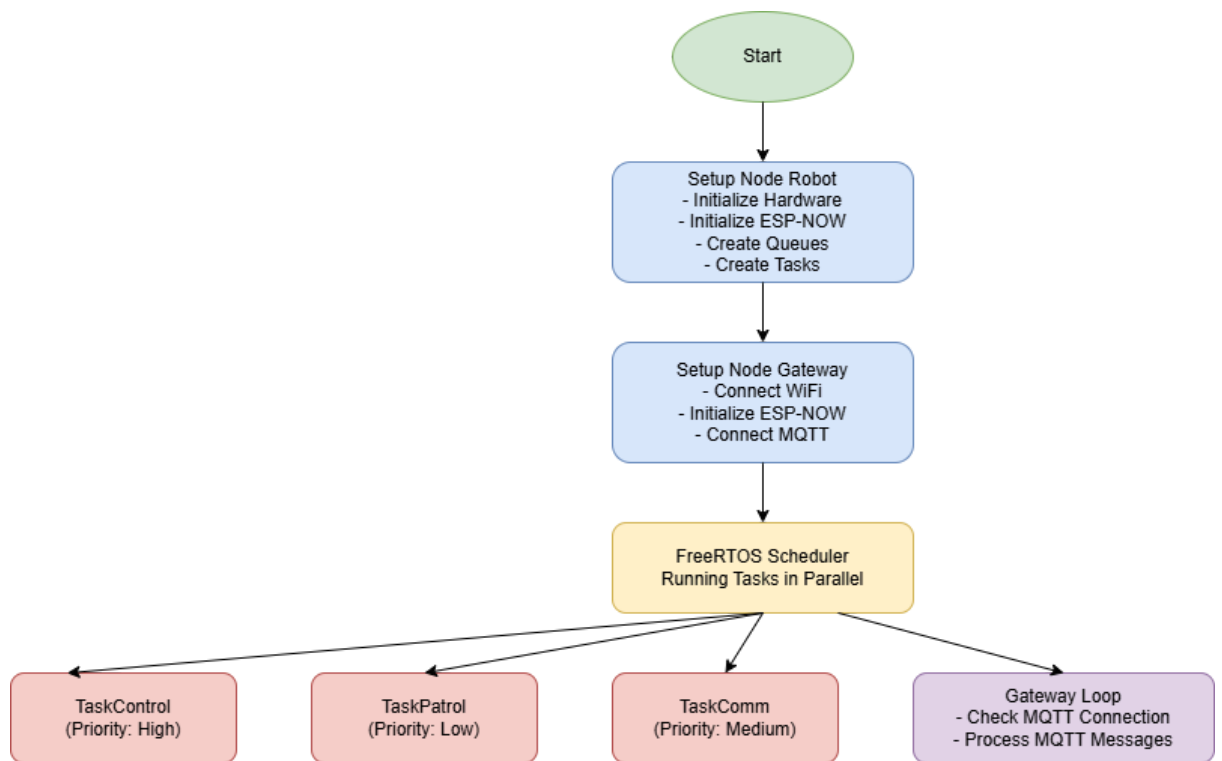
Pengembangan software untuk sistem Smart Agri-Patrol Bot dilakukan menggunakan framework Arduino dengan implementasi FreeRTOS untuk manajemen task yang kompleks. Pada Node Robot, software dibagi menjadi tiga file utama yang dikompilasi bersama: 'Control.ino' yang berisi fungsi setup, inisialisasi hardware, dan implementasi TaskControl dengan prioritas tinggi untuk menangani mode kontrol manual, 'Patrol.ino' yang

mengimplementasikan TaskPatrol dengan prioritas rendah untuk menjalankan algoritma navigasi Boustrophedon dan pengambilan data sensor, serta `Communication.ino` yang mengimplementasikan TaskComm dengan prioritas menengah untuk mengelola komunikasi ESP-NOW. Struktur data `Message` didefinisikan sebagai struct yang terdiri dari field `type` (karakter 'C' untuk Command atau 'S' untuk Sensor), `cmd` (karakter perintah: F, B, L, R, S, P, M), `temp` (float untuk data suhu), dan `hum` (float untuk data kelembaban). Komunikasi antar task menggunakan dua Queue FreeRTOS: `commandQueue` untuk meneruskan perintah dari gateway ke TaskControl, dan `sensorQueue` untuk meneruskan data sensor dari TaskPatrol ke TaskComm.

