



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

**Seize the Class: Smart Classroom Occupancy Control System**

**GROUP 7**

<b>AISYAH ARIFATUL ALYA</b>	<b>2206059383</b>
<b>ANNISA ARDELIA SETIAWAN</b>	<b>2206059471</b>
<b>ARIA BIMA SAKTI</b>	<b>2206062970</b>
<b>PHOEBE IVANA</b>	<b>2206820320</b>

## PREFACE

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa atas segala limpahan rahmat dan karunia-Nya, sehingga kami dapat menyelesaikan makalah ini yang berjudul *"Seize the Class: Smart Classroom Occupancy Control System"*. Makalah ini disusun untuk memberikan solusi inovatif terhadap permasalahan efisiensi penggunaan energi listrik di ruang kuliah dengan memanfaatkan teknologi modern berbasis Internet of Things (IoT) dan mesh network.

Melalui makalah ini, kami berupaya menghadirkan sistem yang dapat mengoptimalkan pengendalian perangkat listrik, seperti lampu dan AC, secara otomatis dan terintegrasi. Sistem ini tidak hanya dirancang untuk meningkatkan efisiensi energi tetapi juga untuk memberikan kemudahan pengawasan dan kontrol bagi pengelola gedung. Harapan kami, solusi yang kami tawarkan dapat menjadi langkah nyata dalam mendukung keberlanjutan lingkungan melalui pengurangan pemborosan energi.

Kami menyadari bahwa makalah ini tidak akan terwujud tanpa bimbingan, dukungan, dan masukan dari berbagai pihak. Untuk itu, kami mengucapkan terima kasih kepada dosen pembimbing, rekan-rekan, serta semua pihak yang telah memberikan kontribusi dalam proses penyusunan makalah ini. Kami menyadari bahwa makalah ini masih jauh dari sempurna. Oleh karena itu, kami sangat terbuka terhadap saran dan kritik yang membangun untuk pengembangan lebih lanjut.

Semoga makalah ini dapat memberikan manfaat dan inspirasi, khususnya bagi para pembaca yang tertarik pada pengelolaan energi berbasis teknologi.

Depok, December 09, 2024

Group 7

## TABLE OF CONTENTS

<b>CHAPTER 1.....</b>	<b>4</b>
<b>INTRODUCTION.....</b>	<b>4</b>
1.1 PROBLEM STATEMENT.....	4
1.2 PROPOSED SOLUTION.....	5
1.3 ACCEPTANCE CRITERIA.....	6
1.4 OBJECTIVES.....	6
1.5 SCOPE AND LIMITATIONS.....	7
1.6 ROLES AND RESPONSIBILITIES.....	7
1.7 TIMELINE AND MILESTONES.....	8
<b>CHAPTER 2.....</b>	<b>9</b>
<b>IMPLEMENTATION.....</b>	<b>9</b>
2.1 HARDWARE DESIGN AND SCHEMATIC.....	9
2.2 SOFTWARE DEVELOPMENT.....	10
2.2.1 Leaf node.....	11
2.2.2 Root Node.....	12
2.3 HARDWARE AND SOFTWARE INTEGRATION.....	13
2.4 USER INTERFACE DESIGN.....	15
<b>CHAPTER 3.....</b>	<b>17</b>
<b>TESTING AND EVALUATION.....</b>	<b>17</b>
3.1 TESTING.....	17
3.2 RESULT.....	17
3.3 EVALUATION.....	18
<b>CHAPTER 4.....</b>	<b>20</b>
<b>CONCLUSION.....</b>	<b>20</b>
4.1 SUMMARY.....	20
4.2 FUTURE WORKS.....	20
<b>REFERENCES.....</b>	<b>22</b>
<b>APPENDICES.....</b>	<b>23</b>

## CHAPTER 1

### INTRODUCTION

#### 1.1 PROBLEM STATEMENT

Penggunaan energi listrik yang tidak efisien pada ruang kuliah di institusi pendidikan merupakan salah satu permasalahan yang sering kali diabaikan tetapi memiliki dampak yang signifikan. Lampu dan AC, yang menjadi kebutuhan utama untuk kenyamanan proses belajar mengajar, sering kali dibiarkan menyala meskipun ruangan tidak sedang digunakan atau hanya digunakan secara parsial. Kondisi ini tidak hanya menyebabkan pemborosan energi tetapi juga meningkatkan biaya operasional secara keseluruhan. Ketergantungan pada sistem manual untuk mengelola perangkat listrik di ruang kuliah semakin memperburuk situasi, karena sistem tersebut kurang mendukung upaya pengelolaan energi yang efektif. Akibatnya, institusi harus menghadapi biaya operasional yang tidak perlu sekaligus memberikan kontribusi pada peningkatan emisi karbon, yang berdampak negatif terhadap lingkungan.

Selain itu, sistem manual yang saat ini umum digunakan oleh pengelola gedung tidak menawarkan fleksibilitas maupun efisiensi dalam pengawasan dan pengendalian perangkat listrik di setiap ruangan. Pengelola seringkali kesulitan memantau kondisi lampu dan AC di banyak ruangan secara *real-time*, sehingga sulit untuk memastikan perangkat tersebut hanya aktif saat dibutuhkan. Ketidakmampuan untuk memantau dan mengontrol perangkat listrik secara efisien tidak hanya membebani anggaran institusi, tetapi juga menghambat upaya untuk menerapkan pengelolaan energi yang lebih ramah lingkungan. Dalam jangka panjang, hal ini dapat menciptakan ketergantungan pada pola konsumsi energi yang boros, yang tidak sejalan dengan upaya keberlanjutan yang kini menjadi perhatian global.

Dengan tantangan ini, menjadi jelas bahwa diperlukan sebuah solusi yang inovatif dan modern untuk mengatasi masalah ini. Solusi tersebut harus mampu mengintegrasikan teknologi otomatisasi dan Internet of Things (IoT) untuk menciptakan sistem yang dapat memantau, mengendalikan, dan mengoptimalkan penggunaan perangkat listrik di ruang kuliah secara efektif. Dengan pendekatan yang lebih terintegrasi, institusi pendidikan tidak hanya dapat mengurangi pemborosan

energi dan biaya operasional, tetapi juga berkontribusi pada pelestarian lingkungan melalui pengurangan emisi karbon. Penerapan sistem seperti ini juga akan membantu institusi dalam menerapkan kebijakan pengelolaan energi yang lebih berkelanjutan dan responsif terhadap kebutuhan masa depan.

