



UNIVERSITAS  
INDONESIA

Veritas, Pudicit, Seputra

FAKULTAS

TEKNIK

FINAL PROJECT REAL TIME SYSTEM & IOT

# Smart Agri-Patrol Bot

*IoT-Based Agricultural Patrol UGV Using Real-Time  
Operating System with Node-RED*

Presented by GROUP 9 - WN

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





UNIVERSITAS  
INDONESIA

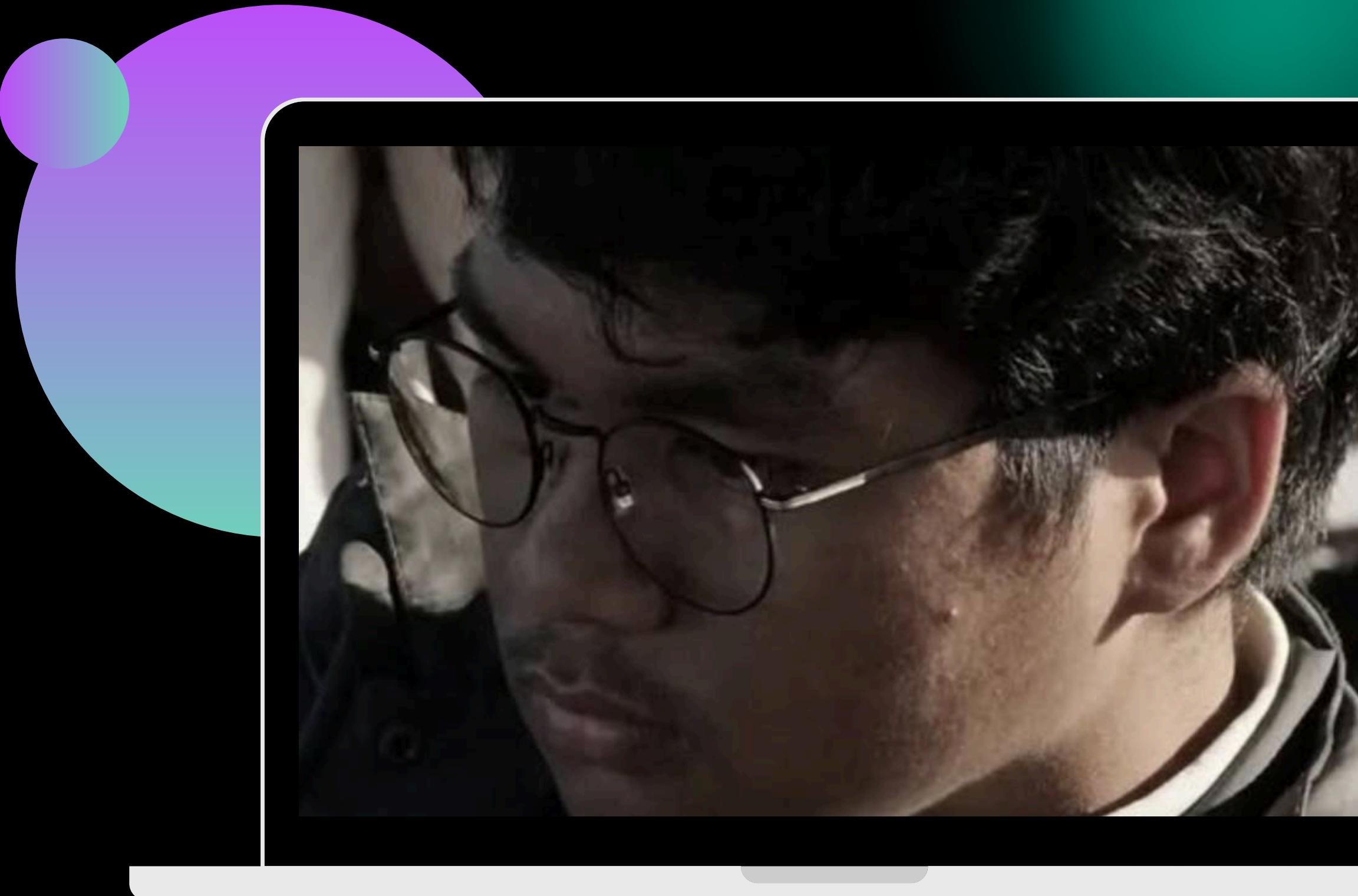
Veritas, Pudicitia, Scientia

FAKULTAS

TEKNIK

# Our Discussion

1. Latar Belakang Masalah
2. Solusi & Tujuan
3. Arsitektur Sistem
4. Perancangan Hardware
5. Implementasi Software
6. Protokol Komunikasi & IoT
7. Dashboard Node-RED
8. Logika Patroli
9. Hasil Pengujian & Kesimpulan



# Latar Belakang Masalah

Pertanian modern menghadapi tantangan besar dalam memantau kondisi lahan yang luas. Metode konvensional (pengecekan manual) memiliki banyak keterbatasan yang menghambat produktivitas dan respon cepat terhadap perubahan cuaca.

## 1 Inefisiensi Sumber Daya (Waktu, Tenaga, Biaya)

- Pengecekan manual memakan waktu dan tenaga yang signifikan, terutama di lahan luas atau sulit dijangkau.
- Membutuhkan alokasi SDM rutin yang meningkatkan biaya operasional secara konsisten.

## 2 Kualitas Data Rendah (Tidak Real-Time & Human Error)

- Frekuensi pengukuran rendah (jarang), padahal cuaca berubah cepat.
- Data tidak real-time, menyebabkan keterlambatan penanganan tanaman.
- Rentan terjadi kesalahan pencatatan atau pembacaan sensor oleh manusia.

## 3 Keterbatasan Cakupan & Urgensi Otomasi

- Pengecekan manual hanya terfokus pada area yang mudah dijangkau (cakupan terbatas).
- Diperlukan solusi teknologi (IoT & Robotika) untuk mengotomasi pemantauan secara presisi, luas, dan tanpa intervensi manusia terus-menerus.