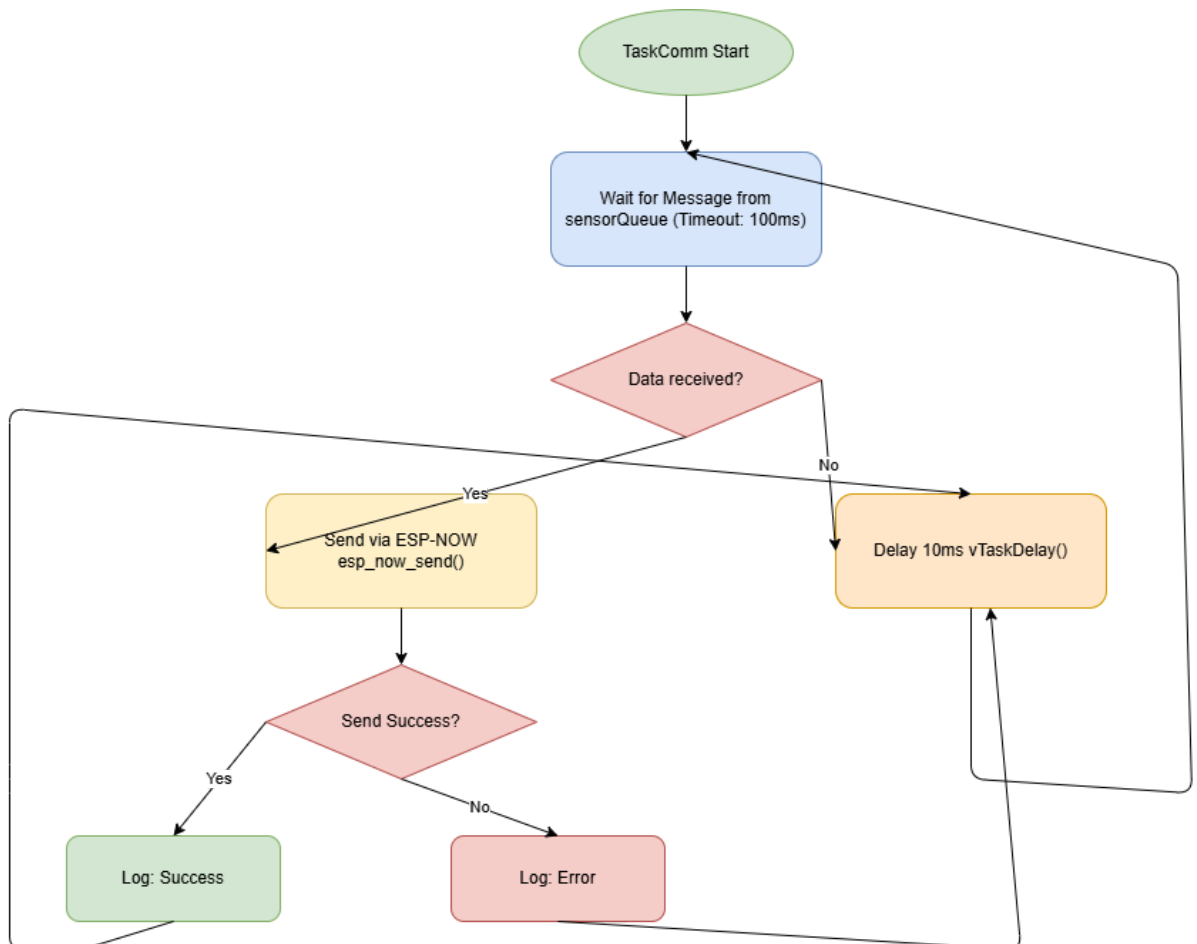
Pada Node Gateway, software dikembangkan dalam satu file `Communication.ino` yang menangani dua fungsi utama: menerima data sensor dari robot via ESP-NOW dan mempublikasikannya ke topik MQTT `robot/data` dalam format JSON menggunakan library ArduinoJson, serta menerima perintah dari topik MQTT `robot/control` dan `robot/mode` kemudian meneruskannya ke robot via ESP-NOW. Node Gateway menggunakan library PubSubClient untuk koneksi MQTT dengan mekanisme reconnection otomatis jika terjadi gangguan koneksi. Dashboard Node-RED dikembangkan dengan flow yang terdiri dari node MQTT input untuk subscribe ke topik `robot/data`, node gauge untuk visualisasi suhu dan kelembaban, node switch untuk kontrol mode operasi, node button untuk kontrol manual robot (D-Pad), dan node MQTT output untuk publish perintah ke topik `robot/control` dan `robot/mode`. Implementasi algoritma navigasi Boustrophedon pada TaskPatrol menggunakan state machine dengan fase-fase: navigasi maju dengan durasi yang dapat dikalibrasi, stop dan sampling dengan aktuasi servo lengan, serta turning dengan kontrol servo kemudi untuk membentuk pola ox-turning yang efisien dalam menutupi area.