## 1.2 PROPOSED SOLUTION

Untuk mengatasi permasalahan penggunaan energi listrik yang tidak efisien di ruang kuliah, diperlukan sebuah sistem yang dirancang untuk mengotomatisasi pengendalian perangkat listrik seperti lampu dan AC. Sistem yang kami buat bernama ***Seize the Class: Smart Classroom Occupancy Control System***. Sistem ini memanfaatkan teknologi *mesh network* berbasis *painlessMesh* untuk menghubungkan perangkat secara efisien dan *platform* IoT Blynk sebagai *interface monitoring* dan kontrol. Dua *mode* operasi utama disediakan, yaitu ***Auto Mode*** dan ***Override Mode***. Pada *Auto Mode*, pengendalian lampu dan AC dilakukan secara otomatis berdasarkan deteksi gerakan menggunakan sensor RCWL0516. Sistem ini menggunakan *motion counter* untuk memastikan bahwa lampu dan AC hanya menyala selama ruangan aktif digunakan, dan akan mati setelah durasi yang ditentukan. Sementara itu, pada *Override Mode*, pengelola dapat mengambil alih kontrol secara manual, di mana lampu dan AC tetap menyala hingga dinonaktifkan secara manual oleh pengelola tanpa dipengaruhi oleh deteksi gerakan.

Sistem ini juga dilengkapi dengan sensor DHT11 untuk memantau suhu dan kelembaban di dalam ruangan, dengan data yang ditampilkan secara *real-time* melalui platform Blynk *Console*. Teknologi *mesh network* memungkinkan komunikasi secara *reliable* antara *Root Node*, yang terhubung dengan Wi-Fi untuk menerima data sensor dan mengirimkan perintah kontrol, serta *Leaf Node*, yang mengintegrasikan sensor dan aktuator seperti lampu dan AC. Dengan solusi ini, sistem diharapkan dapat mengoptimalkan penggunaan energi listrik, memberikan fleksibilitas dalam pengawasan dan kontrol perangkat listrik, serta mendukung keberlanjutan lingkungan dengan mengurangi pemborosan energi secara signifikan.

### 1.3 ACCEPTANCE CRITERIA

Berikut adalah *acceptance criteria* dari proyek kami:

1. Sistem dapat mendeteksi gerakan menggunakan sensor RCWL0516 dan menyalakan lampu/AC secara otomatis.
2. *Timer* dapat mematikan lampu/AC setelah jangka waktu tertentu, dan jika tidak ada gerakan baru yang terdeteksi.
3. Saat *mode override* aktif, maka lampu dan AC hanya bisa dinyalakan/dimatikan secara *manual* melalui tombol pada Blynk *Console*.
4. Pada Blynk *Console*, terdapat tombol untuk mengganti *mode*, yaitu antara *Auto Mode* dan *Override Mode*. Terdapat juga tombol untuk menyalakan/mematikan lampu dan AC saat dalam *mode override*.
5. Sistem dapat menampilkan data suhu, kelembapan, status lampu/AC, dan status ruangan (*Occupied/Not Occupied*) pada Serial Monitor.
6. Data dan perintah dapat dikirimkan secara antara *Leaf Node* dan *Root Node* melalui jaringan *painlessMesh*.

### 1.4 OBJECTIVES

Berikut adalah *objectives* dari proyek kami:

1. Mengurangi pemborosan energi listrik di ruang kuliah melalui sistem otomatisasi berbasis IoT.
2. Memungkinkan pengawasan dan pengendalian perangkat listrik secara *real-time* melalui Blynk *Console*.
3. Mendukung pengurangan emisi karbon untuk keberlanjutan lingkungan.
4. Menyediakan fleksibilitas dengan dua mode operasi, yaitu *Auto Mode* dan *Override Mode*.
5. Menggunakan teknologi *mesh network* yang memungkinkan ekspansi sistem secara mudah.

## 1.5 SCOPE AND LIMITATIONS

*Scope* dari proyek kami adalah sebagai berikut:

1. Sistem dirancang untuk ruang kuliah dengan fokus pada perangkat lampu dan AC.
2. Menggunakan teknologi *painlessMesh* dan platform Blynk untuk komunikasi antar perangkat.
3. Menyediakan monitoring data suhu, kelembapan, dan status perangkat secara real-time.
4. Mendukung dua mode operasi untuk fleksibilitas pengendalian perangkat.

*Limitations* dari proyek kami adalah sebagai berikut:

1. Memerlukan koneksi internet yang stabil untuk fungsi monitoring melalui Blynk *Console*.
2. Jangkauan *mesh network* terbatas pada kekuatan sinyal perangkat ESP32.
3. Sensor gerak tidak dapat membedakan jenis gerakan secara spesifik.
4. Sistem belum dirancang untuk lingkungan ekstrem yang dapat mempengaruhi performa.
5. Instalasi awal memerlukan pengetahuan teknis tentang jaringan *painlessMesh* dan perangkat keras.

## 1.6 ROLES AND RESPONSIBILITIES

Berikut ini merupakan *roles* dan *responsibilities* dari anggota kelompok kami:

Roles	Responsibilities	Person
Leader	<ul style="list-style-type: none"><li>● Membuat kode sistem</li><li>● Membeli komponen proyek</li><li>● Merakit rangkaian fisik</li><li>● Mengkonfigurasi Blynk <i>Console</i></li><li>● Melakukan uji coba <i>hardware</i></li><li>● Melakukan <i>debugging</i> kode</li><li>● Menulis laporan proyek</li></ul>	Aisyah Arifatul Alya

Member	<ul style="list-style-type: none"> <li>• Menulis laporan proyek</li> <li>• Membuat presentasi PowerPoint</li> <li>• Membuat dokumentasi <i>readme</i></li> <li>• Membuat diagram <i>flowchart</i> &amp; skematik</li> </ul>	Annisa Ardelia Setiawan
Member	<ul style="list-style-type: none"> <li>• Menulis laporan proyek</li> <li>• Membuat presentasi <i>PowerPoint</i></li> </ul>	Aria Bima Sakti
Member	<ul style="list-style-type: none"> <li>• Mengembangkan kode sistem</li> <li>• Melakukan <i>debugging</i> kode</li> <li>• Menulis laporan proyek</li> </ul>	Phoebe Ivana