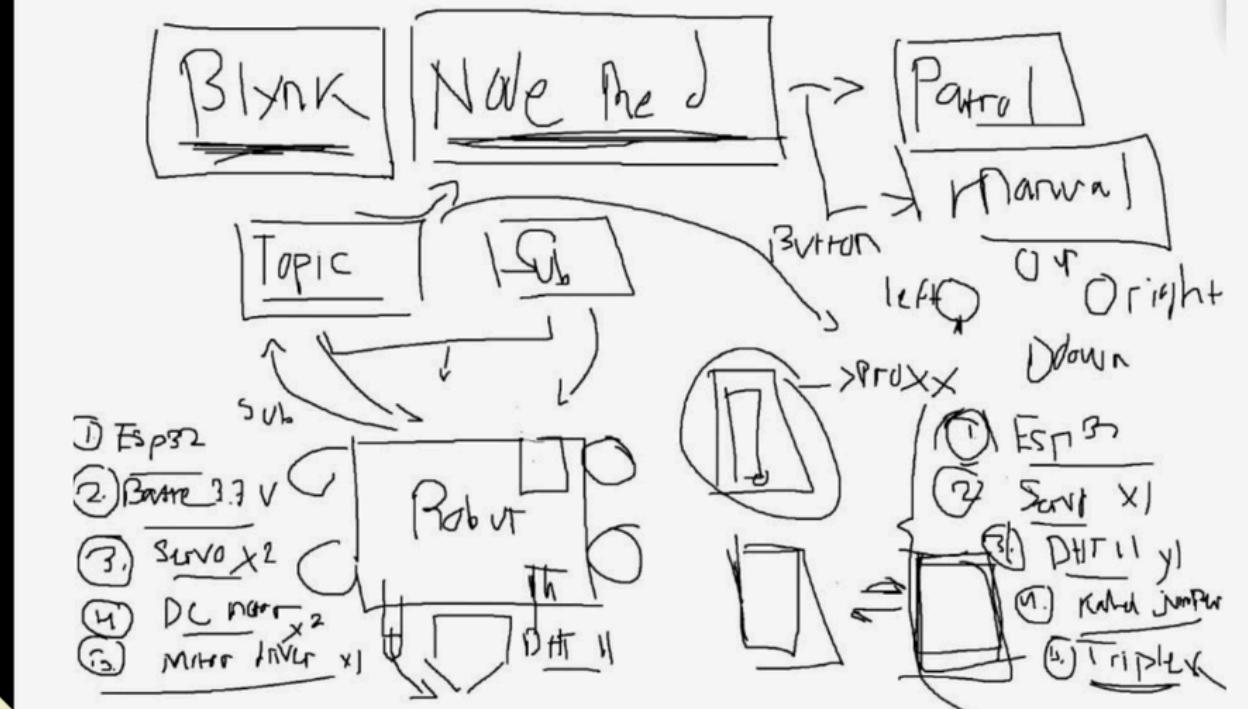
# Solusi dan Tujuan

## Solusi yang Diusulkan:

- Pengembangan robot Unmanned Ground Vehicle (UGV) otomotif berbasis IoT untuk memantau kondisi mikroklimat pertanian.
- Menggabungkan mikrokontroler (ESP32), sistem operasi real-time (FreeRTOS), dan komunikasi nirkabel (ESP-NOW & MQTT) ke dalam satu ekosistem yang terpadu.
- Sistem dibagi menjadi dua node terpisah, Node Robot (di lapangan) dan Node Gateway (jembatan internet), untuk menjaga stabilitas koneksi dan efisiensi daya.

## Tujuan Proyek

- Efisiensi Sumber Daya: Menggantikan pengecekan manual yang memakan waktu dengan patroli robotik otomatis yang mampu menjangkau area luas.
- Pemantauan Real-Time: Menyediakan data suhu dan kelembaban lahan secara live melalui Dashboard Node-RED untuk pengambilan keputusan yang cepat.
- Peningkatan Akurasi: Meminimalisir human error dalam pencatatan data lingkungan melalui sensor digital yang presisi.
- Fleksibilitas Kontrol: Memberikan opsi kontrol ganda: Otomatis (Patroli Pola Boustrophedon) dan Manual (Kendali Jarak Jauh) sesuai kebutuhan petani.



# Arsitektur Sistem

## Konsep Dual-Node

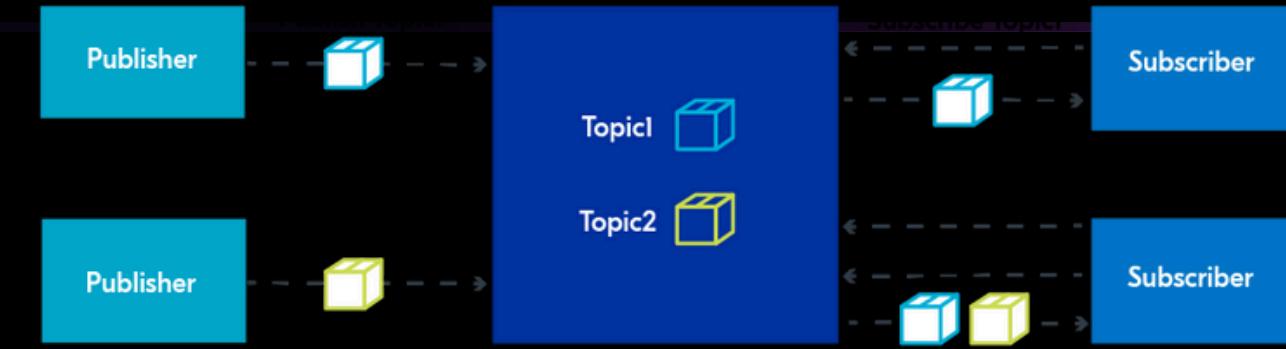
Sistem dirancang dengan memisahkan beban kerja menjadi dua unit independen untuk menjaga performa real-time robot.

- Node Robot (Mobile Unit): Fokus 100% pada tugas lapangan (gerak, sensor, aktuator) tanpa terbebani koneksi internet yang berat.
- Node Gateway (Base Station): Bertindak sebagai jembatan (bridge) khusus untuk koneksi internet dan manajemen data cloud.

## Alur Komunikasi & Protokol

### Menggunakan ESP-NOW dan MQTT

- Lokal (Robot <-> Gateway): Menggunakan ESP-NOW.
  - Kenapa? Protokol ini sangat cepat (low latency), hemat daya, dan tidak membutuhkan router WiFi di tengah sawah. Jangkauan bisa mencapai ratusan meter (line-of-sight).
- Cloud (Gateway <-> Internet): Menggunakan MQTT (via WiFi).
  - Kenapa? Protokol standar IoT yang ringan dan stabil untuk mengirim data telemetri ke dashboard.



## Peran Node Robot

Otak yang berjalan mandiri di lapangan untuk mengendalikan pergerakan dan membaca sensor secara real-time tanpa tergantung internet.