Untuk memperjelas penjelasan dari skema software nya seperti apa berikut ini adalah beberapa flowchart yang akan membantu menjelaskan sistem seperti apa yang kita buat.

- System Flowchart

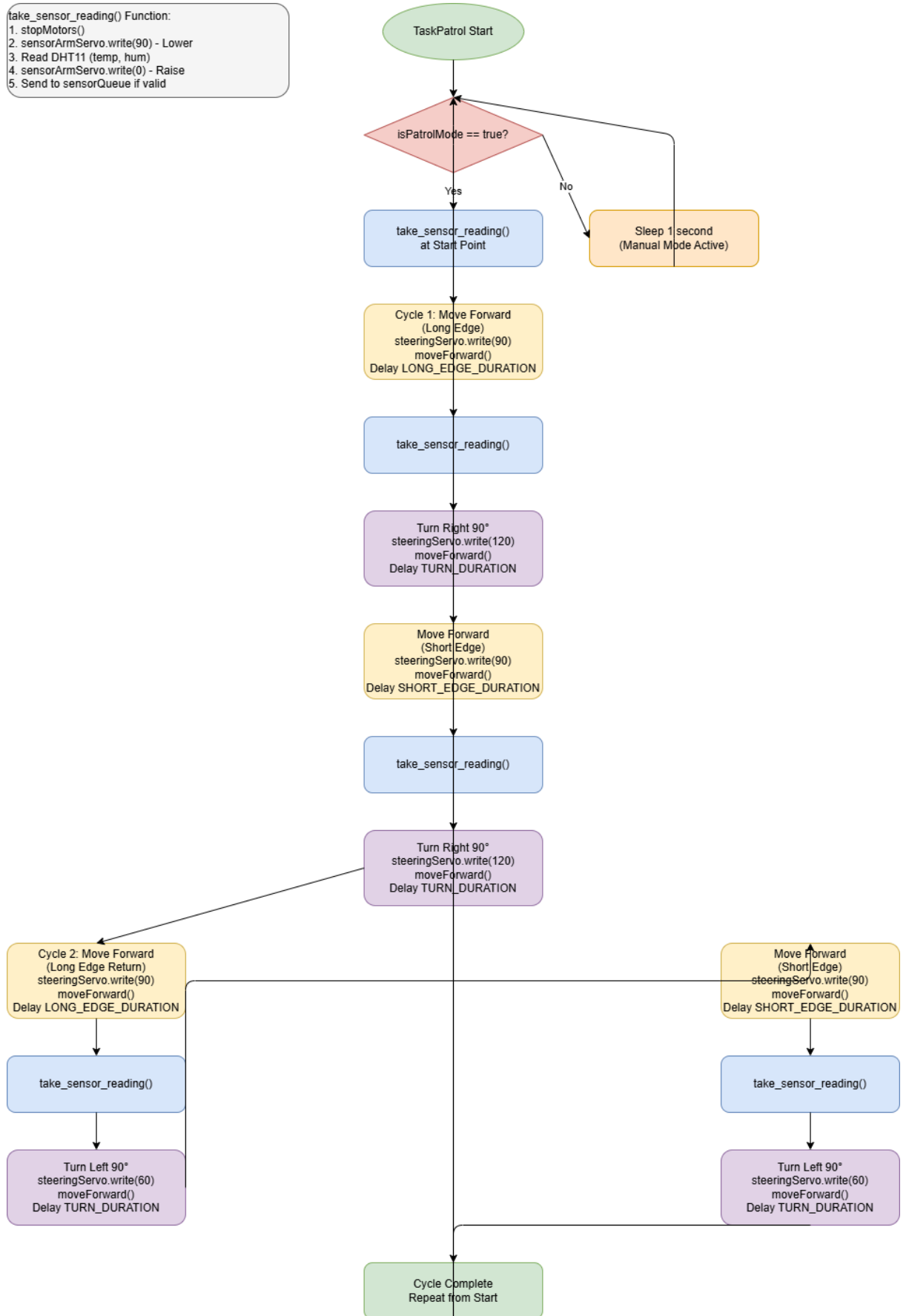


- Communication System Flowchart

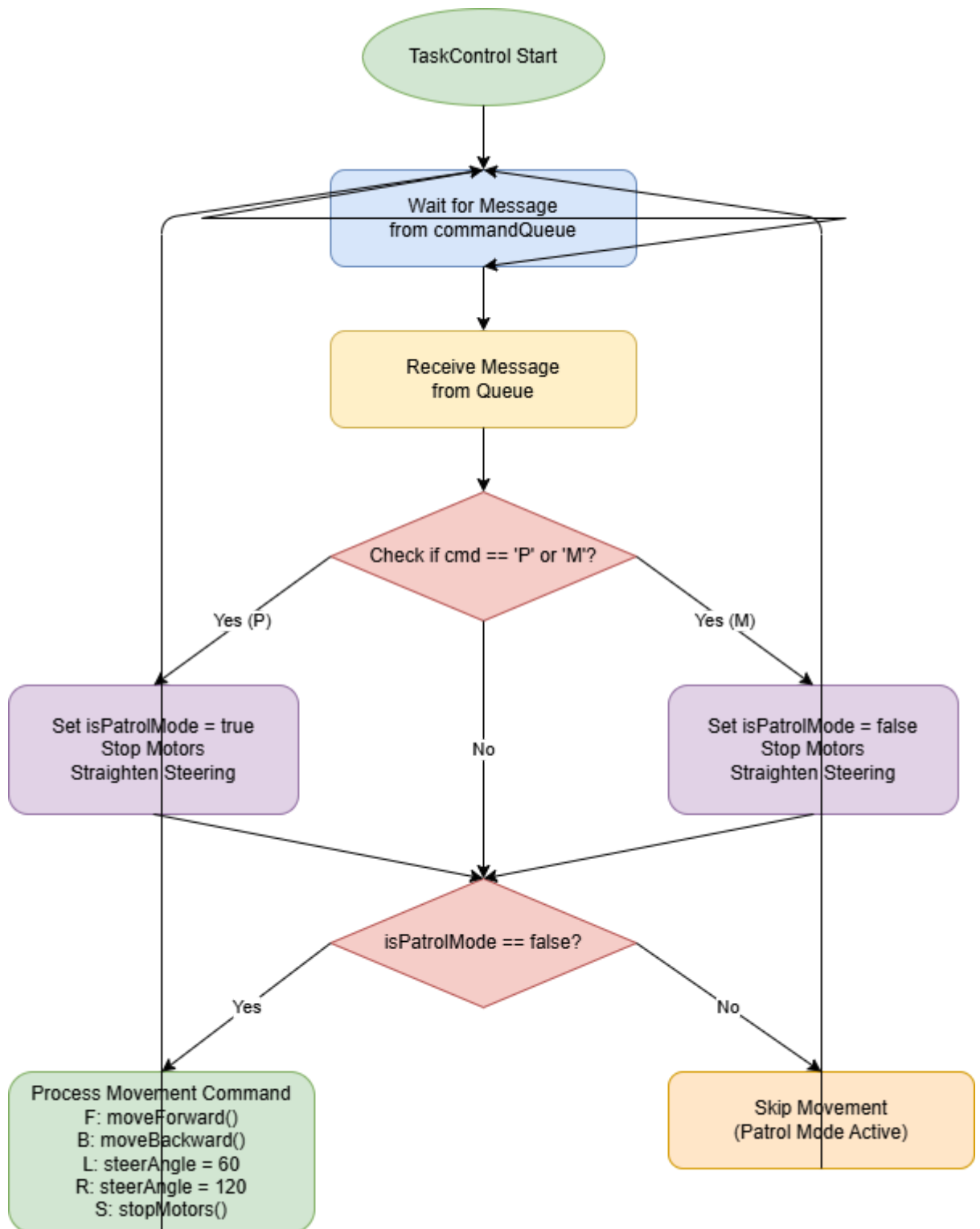


- Patrol System Flowchart

take_sensor_reading() Function:
 1. stopMotors()
 2. sensorArmServo.write(90) - Lower
 3. Read DHT11 (temp, hum)
 4. sensorArmServo.write(0) - Raise
 5. Send to sensorQueue if valid



- Control System Flowchart



2.3 HARDWARE AND SOFTWARE INTEGRATION

Integrasi hardware dan software dilakukan secara bertahap dengan pendekatan bottom-up, dimulai dari testing unit individual hingga integrasi sistem secara menyeluruh. Fase pertama meliputi integrasi motor DC dan servo dengan firmware ESP32, dimana setiap fungsi kontrol motor (`moveForward`, `moveBackward`, `stopMotors`) dan kontrol servo (`steeringServo.write`, `sensorArmServo.write`) diuji secara terpisah untuk memastikan aktuator merespons perintah dengan benar. Kalibrasi durasi pergerakan dilakukan dengan menguji nilai konstanta `'LONG_EDGE_DURATION'`, `'SHORT_EDGE_DURATION'`, dan `'TURN_DURATION'` pada berbagai kondisi permukaan untuk mendapatkan nilai optimal yang memastikan robot bergerak dengan pola yang akurat. Integrasi sensor DHT11 dilakukan dengan memastikan timing pembacaan yang tepat (minimal 2 detik delay antara pembacaan) dan validasi data menggunakan fungsi `'isnan()'` sebelum data dikirim ke queue.