## 1.7 TIMELINE AND MILESTONES

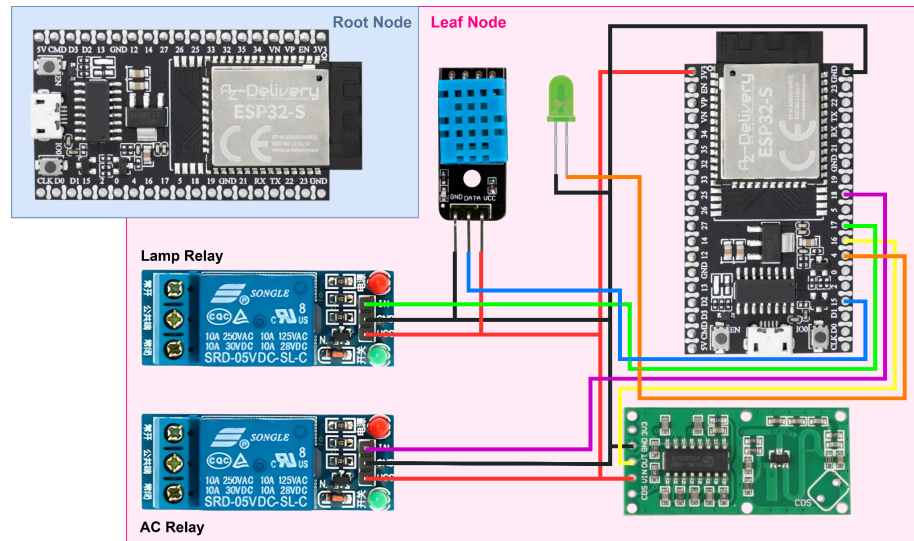
	November 2024							Desember 2024												
Task	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13
Brainstorming idea																				
Software Development																				
Hardware Designing																				
Software and Hardware Integration																				
Testing and Revising																				
Final Product Assembly																				
Making Project Report																				
Presentation																				



## CHAPTER 2

### IMPLEMENTATION

#### 2.1 HARDWARE DESIGN AND SCHEMATIC



Dalam perancangan hardware untuk proyek *Seize the Class: Smart Classroom Occupancy Control System*, kami menggunakan beberapa komponen utama, yaitu 2 ESP32, DHT11 sensor, RCWL-0516 motion sensor, 2 *Relay Module*, dan 1 LED. Sistem ini dirancang untuk mengontrol lampu dan AC secara otomatis berdasarkan deteksi gerakan dan kondisi lingkungan, serta menampilkan data ke dalam jaringan menggunakan *mesh communication*.

Kami menggunakan ESP32 sebagai pengendali utama yang mengelola komunikasi, pengolahan data sensor, dan kontrol perangkat output. Adapun sensor DHT11 digunakan untuk membaca suhu dan kelembapan ruangan. Sensor ini terhubung ke pin GPIO 15 pada ESP32. Data suhu dan kelembapan kemudian diproses oleh ESP32 untuk memantau kondisi lingkungan.

*Motion* sensor RCWL-0516 bekerja dengan mendeteksi perubahan gelombang mikro di sekitar. Ketika gerakan terdeteksi, sensor akan memberikan sinyal HIGH ke ESP32 melalui pin GPIO 16. Data ini digunakan untuk mengaktifkan lampu dan AC melalui *relay module*. LED digunakan untuk memberikan indikasi secara visual yang merepresentasikan adanya deteksi gerakan. LED terhubung ke pin GPIO 4 dan akan menyala ketika gerakan terdeteksi oleh RCWL-0516.

Selanjutnya, terdapat dua *relay* yang digunakan untuk mengontrol perangkat keluaran, yaitu lampu dan AC. *Relay* 1 diatur oleh pin GPIO 17 untuk lampu sedangkan *relay* 2 diatur oleh pin GPIO 18 untuk AC. *Relay* akan aktif ketika ESP32 menerima sinyal gerakan atau saat kondisi lingkungan tertentu terpenuhi.

Ketika *motion* sensor RCWL-0516 mendeteksi gerakan, sensor ini akan memberikan sinyal ke ESP32. Jika gerakan terdeteksi, ESP32 mengaktifkan *relay* untuk menyalakan lampu dan AC serta menyalakan LED indikator. Suhu dan kelembapan yang dibaca dari DHT11 akan ditampilkan dalam jaringan melalui *mesh communication*, sehingga data dapat dimonitor oleh perangkat lain. Sistem ini juga memiliki *mode* otomatis dan manual yang dapat diatur melalui komunikasi mesh.

Komunikasi antara *leaf node* dengan *root node* dilakukan dengan menggunakan jaringan mesh berbasis ESP32, di mana *leaf node* bertugas mengumpulkan data dari sensor, seperti suhu dan kelembapan dari DHT11, serta mendeteksi gerakan menggunakan RCWL-0516. Data tersebut kemudian dikirimkan ke *root node* melalui protokol mesh. Adapun *root node* berfungsi sebagai pengolah dan penerus data ke server MQTT untuk keperluan pemantauan jarak jauh. Dalam sistem ini, *leaf node* akan memastikan bahwa data lingkungan terkini selalu tersedia, sementara *root node* menjadi penghubung utama antara jaringan lokal dengan server, memungkinkan kontrol otomatis dan analisis data secara terpusat.

## 2.2 SOFTWARE DEVELOPMENT

Software Development untuk *Seize the Class: Smart Classroom Occupancy Control System* dilakukan dengan pendekatan yang terstruktur untuk memastikan sistem dapat berfungsi secara optimal sesuai dengan kebutuhan. Proses pengembangan ini mencakup beberapa tahapan utama, mulai dari analisis kebutuhan, perancangan, implementasi, hingga pengujian dan pemeliharaan. Setiap tahapan dirancang untuk menghasilkan sistem yang efisien, andal, dan mudah digunakan, sehingga mampu memberikan solusi yang efektif dalam mengoptimalkan penggunaan energi listrik di ruang kuliah.