- Brain: ESP32 menjalankan FreeRTOS dengan 3 Task paralel (Control, Patrol, Comm).
- Input: Sensor Suhu & Kelembaban (DHT11).
- Output: Motor DC (Gerak Roda) dan Servo (Kemudi & Lengan).

## Peran Node Gateway

Jembatan penghubung yang bertugas menerima data dari robot dan meneruskannya ke internet (Cloud/Dashboard).

- Bridge: Menerima paket data struct dari ESP-NOW → Konversi ke JSON → Kirim ke MQTT Topic robot/data.
- Commander: Menerima perintah dari MQTT Topic robot/control → Konversi ke struct Command → Kirim ke Robot via ESP-NOW.



UNIVERSITAS  
INDONESIA

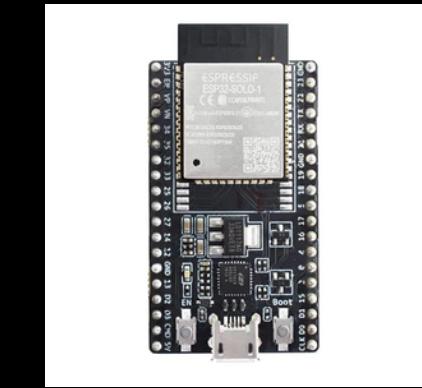
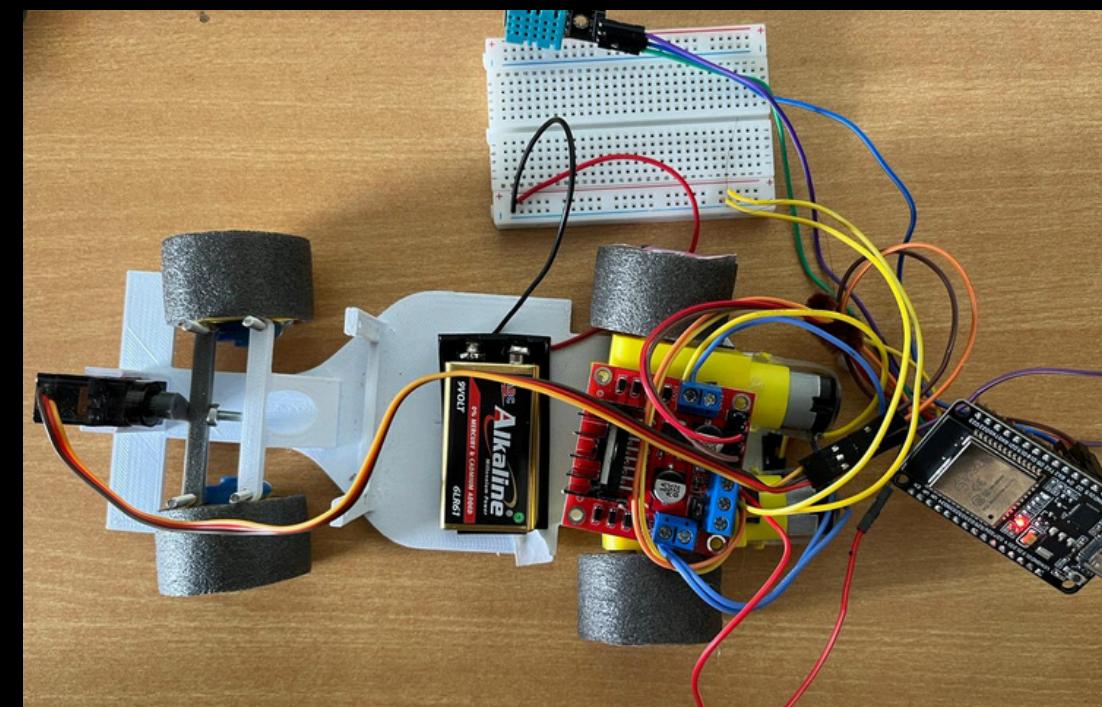
Veritas, Pudicit, Seputra

FAKULTAS

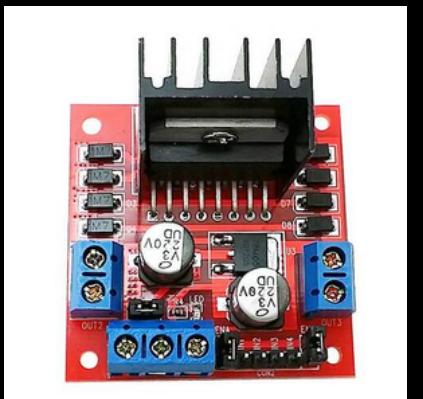
TEKNIK

# Perancangan Hardware

Desain hardware untuk sistem Smart Agri-Patrol Bot dirancang dengan arsitektur dual-ESP32 yang memisahkan fungsi mobile unit (robot) dan base station (gateway).



ESP32 Dev  
Kit



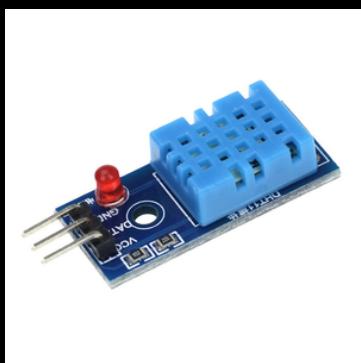
L29 Motor  
Driver



Servo  
SG90N



ESP32 Dev  
Kit



DHT11  
Sensor

# Perancangan Software

## *Platform & Framework*

- Bahasa: C++ dengan Framework Arduino.
  - Core System: Implementasi FreeRTOS pada ESP32 (Dual-Core).
  - Struktur: Program dibagi menjadi 3 Task independen yang berjalan secara paralel (Multitasking).

## Manajemen Task (FreeRTOS)

- TaskControl (Prioritas: TINGGI)
    - Fungsi: Menangani mode Manual dan Emergency Stop.
    - Kenapa Tinggi? Agar robot bisa langsung berhenti atau berbelok saat user menekan tombol, meskipun sedang sibuk melakukan hal lain.
  - TaskComm (Prioritas: MENENGAH)
    - Fungsi: Mengirim data antrian (Queue) ke Gateway via ESP-NOW.
    - Kenapa Menengah? Menjaga aliran data tetap lancar tanpa memblokir kontrol motor.
  - TaskPatrol (Prioritas: RENDAH)
    - Fungsi: Logika navigasi otomatis (Jalan → Stop → Sampling).
    - Kenapa Rendah? Ini adalah proses latar belakang yang bisa diinterupsi kapan saja oleh kontrol manual.