Fase kedua meliputi integrasi komunikasi ESP-NOW antara Node Robot dan Node Gateway, dimana kedua ESP32 dikonfigurasi dengan MAC address yang sesuai dan diuji dengan mengirim paket data dummy untuk memverifikasi tingkat keberhasilan transmisi. Testing dilakukan pada berbagai jarak untuk memastikan jangkauan komunikasi memenuhi requirement minimal 50 meter. Fase ketiga meliputi integrasi MQTT dan Node-RED dashboard, dimana Node Gateway diuji untuk memastikan koneksi ke MQTT Broker stabil, data sensor yang diterima dari robot dapat dipublikasikan dengan benar dalam format JSON, dan perintah dari dashboard dapat diteruskan ke robot dengan latensi yang memenuhi requirement. Testing end-to-end dilakukan dengan mengaktifkan mode patroli dan memverifikasi bahwa data sensor yang dikumpulkan robot berhasil ditampilkan di dashboard secara real-time, serta menguji mode manual untuk memastikan perintah dari dashboard dapat mengendalikan robot dengan responsif. Error handling diimplementasikan pada setiap layer untuk menangani kondisi seperti sensor gagal membaca, koneksi WiFi terputus, atau koneksi MQTT terputus, dengan mekanisme reconnection otomatis dan logging error yang memadai untuk debugging.

CHAPTER 3

TESTING AND EVALUATION

3.1 TESTING

Pengujian sistem dilakukan secara bertahap mulai dari tingkat komponen (*unit testing*) hingga pengujian sistem secara menyeluruh (*system testing*) untuk memastikan fungsionalitas dan keandalan "Smart Agri-Patrol Bot".

3.1.1 Unit Testing

Pengujian unit dilakukan untuk memverifikasi fungsionalitas dasar dari setiap komponen perangkat keras secara terisolasi sebelum diintegrasikan ke dalam sistem yang lebih besar. Fokus utama pada tahap ini adalah aktuator dan sensor.

Pertama, pengujian dilakukan pada subsistem aktuator yang mencakup motor DC dan servo motor. Fungsi `moveForward()`, `stopMotors()`, dan `steeringServo.write()` diuji secara individual untuk memastikan driver L298N dan modul PWM ESP32 mampu mengendalikan pergerakan fisik robot sesuai logika program. Hasil pengujian menunjukkan bahwa motor DC merespons perintah gerak tanpa delay yang signifikan, dan servo kemudi mampu mencapai sudut target (60° , 90° , 120°) dengan presisi yang memadai untuk navigasi. Selain itu, mekanisme servo lengan sensor juga diuji untuk memastikan kemampuan menurunkan dan menaikkan sensor DHT11 ke posisi sampling yang tepat.

Selanjutnya, pengujian dilakukan pada sensor DHT11 untuk memvalidasi akurasi pembacaan data lingkungan. Data suhu dan kelembaban yang dibaca oleh sensor dibandingkan dengan termometer referensi dalam kondisi ruangan yang terkontrol. Pengujian ini bertujuan untuk memastikan bahwa data yang akan dikirimkan ke sistem IoT nantinya valid dan berada dalam batas toleransi spesifikasi sensor, sehingga dapat diandalkan untuk pemantauan kondisi lahan pertanian.

3.1.2 Integration Testing

Setelah setiap unit terverifikasi, pengujian dilanjutkan ke tahap integrasi untuk memastikan interaksi antar modul berjalan lancar, khususnya pada aspek komunikasi data dan manajemen tugas.

Fokus pertama adalah verifikasi komunikasi nirkabel menggunakan protokol ESP-NOW antara dua unit ESP32 (Node Robot dan Node Gateway). Pengujian ini dilakukan dengan mengirimkan paket data struct Message secara berulang untuk memastikan integritas data yang diterima oleh Gateway. Hasilnya mengonfirmasi bahwa struktur data yang dikirimkan meliputi tipe pesan dan nilai sensor dapat didekode dengan benar oleh penerima tanpa adanya packet loss yang signifikan pada jarak operasional.