### 2.2.1 Leaf node

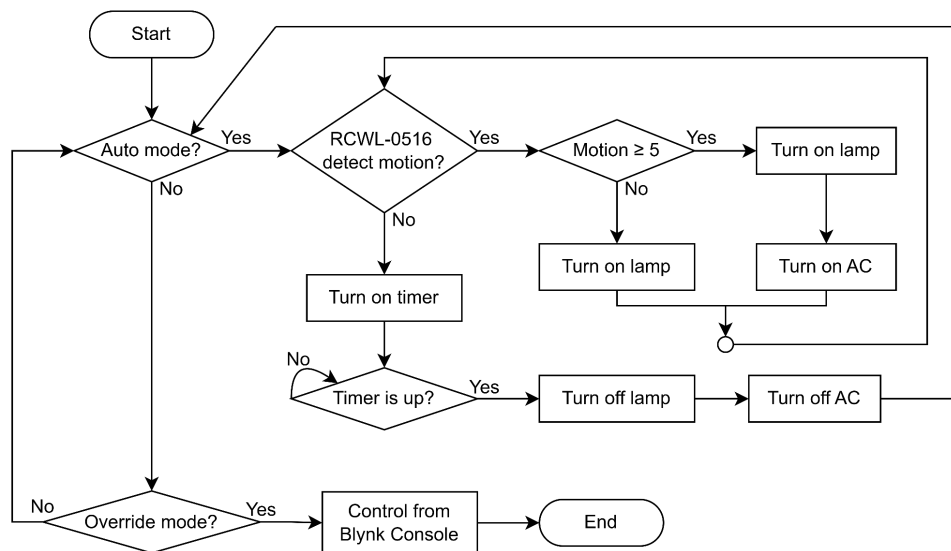
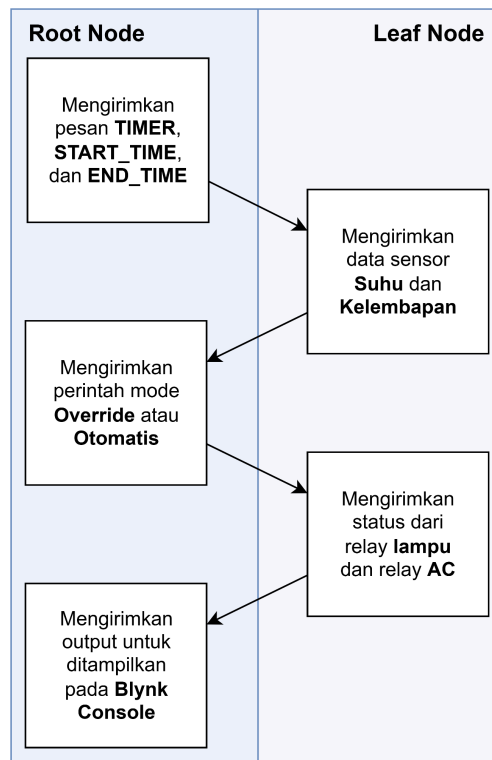


Diagram di atas menggambarkan *control logic* yang digunakan untuk mengelola kedua *device* (lampu dan AC) dalam dua *operation mode*, yaitu **auto mode** dan **override mode**. Proses ini dimulai dengan tahap pemeriksaan, apakah sistem sudah berada dalam **auto mode** atau belum? Jika sudah, maka, sensor RCWL-0516 akan mendeteksi pergerakan yang terjadi. Bilamana tidak terjadi pergerakan, maka sistem akan memulai *timer* yang akan memberikan limitasi seberapa lama lampu dan AC akan menyala. Namun, bilamana terjadi pergerakan yang berhasil untuk dideteksi, maka akan dilakukan pengecekan jumlah gerakan yang terjadi. Apabila lebih besar sama dengan 5, maka hal tersebut menunjukkan bahwa ruangan sedang aktif dan dengan demikian baik lampu maupun AC akan menyala. Sedangkan, bila kurang dari 5, maka hal tersebut menunjukkan bahwa tidak terlalu banyak aktivitas yang terjadi di dalam ruangan (minim), sehingga hanya lampu yang akan menyala untuk menghemat penggunaan energi.

Lebih lanjut, jika sistem berada dalam **override mode**, maka kontrol terhadap kedua *device* (lampu dan AC) akan dilakukan secara manual melalui *Blynk Console*. Pada mode ini, pemegang *Blynk Console* memiliki kendali yang penuh atas menyala/matinya lampu dan AC, tanpa dipengaruhi oleh faktor *motion detection*, maupun *timer logic*. Siklus ini akan terus berulang untuk memastikan bahwa *electronic devices* yang terdapat di dalam kelas dapat digunakan secara efisien serta menghemat penggunaan energi.

### 2.2.2 Root Node



Komunikasi antara root-node dan leaf-node berjalan secara terstruktur dengan pertukaran data yang konstan dan informatif. Root-node memulai komunikasi dengan mengirimkan pesan konfigurasi seperti *TIMER*, *START\_TIME*, dan *END\_TIME* ke *leaf-node*. Pesan ini berfungsi untuk mengatur durasi kerja perangkat dan jadwal aktif sesuai kebutuhan. Setelah menerima pesan tersebut, leaf-node mulai membaca data sensor suhu dan kelembapan, kemudian mengirimkan hasilnya kembali ke *root-node*.

Setelah data diterima, *root-node* memprosesnya dan melanjutkan dengan mengirimkan perintah kepada *leaf-node*. Perintah ini dapat berupa *override* mode yang memungkinkan kontrol manual perangkat atau mode otomatis untuk melakukan pembacaan sensor. *Leaf-node* merespons dengan mengirimkan status perangkat, seperti keadaan relay lampu dan relay AC (ON/OFF), kembali ke *root-node*.

Terakhir, *root-node* menggunakan data ini untuk menampilkan hasilnya di *Blynk Console*, memberikan visualisasi data sensor dan status perangkat kepada pengguna. Proses ini memastikan komunikasi dua arah yang efisien antara node, dan memberikan kontrol penuh dalam monitoring.