## *Inter-Task Communication*

- Menggunakan FreeRTOS Queues (commandQueue & sensorQueue) untuk pertukaran data antar task yang aman (thread-safe) tanpa menyebabkan race condition.

The screenshot shows the Arduino IDE interface with the title "Car-ESP32 | Arduino IDE 2.3.6". The menu bar includes File, Edit, Sketch, Tools, and Help. On the left, there are icons for file operations like Open, Save, and Print. The central workspace displays the following C++ code for a project named "Car-ESP32.ino":

```
#include <ESP32Servo.h>
#include <DHT.h>
#include <freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <freertos/queue.h>
#include <esp_now.h>
#include <WiFi.h>

// =====
// DEFINISI GLOBAL & SETUP
// =====

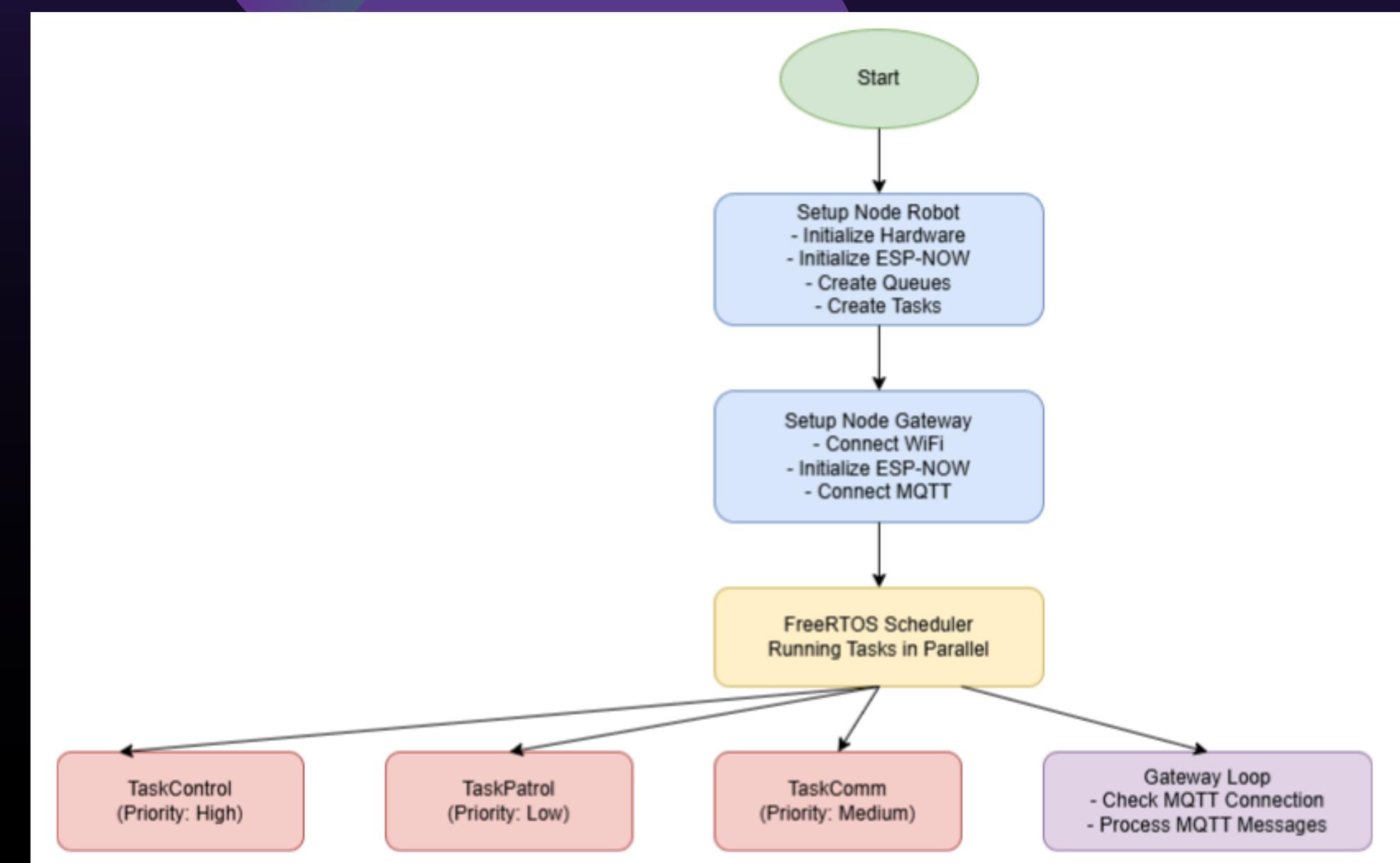
// --- 1. DEFINISI STRUKTUR DATA (WAJIB ADA DI FILE UTAMA) ---
typedef struct Message {
    char type;      // 'C' (Command) atau 'S' (Sensor)
    char cmd;       // F, B, L, R, S, P, M
    float temp;
    float hum;
} Message;

// -- Definisi Pin Hardware --
#define IN1 25 // Pin motor
#define IN4 26 // Pin motor
#define STEERING_SERVO_PIN 15 // Pin servo kemudi
#define SENSOR_ARM_SERVO_PIN 12 // Pin servo lengan sensor
#define DHT_PIN 4           // Pin sensor DHT11
#define DHT_TYPE DHT11

// -- Konfigurasi Global --
// GANTI DENGAN ALAMAT MAC GATEWAY ANDA
uint8_t gatewayAddress[] = {0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF};
```