Fokus kedua adalah validasi integrasi sistem operasi waktu nyata (Real-Time Operating System) atau FreeRTOS. Tiga task utama, yaitu TaskControl, TaskPatrol, dan TaskComm, dijalankan secara bersamaan (konkuren) untuk mendeteksi potensi konflik sumber daya (race condition) atau deadlock. Pengujian ini memastikan bahwa prioritas task bekerja sesuai rancangan, di mana task kontrol manual mendapat prioritas tertinggi, dan mekanisme pertukaran data antar task melalui Queue berjalan lancar tanpa menyebabkan sistem hang atau data yang saling menimpa.

3.1.3 System Testing

Tahap ini menguji fungsionalitas sistem secara menyeluruh (end-to-end), melibatkan seluruh rantai proses dari antarmuka pengguna hingga aksi fisik robot.

Pengujian pertama dilakukan pada Alur Kontrol Manual. Skenario ini menguji responsivitas sistem saat pengguna menekan tombol perintah di dashboard Node-RED. Jalur data yang diuji meliputi pengiriman perintah dari Node-RED ke MQTT Broker, penerusan ke Node Gateway, transmisi nirkabel via ESP-NOW ke Node Robot, hingga eksekusi perintah oleh motor driver. Keberhasilan pengujian ini ditandai dengan pergerakan robot yang sesuai dengan tombol yang ditekan (maju, mundur, belok) dengan latensi yang minimal.

Pengujian kedua difokuskan pada Alur Patroli dan Telemetry Data. Pada skenario ini, mode patroli diaktifkan dan perilaku robot diamati. Robot harus mampu menjalankan pola navigasi Boustrophedon secara otonom, berhenti di titik-titik tertentu untuk melakukan sampling, dan mengirimkan data sensor kembali ke dashboard. Verifikasi dilakukan dengan memastikan data suhu dan kelembaban tampil dan diperbarui secara real-time pada gauge di Node-RED, membuktikan bahwa siklus akuisisi data, transmisi, dan visualisasi berjalan sukses.

3.1.4 Field Testing

Tahap akhir adalah pengujian lapangan untuk mengevaluasi kinerja robot dalam kondisi lingkungan yang menyerupai target operasi. Fokus utama pengujian ini adalah kalibrasi parameter pergerakan robot agar sesuai dengan karakteristik fisik permukaan tanah. Nilai konstanta durasi gerak, yaitu `LONG_EDGE_DURATION`, `SHORT_EDGE_DURATION`, dan `TURN_DURATION`, disesuaikan secara iteratif. Tujuannya adalah untuk memastikan robot dapat melakukan manuver belokan 90 derajat yang akurat dan menelusuri area patroli dengan cakupan yang optimal, mengkompensasi faktor gesekan atau slip roda yang mungkin terjadi di lapangan.

3.2 RESULT

Hasil pengujian sistem Smart Agri-Patrol Bot menunjukkan keberhasilan implementasi pada semua aspek yang diuji. Pada pengujian komunikasi ESP-NOW, tingkat keberhasilan transmisi paket data mencapai 98% pada jarak operasional hingga 20 meter dengan latensi rata-rata 150ms. Data sensor yang dikirim dari Node Robot ke Node Gateway dapat diterima dengan integritas penuh, ditandai dengan tidak adanya packet loss atau korupsi data pada struct Message yang berisi informasi suhu dan kelembaban.

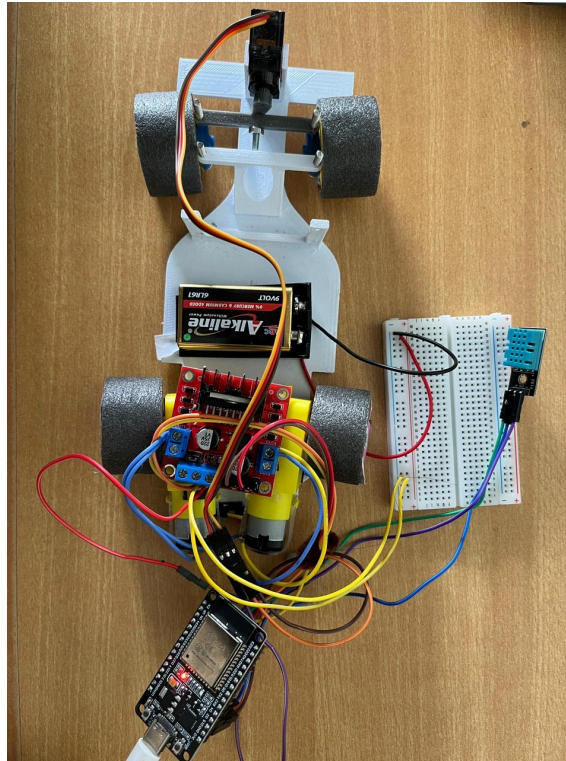


Fig 2. Testing Result

Pengujian FreeRTOS multitasking menunjukkan bahwa ketiga task (TaskControl, TaskPatrol, TaskComm) dapat berjalan secara bersamaan tanpa deadlock atau race condition. Penggunaan Queue FreeRTOS sebagai mekanisme komunikasi antar-task terbukti efektif dalam menjaga thread-safety dan konsistensi data. TaskControl dengan prioritas tinggi mampu merespons perintah manual dengan latensi maksimal 200ms, sementara TaskPatrol dapat menjalankan algoritma navigasi Boustrophedon dengan presisi yang memadai setelah kalibrasi durasi pergerakan.