## 2.3 HARDWARE AND SOFTWARE INTEGRATION

Dalam project *Seize the Class: Smart Classroom Occupancy Control System*, kami melakukan integrasi antara *hardware* dengan *software* sebagai berikut:

### a) Mendeteksi Lingkungan

Pada bagian/fitur ini, kami menggunakan *hardware* berupa sensor DHT11 yang dimanfaatkan untuk mendeteksi suhu dan kelembapan. Selain itu, kami juga menggunakan RCWL-0516 *motion* sensor untuk mendeteksi gerakan-gerakan yang terjadi di dalam kelas. Adapun sensor DHT11 terhubung ke pin GPIO 15 pada ESP32 untuk mengirimkan data suhu dan kelembapan, sedangkan RCWL-0516 akan memberikan sinyal digital (HIGH/1) tepat ketika terdapat gerakan yang terdeteksi. Lebih lanjut, dari sisi *software*, data yang diterima dari sensor akan diolah oleh ESP32 dengan menggunakan kode berbasiskan kepada FreeRTOS. Di dalam implementasi kode, setiap sensor akan memiliki *task* yang terpisah, baik untuk membaca maupun memproses data secara parallel, i.e. *task* RCWL-0516 untuk memantau sinyal *motion* secara *real-time*, *task* pembacaan data suhu dan kelembapan, etc.

### b) Control Output Devices

Pada bagian ini, data yang diproses oleh ESP32 akan digunakan untuk mengontrol *output devices*, seperti lampu dan AC *via relay module*. Adapun *relay* 1 akan terhubung ke GPIO 17 untuk *menghandle* lampu, sedangkan *relay* 2 akan terhubung ke GPIO 18 untuk *menghandle* AC. Di sisi lain, LED akan terhubung ke GPIO 4 dan berperan sebagai indikator secara visual, yang merepresentasikan terjadinya pergerakan di dalam kelas. Dari sisi *software*, program di bentuk sedemikian rupa untuk menentukan kapan *relay* akan diaktifkan atau dinonaktifkan, dan hal ini dilakukan berdasarkan data yang diperoleh dari sensor. Selain itu, proses aktivasi dan deaktivasi *relay* dilakukan dengan menggunakan `digitalWrite()` *function* yang akan mengontrol status dari pin GPIO.

#### c) Komunikasi antar *Devices*

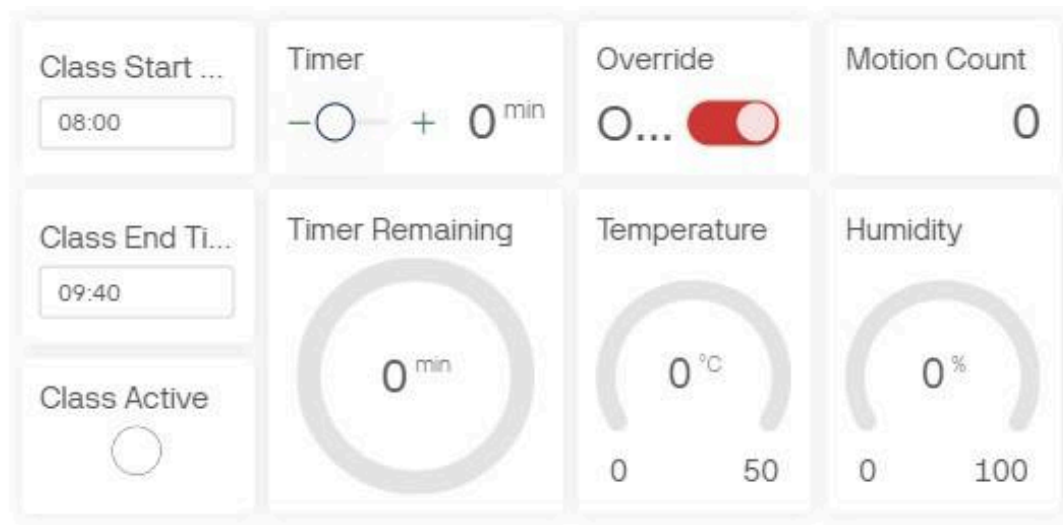
Dalam percobaan ini, kami menggunakan dua buah ESP32 agar dapat saling berkomunikasi dengan menggunakan *mesh network protocol*, di mana *leaf node* akan mengumpulkan data yang diperoleh/berasal dari sensor, sedangkan *root node* akan menerima data tersebut. Dengan diterapkannya *mesh network protocol*, maka memungkinkan bagi kedua buah ESP32 untuk saling berkomunikasi, tanpa perlu bergantung kepada penggunaan koneksi internet secara langsung. Dari sisi *software*, data yang sudah terkumpul oleh *leaf node* akan segera dikirimkan menuju ke *root node* melalui *mesh function*, yaitu `sendBroadcast()`. Setelah itu, *root node* akan memproses data tersebut, kemudian meneruskannya menuju ke MQTT server untuk melakukan pemantauan status/keadaan kelas secara *remote*. Adapun penggunaan `painlessMesh` library bertujuan untuk memastikan bahwa sinkronisasi yang terjadi antar perangkat *via* jaringan *mesh* dapat berlangsung dengan baik.

#### d) *Operation Mode* dan Koneksi ke Server

Pada sistem ini, kami membentuk dua *mode* operasi, yaitu manual dan otomatis. Kami juga melakukan pengaturan yang memungkinkan dilakukannya *controlling device* secara *remote via* server. Pada mode otomatis, maka hardware yang digunakan (sensor RCWL-0516 dan relay) akan bekerja untuk mendeteksi gerakan serta mengaktifasi baik lampu maupun AC yang terdapat di dalam kelas, *based on* kondisi lingkungan pada saat itu. Sedangkan, *control logic* untuk mode otomatis akan dihandle oleh *software* pada ESP32, dengan penggunaan *function* `taskControlAuto()`, `relayAutoCallback()`, `taskMotion()`, `isClassActive()`, dan `task DHT()`.