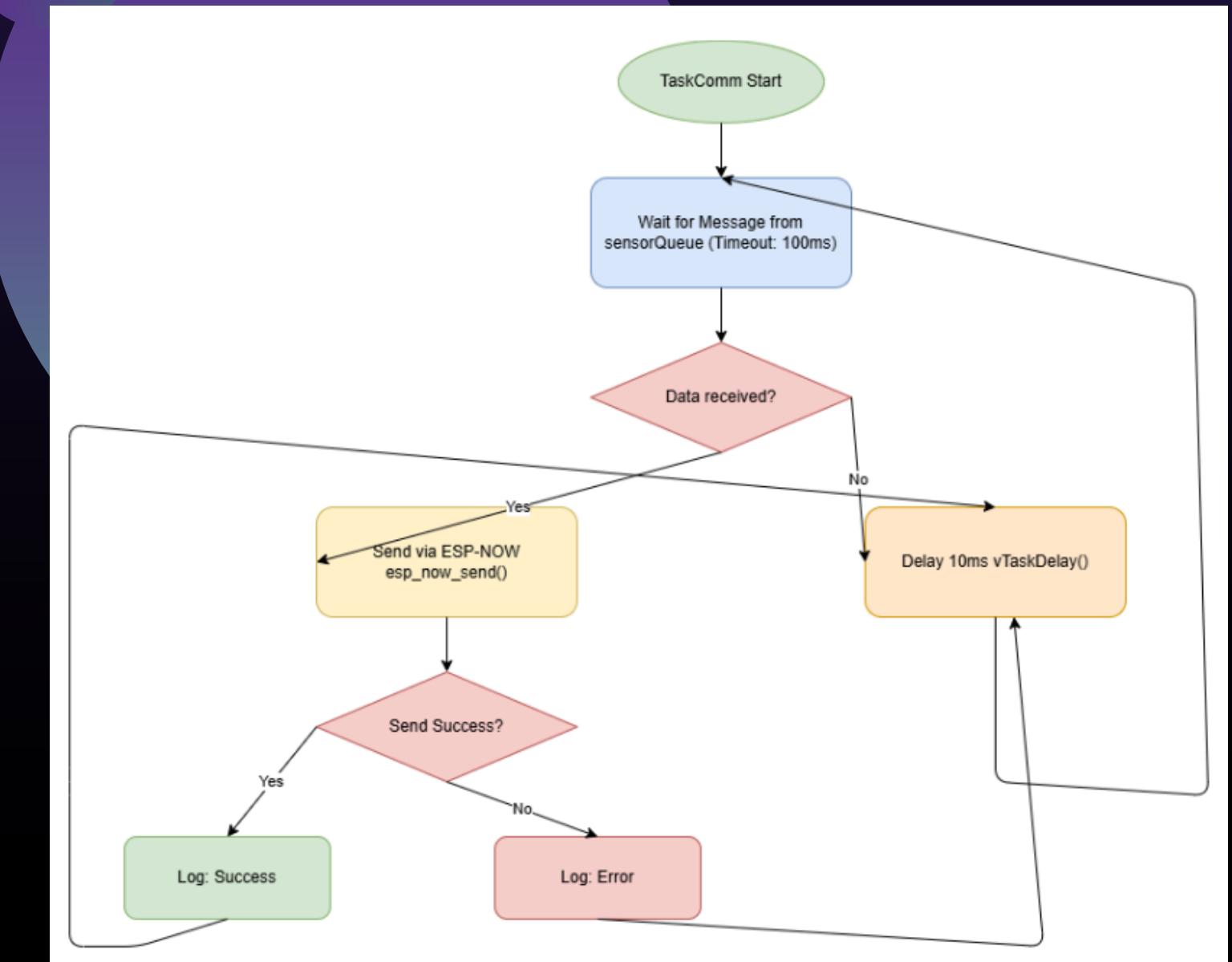
# Perancangan Software

## *System Flowchart*



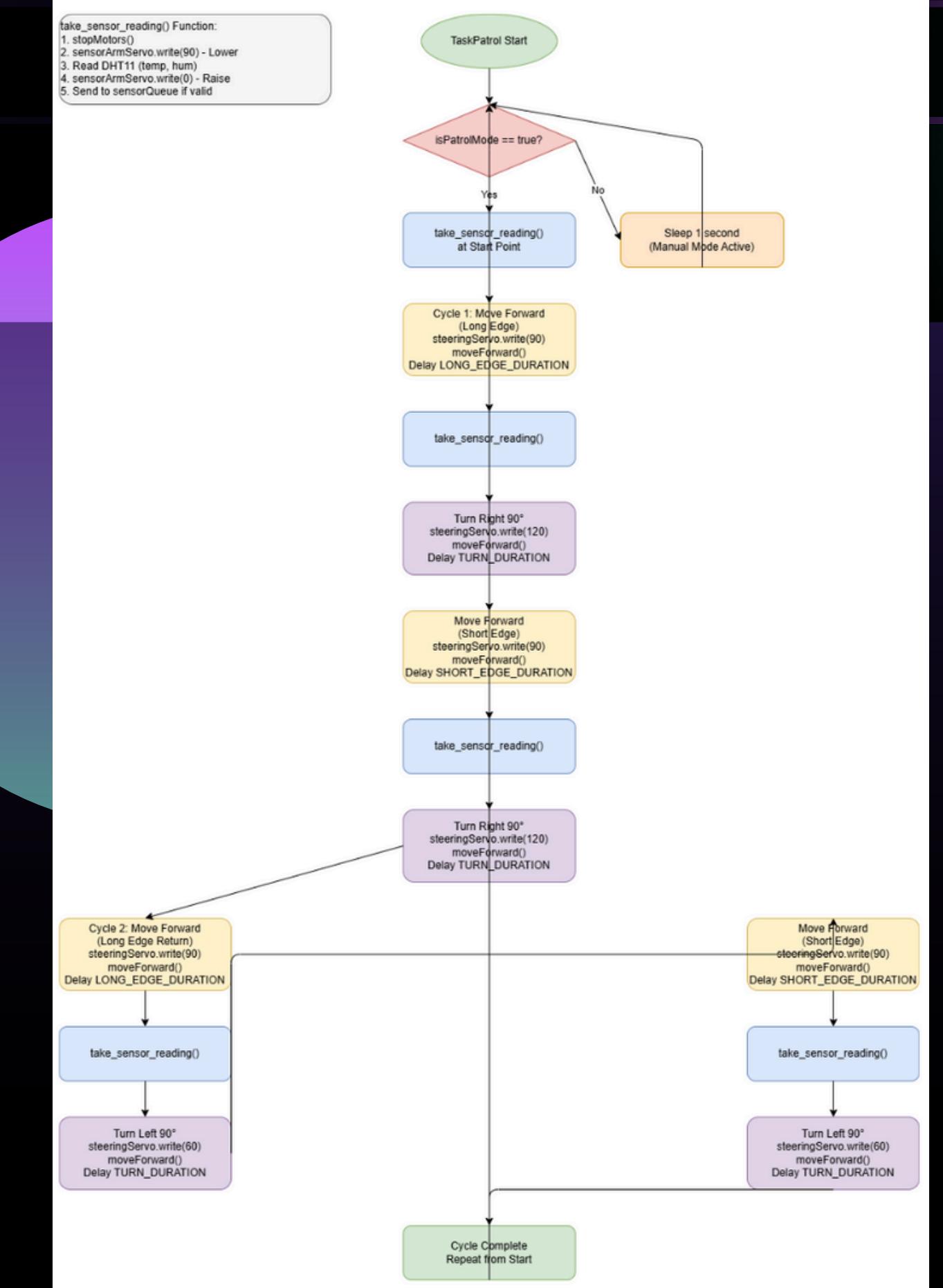
# Perancangan Software

## Communication System Flowchart



# Perancangan Software

## *Patrol System Flowchart*



# Perancangan Software

## *Control System Flowchart*



# Protokol Komunikasi & IoT

## Data Flow

- Step 1 (Akuisisi): Sensor DHT11 membaca suhu & kelembaban → Dikirim ke ESP32 Robot.
- Step 2 (Lokal/Wireless): Robot mengirim paket data struct ke Gateway via ESP-NOW.
- Step 3 (Internet/Cloud): Gateway mengkonversi data ke JSON → Publish ke MQTT Broker (Topik: robot/data).
- Step 4 (Visualisasi): Dashboard Node-RED menerima data (Subscribe) → Tampilkan di Gauge/Chart.