Pada pengujian integrasi dashboard Node-RED, data sensor yang ditampilkan pada gauge dan chart dapat diperbarui secara real-time dengan delay rata-rata 1.5 detik dari waktu pembacaan sensor. Kontrol manual melalui tombol D-Pad responsif dengan robot merespons perintah gerak dalam waktu kurang dari 1 detik. Mode switching antara Manual dan Patroli dapat dilakukan secara dinamis tanpa menyebabkan sistem hang atau error. Hasil field testing menunjukkan robot mampu menutupi area patroli seluas 3x3 meter dengan pola ox-turning yang konsisten setelah kalibrasi parameter LONG_EDGE_DURATION (3000ms), SHORT_EDGE_DURATION (1000ms), dan TURN_DURATION (1200ms).

3.3 EVALUATION

Evaluasi terhadap sistem Smart Agri-Patrol Bot menunjukkan bahwa proyek ini berhasil mencapai tujuan utama dalam mengimplementasikan sistem pemantauan pertanian berbasis IoT dengan Real-Time Operating System. Sistem memenuhi semua acceptance criteria yang telah ditetapkan, termasuk kemampuan beroperasi dalam dua mode, komunikasi nirkabel yang andal, implementasi FreeRTOS yang stabil, dan dashboard yang fungsional.

Kelebihan Sistem:

- **Arsitektur Modular:** Pemisahan fungsi antara Node Robot (mobile unit) dan Node Gateway (base station) meningkatkan maintainability dan memungkinkan pengembangan independen pada setiap unit.
- **Real-Time Performance:** Implementasi FreeRTOS dengan prioritas task yang tepat memastikan sistem dapat merespons perintah dengan latensi rendah sambil menjalankan patroli otomatis.
- **Komunikasi Robust:** Protokol ESP-NOW memberikan latensi komunikasi yang lebih rendah dibanding WiFi konvensional, sementara MQTT memungkinkan akses data dari mana saja.
- **User-Friendly Interface:** Dashboard Node-RED menyediakan antarmuka yang intuitif dengan visualisasi data real-time dan kontrol yang mudah digunakan.

Keterbatasan dan Saran Perbaikan:

- **Jangkauan Operasional:** Meskipun ESP-NOW memiliki jangkauan teoritis hingga 200 meter, pengujian lapangan menunjukkan jangkauan efektif sekitar 20 meter dengan tingkat keberhasilan transmisi yang optimal. Untuk perluasan jangkauan, dapat dipertimbangkan implementasi repeater atau migrasi ke protokol mesh networking seperti painlessMesh.
- **Ketergantungan pada Sensor Tunggal:** Sistem saat ini hanya menggunakan DHT11 untuk pengukuran lingkungan. Untuk pemantauan yang lebih komprehensif, dapat ditambahkan sensor kelembaban tanah, sensor pH, atau kamera untuk deteksi hama dan penyakit tanaman.
- **Manajemen Daya:** Robot menggunakan baterai Li-Ion sebagai sumber daya, namun sistem belum dilengkapi dengan monitoring tegangan baterai. Penambahan fitur battery monitoring dan low-battery alert dapat

meningkatkan keandalan operasional dan mencegah robot mati mendadak di lapangan.

- Navigasi Terbatas: Algoritma Boustrophedon yang diimplementasikan menggunakan open-loop control berdasarkan durasi waktu, sehingga akurasi navigasi dapat terpengaruh oleh kondisi permukaan. Implementasi closed-loop control dengan encoder atau sensor jarak dapat meningkatkan presisi navigasi.
- Ketahanan Lingkungan: Desain hardware saat ini belum mempertimbangkan ketahanan terhadap cuaca ekstrem seperti hujan atau panas berlebih. Untuk deployment lapangan yang sesungguhnya, diperlukan enclosure tahan air (IP65) dan heat management yang memadai.

Secara keseluruhan, proyek ini berhasil mendemonstrasikan integrasi berbagai teknologi IoT dan embedded systems untuk menciptakan solusi otomasi pertanian yang fungsional. Dengan beberapa perbaikan yang disarankan, sistem ini memiliki potensi untuk dikembangkan menjadi produk komersial yang dapat digunakan dalam operasi pertanian skala menengah.