Di sisi lain, untuk *mode* manual, *user* dapat mengontrol *device* secara langsung melalui *command* yang dikirimkan melalui Blynk Console. Untuk *root node* yang terhubung ke Wi-Fi, ia akan berperan sebagai *connector* utama *in between mesh network* dengan MQTT. Dari sisi *software*, kode menggunakan *function* `mesh.onReceive()`, `handleReceivedMessage()`, `mesh.sendBroadcast()`, `Blynk.virtualWrite()`, dan `onNewConnection()` untuk menerima data dari *leaf node via mesh network*, dan kemudian mengirim data tersebut ke server.

## 2.4 USER INTERFACE DESIGN



Sistem *Seize the Class: Smart Classroom Occupancy Control System* menggunakan *Blynk Console* sebagai antarmuka pengguna untuk memantau dan mengendalikan perangkat dalam ruang kelas secara *real-time*. Pada *Leaf Node*, data sensor suhu dan kelembapan dari DHT11 serta status perangkat (lampu dan AC) dikirimkan ke *Root Node* menggunakan *mesh network* berbasis *painlessMesh*. Data ini kemudian diteruskan ke *platform* Blynk melalui koneksi Wi-Fi yang dimiliki *Root Node*. Dengan ini, pengguna dapat memantau informasi seperti suhu ruangan, kelembapan udara, dan status perangkat secara langsung pada *Blynk Console*.

Antarmuka di *Blynk Console* menyediakan dua mode operasi utama, ***Auto Mode*** dan ***Override Mode***, yang dapat diaktifkan melalui tombol yang ditampilkan di aplikasi. Dalam *Auto Mode*, kontrol lampu dan AC dilakukan secara otomatis berdasarkan input dari sensor RCWL-0516 pada *LeafNode*, yang mendeteksi gerakan di dalam ruangan. Status ruangan akan diperbarui sebagai "*Occupied*" jika sensor mendeteksi aktivitas, sementara timer akan mematikan perangkat jika tidak ada gerakan baru dalam waktu tertentu.

Sementara itu, dalam *Override Mode*, kontrol perangkat sepenuhnya dilakukan secara manual oleh pengguna melalui tombol khusus di antarmuka. Pada mode ini, pengguna dapat menyalakan atau mematikan lampu dan AC tanpa dipengaruhi oleh data sensor. Status perangkat ditampilkan dalam bentuk indikator on/off pada *Blynk Console*, memberikan transparansi penuh atas kondisi perangkat di ruangan.

Data suhu dan kelembapan dari sensor DHT11 juga ditampilkan dalam bentuk *widget* grafik untuk memberikan pengguna wawasan *real-time* terhadap kondisi lingkungan ruangan. Proses pengiriman data dari *Leaf Node* ke *Root Node* hingga ke Blynk *Console* dirancang dengan protokol yang andal untuk memastikan informasi yang ditampilkan akurat dan *up-to-date*. Desain antarmuka ini mendukung pengelolaan perangkat yang efisien sekaligus memberikan fleksibilitas tinggi kepada pengguna.



## CHAPTER 3

### TESTING AND EVALUATION

#### 3.1 TESTING

Pada proyek ini, kami melakukan serangkaian pengujian untuk memastikan bahwa sistem dapat bekerja sesuai dengan desain yang sudah dirancang, serta dapat memberikan output yang sesuai dengan harapan berdasarkan pengujian input dari user ataupun sensor. Berikut ini merupakan beberapa pengujian yang dilakukan:

- Test Monitoring, di mana terjadi verifikasi terhadap pembacaan data suhu dan kelembapan dengan menggunakan sensor DHT11, yang akan ditunjukkan pada Blynk *interface*.
- Test Motion Detection, di mana akan ada pengujian proses deteksi gerakan, dengan menggunakan sensor RCWL0516 agar dapat menyalakan/dan mematikan baik lampu maupun AC dengan menggunakan mode auto.
- Test Override Control, di mana akan ada evaluasi terhadap control secara manual melalui interface Blynk, agar tidak terpengaruh oleh proses deteksi motion.
- Test Mesh Network, di mana akan ada pengujian terhadap stabilitas komunikasi yang terjadi antar node di dalam jaringan mesh, dengan tujuan untuk memastikan bahwa data dapat dikirimkan dengan benar in between root dan leaf node.
- Test UI, yaitu menguji bahwa interface Blynk dapat memantau perubahan status atau data yang terjadi secara real-time.

Proyek ini dilakukan dengan menggunakan Arduino IDE, di mana kami akan menguji bahwa sistem dapat bekerja dengan optimal selama hardware dan software terintegrasi dengan maksimal.

## 3.2 RESULT

```
Broadcasting relay status: RELAY_STATUS:LAMP_OFF,AC_OFF  
Temperature: 26.70°C, Humidity: 38.00%  
Broadcasting DHT data: Temperature: 26.70°C, Humidity: 38.00%  
Broadcasting status: RELAY_STATUS:OFF  
Broadcasting relay status: RELAY_STATUS:LAMP_OFF,AC_OFF  
Temperature: 26.70°C, Humidity: 38.00%  
Broadcasting DHT data: Temperature: 26.70°C, Humidity: 38.00%  
Broadcasting status: RELAY_STATUS:OFF
```

### *Leaf Node*

```
Sent class end time to Node ID: 47852977  
Message received: RELAY_STATUS:LAMP_OFF,AC_OFF  
Lamp Status: OFF | AC Status: OFF  
Lamp Status: 0 | AC Status: 0 | Class Active Status: 0  
Message received: Temperature: 26.70°C, Humidity: 38.00%  
Received DHT data: Temperature: 26.70°C, Humidity: 38.00%  
Temperature: 26.70°C | Humidity: 38.00%  
[277610]Connecting to blynk.cloud:80
```

### *Root Node*

Pengujian dilakukan pada berbagai kondisi ruang dengan tujuan untuk memastikan ketepatan dalam mendeteksi kehadiran dan mengontrol perangkat. Hasil dari pengujian ini menunjukkan bahwa sistem kami efektif dalam mengontrol perangkat dan memonitor kondisi lingkungan secara *real-time*, sehingga memberikan solusi yang efisien dalam pengelolaan energi di ruang kelas.

Pada **mode *Auto***, ketika sensor mendeteksi gerakan pertama, lampu akan menyala. Setelah dua atau lebih gerakan terdeteksi, sistem akan menyalakan AC secara otomatis. Proses ini menunjukkan bahwa mode Auto dapat berfungsi dengan baik dalam mengatur perangkat secara otomatis berdasarkan tingkat aktivitas di dalam ruangan.