## Protokol

### ESP NOW (Robot <-> Gateway)

- Jenis: Peer-to-Peer (P2P), tanpa koneksi WiFi Router.
- Kelebihan: Latensi sangat rendah (<10ms), hemat daya, jangkauan jauh (hingga ratusan meter line-of-sight).
- Format Data: Struct C++ (char type, float temp, float hum).

## Keamanan & Efisiensi

- Isolasi Jaringan: Robot tidak terekspos langsung ke internet (lebih aman dari serangan luar).
- Optimasi Payload: Data antar-ESP dikirim dalam bentuk binary struct (kecil & cepat), baru diubah ke JSON di Gateway (mudah dibaca sistem lain).

### MQTT (Gateway <-> Cloud)

- Jenis: Publish/Subscribe via TCP/IP.
- Kelebihan: Ringan (lightweight), hemat bandwidth, standar industri IoT.
- Broker: HiveMQ Cloud (Public Broker).



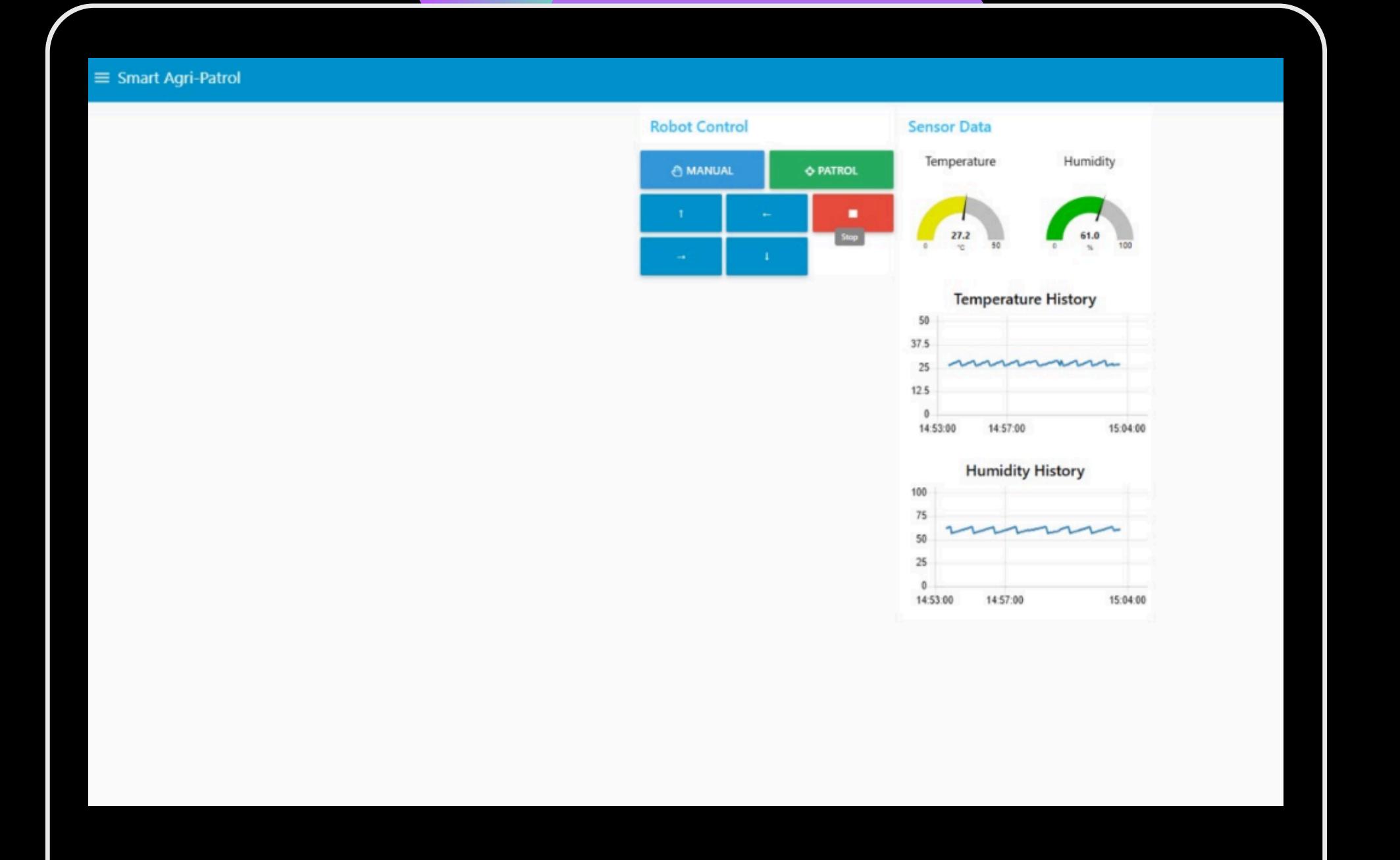
UNIVERSITAS  
INDONESIA

Virtus, Prudentia, Justitia

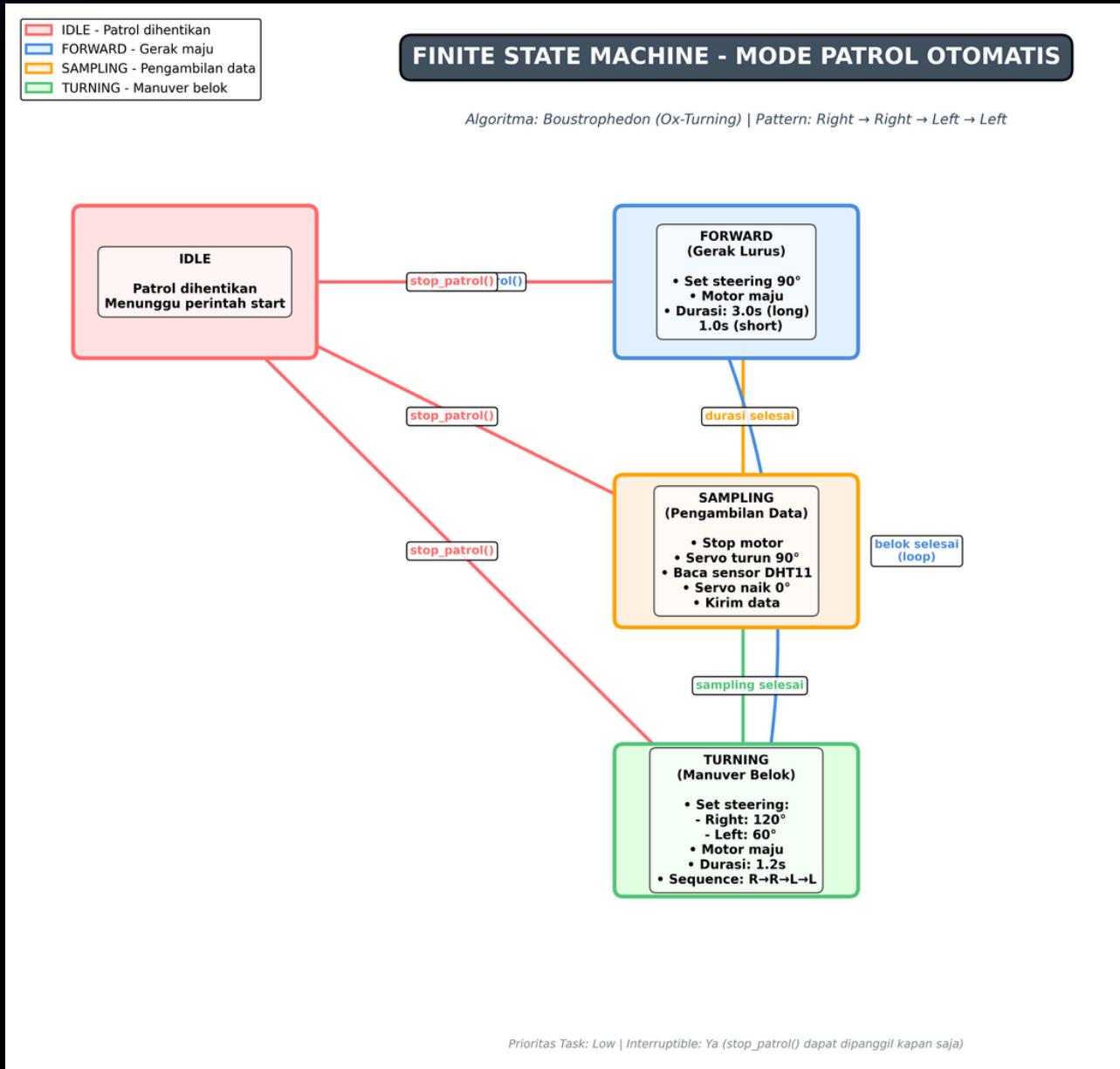
FAKULTAS

TEKNIK

# Dashboard Node-RED



# Logika Patroli



## Step 1

Forward

## Step 2

Sampling

## Step 3

Turning

Pada mode manual jika user menekan tombol stop di dashboard task ini otomatis akan diberhentikan



UNIVERSITAS  
INDONESIA

Veritas, Pudicit, Seputra

FAKULTAS

TEKNIK

# Hasil Pengujian

## Serial Monitor

```
macaddressrobotchecker.ino
13 WiFi.mode(WIFI_STA);
14 delay(100);
15
16 // Method 1: WiFi.macAddress()
17 Serial.println("MAC Address Methods:");
18 Serial.println("-----");
19 Serial.print("Method 1 (WiFi.macAddress): ");
20 Serial.println(WiFi.macAddress());
21
22 // Method 2: esp_wifi_get_mac()
23 uint8_t mac[6];
24 esp_wifi_get_mac(WIFI_IF_STA, mac);

Output Serial Monitor X
Message [Enter to send message to 'ESP32 Dev Module' on 'COM6']

14:39:24.565 -> rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
14:39:24.565 -> configSip 0, SPIWP:0xee
14:39:24.565 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
14:39:24.565 -> mode:DIO, clock div:1
14:39:24.565 -> load:0x3fff0030,len:4980
14:39:24.565 -> load:0x40078000,len:16612
14:39:24.565 -> load:0x40080400,len:3480
14:39:24.565 -> entry 0x400805b4