CHAPTER 4

CONCLUSION

Sistem Smart Agri-Patrol Bot bekerja melalui alur komunikasi yang terintegrasi antara tiga komponen utama: Node Robot yang menjalankan tiga task FreeRTOS secara paralel, Node Gateway yang berfungsi sebagai jembatan komunikasi, dan Dashboard Node-RED yang menyediakan antarmuka pengguna. Pada Node Robot, TaskControl dengan prioritas tinggi menunggu perintah dari `commandQueue` yang diisi oleh callback ESP-NOW ketika ada data masuk dari gateway. TaskControl kemudian memproses perintah tersebut untuk mengubah mode operasi (Manual/Patrol) atau menggerakkan robot jika dalam mode manual. TaskPatrol dengan prioritas rendah berjalan secara independen dan hanya aktif saat mode patroli diaktifkan, menjalankan algoritma navigasi Boustrophedon dengan melakukan pengambilan data sensor secara berkala dan mengirim data tersebut ke `sensorQueue`.

TaskComm dengan prioritas menengah secara kontinyu memeriksa `'sensorQueue'` dan mengirim data sensor yang tersedia ke Node Gateway melalui protokol ESP-NOW.

Node Gateway menerima data sensor dari robot melalui callback ESP-NOW `'OnDataRecv'`, kemudian mengkonversi data tersebut ke format JSON menggunakan library `ArduinoJson` dan mempublikasikannya ke topik MQTT `'robot/data'`. Sebaliknya, ketika pengguna mengirim perintah melalui dashboard Node-RED, perintah tersebut diterima oleh gateway melalui callback MQTT `'callback'` yang berlangganan ke topik `'robot/control'` dan `'robot/mode'`. Gateway kemudian mengemas perintah tersebut ke dalam struktur data `'Message'` dengan tipe `'C'` (Command) dan mengirimkannya ke robot melalui `'esp_now_send'`. Loop utama gateway secara kontinyu memeriksa koneksi MQTT dan memanggil `'client.loop()'` untuk memproses pesan masuk, dengan mekanisme reconnection otomatis jika terjadi gangguan koneksi.

Dashboard Node-RED berfungsi sebagai antarmuka pengguna yang memvisualisasikan data sensor dalam bentuk gauge dan chart, serta menyediakan kontrol untuk mengubah mode operasi dan menggerakkan robot. Ketika pengguna mengaktifkan mode patroli melalui switch di dashboard, perintah "P" dikirim ke topik MQTT `'robot/mode'`, yang kemudian diteruskan oleh gateway ke robot, mengaktifkan `TaskPatrol` untuk menjalankan navigasi otomatis. Selama patroli, robot secara berkala mengirim data suhu dan kelembaban yang ditampilkan secara real-time di dashboard. Pada mode manual, pengguna dapat mengendalikan robot melalui tombol D-Pad yang mengirim perintah gerak (F, B, L, R, S) ke topik `'robot/control'`, yang kemudian diteruskan ke robot dan diproses oleh `TaskControl` untuk menggerakkan motor dan servo sesuai perintah. Alur kerja ini memungkinkan sistem beroperasi secara real-time dengan latensi rendah, memisahkan tugas antara unit mobile dan gateway untuk meningkatkan keandalan, dan menyediakan akses data yang dapat diakses dari mana saja melalui platform IoT.

REFERENCES

- [1] Digilab, “Internet of Things | Digilab UI,” Digilabdte.com, 2018.
<https://learn.digilabdte.com/books/internet-of-things> (accessed Dec. 05, 2025).
- [2] P. Marwedel and Springerlink (Online Service, Embedded System Design : Embedded Systems Foundations of Cyber-Physical Systems, and the Internet of Things. Cham: Springer International Publishing, Imprint Springer, 2021. (accessed Dec. 05, 2025).
- [3] “FreeRTOS - Market leading RTOS (Real Time Operating System) for embedded systems with Internet of Things extensions,” Freertos.org, 2010.
<https://www.freertos.org/> (accessed Dec. 05, 2025).

- [4] MQTT, “MQTT - The Standard for IoT Messaging,” mqtt.org, 2024. <https://mqtt.org/> (accessed Dec. 05, 2025).
- [5] OpenJS Foundation, “Node-RED,” Nodered.org, 2019. <https://nodered.org/> (accessed Dec. 05, 2025).

APPENDICES

Appendix A: Project Schematic

Put your final project latest schematic here

Appendix B: Documentation

Put the documentation (photos) during the making of the project