Pada **mode *Override***, ketika diaktifkan melalui antarmuka Blynk, perangkat akan tetap berada pada status yang dipilih, terlepas dari adanya gerakan. Hal ini menunjukkan bahwa sistem memberikan fleksibilitas dalam kontrol manual.

Monitoring data suhu dan kelembaban yang diperoleh dari sensor DHT11 secara akurat ditampilkan di dashboard Blynk dan memperbarui data secara *real-time*. Pengujian *mesh network* dan komunikasi antar *Leaf Node* dan *Root Node* menggunakan jaringan *mesh* berjalan lancar, meskipun terkadang terjadi gangguan sinyal yang mempengaruhi keterlambatan pengiriman data.

### 3.3 EVALUATION

Selama pengujian, beberapa tantangan dan kesalahan teknis terdeteksi yang mempengaruhi kinerja sistem:

#### 1. Stabilitas Koneksi *Mesh Network*

Pada beberapa titik, jaringan *mesh* antara *Leaf Node* dan *Root Node* mengalami gangguan, yang menyebabkan keterlambatan dalam pengiriman data. Hal ini mungkin disebabkan oleh interferensi sinyal atau masalah dengan konfigurasi jaringan. Kami akan melakukan evaluasi lebih lanjut untuk meningkatkan kestabilan jaringan.

#### 2. Koneksi Blynk

Beberapa kali terjadi delay dalam sinkronisasi antara Blynk dan perangkat, yang mempengaruhi kecepatan pengambilan data dan kontrol perangkat. Hal ini kemungkinan disebabkan oleh kecepatan internet yang tidak stabil atau masalah dalam integrasi Blynk dengan Wi-Fi.

#### 3. Pengaturan Mode dan Prioritas

Beberapa pengujian menunjukkan bahwa mode *Auto* dan *Override* tidak selalu berfungsi dengan prioritas yang jelas saat beroperasi bersamaan. Hal ini perlu ditangani dengan memperbaiki logika pengendalian prioritas di dalam perangkat lunak.

Berdasarkan hasil evaluasi tersebut, berikut adalah beberapa langkah perbaikan yang akan dilakukan:

1. Perbaikan pada *Mesh Network*: Kami akan melakukan optimasi konfigurasi jaringan *mesh* untuk meningkatkan kestabilan dan mengurangi gangguan.

2. Peningkatan Kinerja Blynk: Kami akan memeriksa koneksi Wi-Fi dan memastikan kecepatan transfer data dari *Root Node* ke Blynk lebih cepat dan lebih stabil.
3. Penyempurnaan Mode *Auto* dan *Override*: Logika kontrol mode akan kami sesuaikan untuk memastikan perangkat berfungsi sesuai dengan pengaturan yang diinginkan tanpa gangguan.

## CHAPTER 4

### CONCLUSION

#### 4.1 SUMMARY

Proyek *Seize the Class: Smart Classroom Occupancy Control System* berhasil mengintegrasikan komunikasi dua arah antara *root-node* dan *leaf-node* dalam sistem berbasis IoT untuk mengelola perangkat listrik di ruang kelas secara efisien. Sistem ini mampu membaca data suhu dan kelembapan melalui sensor DHT11, mendeteksi pergerakan di ruangan menggunakan *motion sensor*, serta menyediakan dua mode operasi, yaitu *Auto Mode* untuk kontrol otomatis berdasarkan aktivitas di ruangan dan *Override Mode* untuk kontrol manual. Data hasil monitoring ditampilkan secara real-time di *Blynk Console*, sehingga pengguna dapat memantau kondisi perangkat secara langsung.

Meski sistem telah berjalan dengan baik, beberapa tantangan tetap ada, seperti sinkronisasi dan stabilitas komunikasi antara *leaf-node* dan *root-node*. Namun, perangkat berhasil menjalankan fungsi utama, termasuk pengiriman data sensor, penerimaan perintah kontrol, dan pemantauan status perangkat seperti lampu dan AC. Dengan penggunaan mesh networking berbasis ESP32, sistem ini dapat bekerja di jaringan lokal tanpa memerlukan infrastruktur jaringan eksternal yang kompleks.

#### 4.2 FUTURE WORKS

Kedepannya, pengembangan dapat difokuskan pada penyempurnaan protokol pengiriman data antara *leaf-node* dan *root-node* untuk meningkatkan stabilitas dan keandalan komunikasi. Selain itu, fitur notifikasi berbasis MQTT akan ditambahkan untuk memberikan informasi langsung kepada pengguna melalui *platform* berbasis *cloud*. Fitur ini dapat digunakan untuk mengirimkan pemberitahuan terkait perubahan status perangkat atau peringatan tertentu.

Pengembangan lebih lanjut juga akan mencakup peningkatan durabilitas perangkat untuk memastikan sistem dapat digunakan dalam berbagai kondisi lingkungan. Rencana pengembangan lainnya dapat dilakukan dengan mengeksplorasi integrasi teknologi lain, seperti analisis data berbasis AI untuk memberikan wawasan

tambahan kepada pengguna, serta memperluas aplikasi sistem ini tidak hanya untuk pendidikan, tetapi juga untuk manajemen energi di lingkungan perkantoran dan rumah pintar. Sehingga, sistem ini dapat menjadi solusi IoT yang lebih tangguh, relevan, dan bermanfaat bagi berbagai kebutuhan pengguna.