14:39:24.984 -> Robot MAC Address: 00:00:00:00:00:00
14:39:24.984 -> Robot WiFi Channel: 6
14:39:24.984 -> Gateway peer added successfully!
14:39:24.984 -> Target Gateway MAC: D0:EF:76:34:12:C4
14:39:24.984 -> setup complete.
14:42:09.422 -> Patrol: Sampling...
14:42:17.649 -> Patrol: Sampling...
14:42:27.209 -> Patrol: Sampling...
14:43:29.517 -> Patrol: Sampling...
14:43:37.752 -> Patrol: Sampling...
14:43:47.342 -> Patrol: Sampling...
14:43:51.518 -> Patrol: Sampling...
```

Manual Mode

```
car-esp32 | Arduino IDE 2.3.6
File Edit Sketch Tools Help
car-esp32.ino Patrol.ino Communication.ino
void TaskControl(void *pvParameters) {
    Message msg;
    int steerAngle = 90;
    for (;;) {
        if (xQueueReceive(commandQueue, &msg, portMAX_DELAY) == pdPASS) {
            if (msg.cmd == 'P') { isPatrolMode = true; stopMotors(); steeringServo.write(90); }
            else if (msg.cmd == 'M') { isPatrolMode = false; stopMotors(); steeringServo.write(90); }

            if (!isPatrolMode) {
                if (msg.cmd == 'F') moveForward();
                else if (msg.cmd == 'B') moveBackward();
                else if (msg.cmd != 'L' && msg.cmd != 'R') stopMotors();

                if (msg.cmd == 'L') steerAngle = 60;
                else if (msg.cmd == 'R') steerAngle = 120;
                else steerAngle = 90;
                steeringServo.write(steerAngle);
            }
        }
    }
}

Output Serial Monitor X
Message [Enter to send message to 'ESP32 Dev Module' on 'COM6']

14:50:49.077 -> Patrol: Sampling...
14:50:50.581 -> Patrol: Mock Data - Temp: 27.50°C, Hum: 59.00%
14:50:51.574 -> Patrol: Queued Data
14:50:51.620 -> >> Comm: Data sent to Gateway
14:50:53.179 -> Manual Mode - Mock Data - Temp: 28.40°C, Hum: 60.50%
14:50:53.179 -> >> Comm: Data sent to Gateway
14:50:55.179 -> Manual Mode - Mock Data - Temp: 28.70°C, Hum: 61.>> Comm: Data sent to Gateway
14:50:55.179 -> 00%
14:51:01.184 -> Manual Mode - Mock Data - Temp: 26.00°C, Hum: 61>> Comm: Data sent to Gateway
14:51:03.184 -> .50%
```