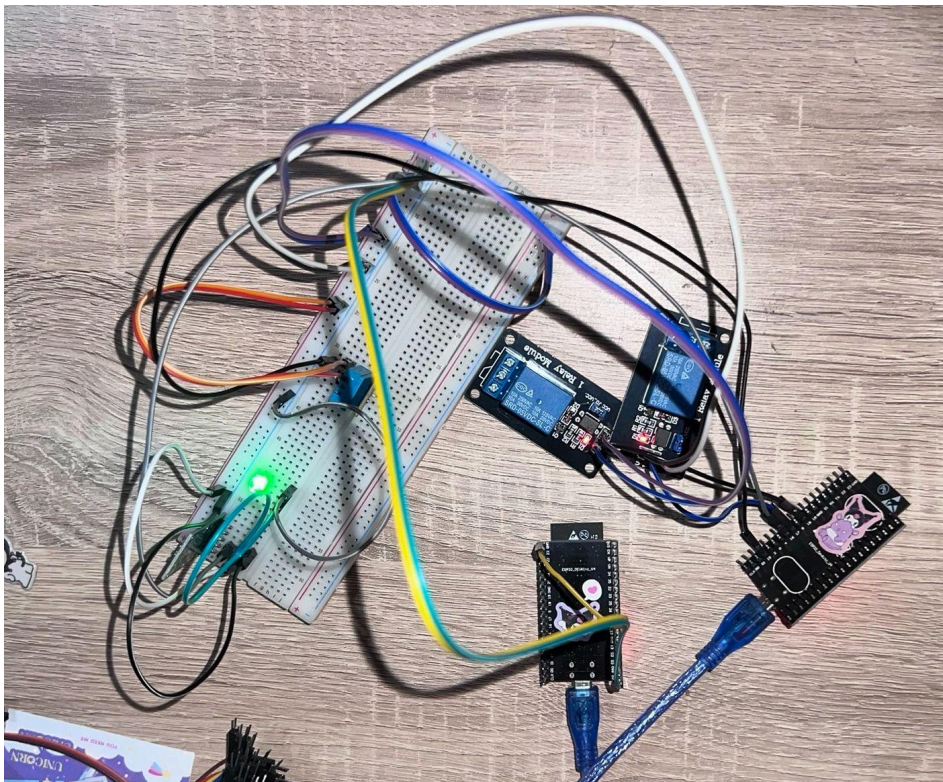
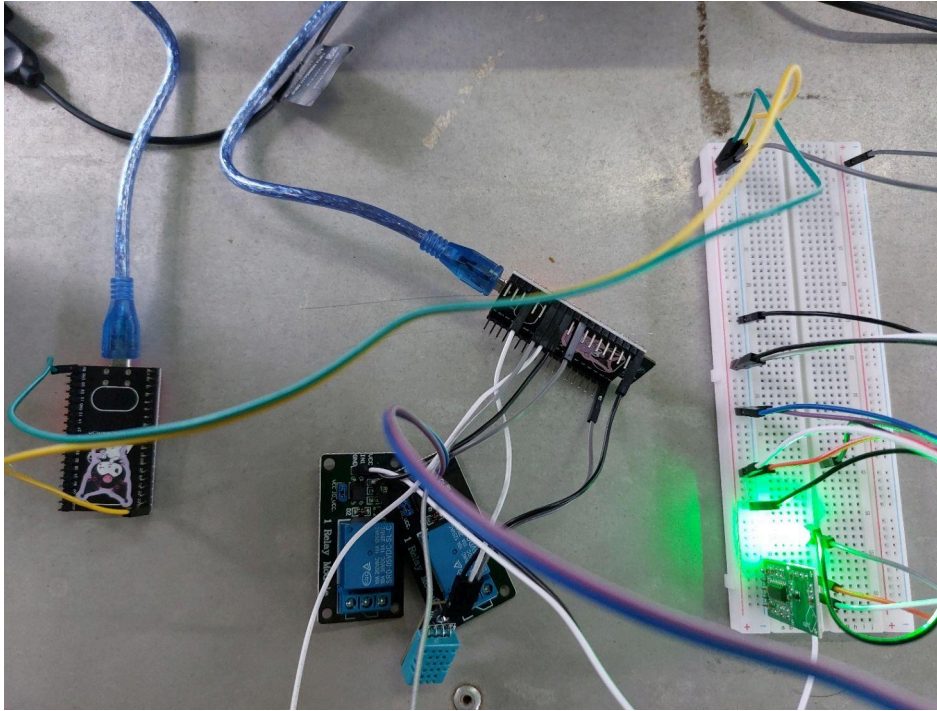
## REFERENCES

- [1] Espressif, “Get Started - ESP32 - ESP-IDF Programming Guide latest documentation,” [espressif.com](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/). Available: <https://docs.espressif.com/projects/esp-idf/en/latest/esp32/get-started/>. [Accessed: Dec. 10, 2024].
- [2] Blynk, “Introduction | Blynk Documentation,” Blynk.io, Mar. 07, 2024. Available: <https://docs.blynk.io/en>. [Accessed: Dec. 10, 2024].
- [3] FreeRTOS, “RTOS Fundamentals - FreeRTOS,” Freertos.org, 2024. Available: <https://freertos.org/Documentation/01-FreeRTOS-quick-start/01-Beginners-guide/01-RTOS-fundamentals>. [Accessed: Dec. 10, 2024].
- [4] Coopdis, Scotty Franzysen, Edwin van Leeuwen, Germán Martín, Maximilian Schwarz, Doanh Doanh, “painlessMesh - Intro to painlessMesh,” GitLab. Available: <https://gitlab.com/painlessMesh/painlessMesh>. [Accessed: Dec. 10, 2024].
- [5] “DHT11 Sensor Pinout, Features, Equivalents & Datasheet,” Components101, Jul. 16, 2021. Available: <https://components101.com/sensors/dht11-temperature-sensor>. [Accessed: Dec. 10, 2024].
- [6] “RCWL-0516,” [automotionsensor.com](https://automotionsensor.com/wp-content/uploads/2020/12/rcwl-0516-data-sheet.pdf). Available: <https://automotionsensor.com/wp-content/uploads/2020/12/rcwl-0516-data-sheet.pdf>. [Accessed: Dec. 10, 2024].
- [7] “RELAY MODULES,” [components101.com](https://components101.com/sites/default/files/component_datasheet/Relay-Module-Datasheet.pdf). Available: [https://components101.com/sites/default/files/component\\_datasheet/Relay-Module-Datasheet.pdf](https://components101.com/sites/default/files/component_datasheet/Relay-Module-Datasheet.pdf). [Accessed: Dec. 10, 2024].



## APPENDICES

### Appendix A: Documentation





## **Appendix B: Source Code**

[https://github.com/arifatalya/Group7-IoT\\_SeizeTheClass](https://github.com/arifatalya/Group7-IoT_SeizeTheClass)