Patrol Mode



UNIVERSITAS  
INDONESIA

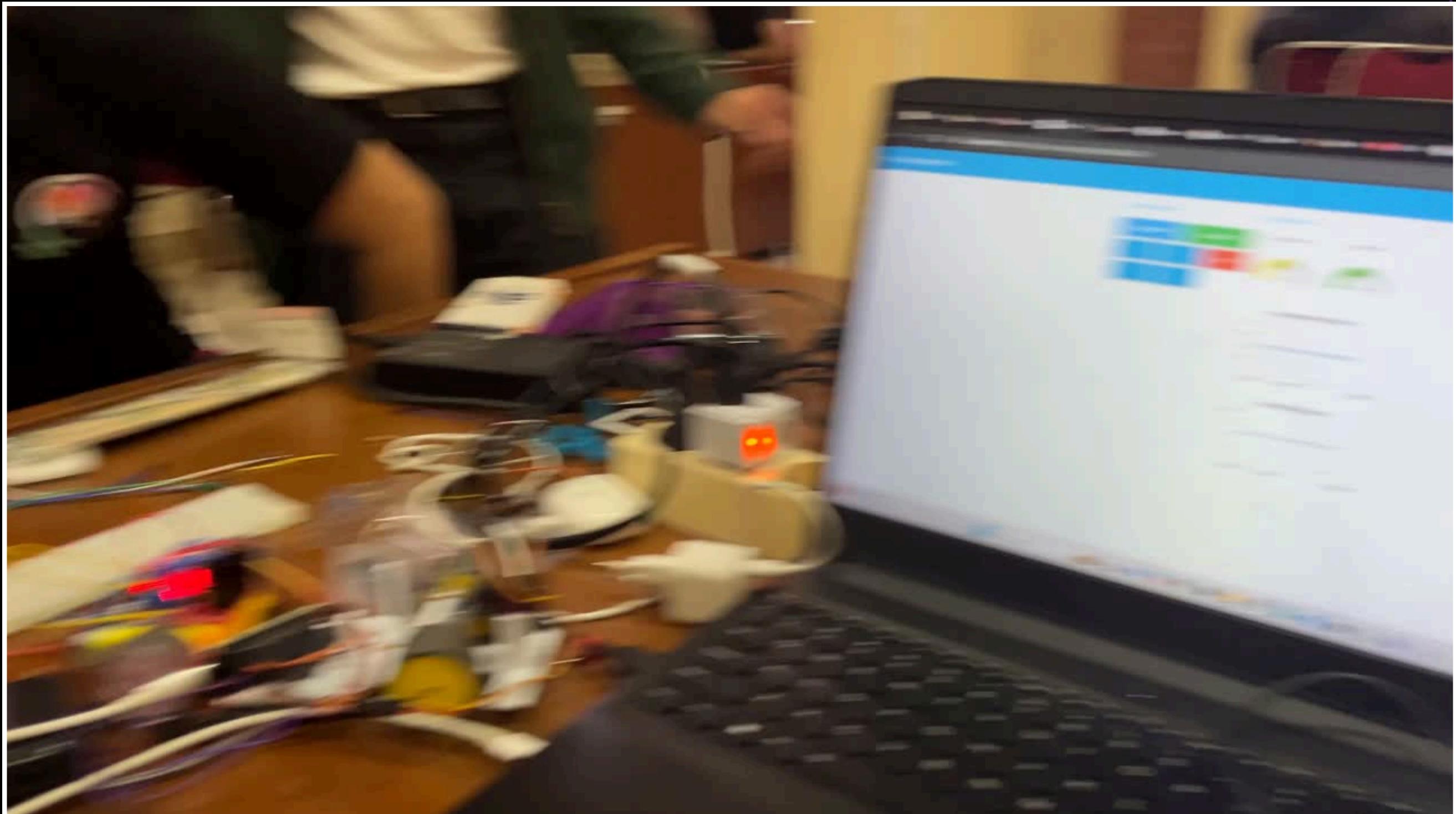
Veritas, Praelatio, Scientia

FAKULTAS

TEKNIK

## FINAL PROJECT REAL TIME SYSTEM & IOT

# Demo





UNIVERSITAS  
INDONESIA

Veritas, Pudicit, Seputra

FAKULTAS

TEKNIK

# Kesimpulan

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.

Node Gateway berfungsi sebagai jembatan komunikasi. Ia menerima data sensor via ESP-NOW, mengonversinya ke JSON, dan memublikasikannya ke topik MQTT (robot/data). Sebaliknya, perintah pengguna dari Node-RED (melalui topik robot/control dan robot/mode) diterima via MQTT, dikemas, dan diteruskan ke robot melalui esp\_now\_send.

Dashboard Node-RED menyediakan antarmuka pengguna untuk visualisasi data (suhu/kelembaban) dan kontrol. Pengguna dapat mengirim perintah mode (Patroli "P") atau gerak manual (F, B, L, R, S) melalui MQTT. Alur kerja ini menjamin operasi real-time dengan latensi rendah dan akses data melalui platform IoT.

# Thank You!

## *Referensi:*

- [1] Digilab, "Internet of Things | Digilab UI," [Digilabdte.com](https://learn.digilabdte.com/books/internet-of-things), 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](https://www.freertos.org/), 2010. <https://www.freertos.org/> (accessed Dec. 05, 2025).
- [4] MQTT, "MQTT - The Standard for IoT Messaging," [mqtt.org](https://mqtt.org/), 2024. <https://mqtt.org/> (accessed Dec. 05, 2025).
- [5] OpenJS Foundation, "Node-RED," [Nodered.org](https://nodered.org/), 2019. <https://nodered.org/> (accessed Dec. 05, 2025).