



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

**DanceRTOS - FreeRTOS ESP32 Attendance System**

**GROUP A5**

<b>Akmal Rabbani</b>	<b>2106731610</b>
<b>Rian Abrar</b>	<b>2106708242</b>
<b>Satya Ananda Sulistio</b>	<b>2106705524</b>
<b>Seno Pamungkas</b>	<b>2106731586</b>

## PREFACE

Dalam era teknologi yang terus berkembang dengan pesat, penerapan Internet of Things (IoT) telah memberikan dampak yang signifikan dalam berbagai bidang, termasuk dalam bidang pendidikan. Laporan ini merangkum hasil dari proyek pengembangan sistem absensi menggunakan ESP32 dan teknologi Radio-Frequency Identification (RFID) yang bertujuan untuk meningkatkan efisiensi serta akurasi proses absensi di lingkungan pendidikan, khususnya pada lingkungan kampus.

Proyek ini dibentuk dengan tujuan utama untuk membawa inovasi dalam pengelolaan kehadiran, memperkenalkan sistem yang tidak hanya efektif, tetapi juga praktis dalam mendukung manajemen kehadiran mahasiswa. Proyek ini didesain dengan menggunakan perangkat keras ESP32 sebagai otak utamanya, dan juga memanfaatkan teknologi RFID yang memungkinkan pengguna dalam melakukan absensi dengan cepat dengan menggunakan kartu identifikasinya.

Laporan ini mencakup penjelasan mendalam tentang komponen utama yang digunakan dalam proyek, termasuk peran ESP32 sebagai pengelola utama sistem, penggunaan RFID Reader dan Kartu untuk identifikasi, dan penggunaan fitur LCD untuk mengkonfirmasi absensi berhasil. Selain itu, proyek ini memanfaatkan Interface Web/Aplikasi untuk pengelolaan data kehadiran

Dengan adanya proyek ini, diharapkan dapat memberikan gambaran yang jelas terhadap konsep, implementasi, serta manfaat yang dapat diperoleh dari sistem absensi menggunakan ESP32 dan RFID dalam meningkatkan efisiensi serta akurasi dalam proses absensi.

Depok, December 5, 2023

Group A5

## 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.....	4
1.3    ACCEPTANCE CRITERIA.....	5
1.4    ROLES AND RESPONSIBILITIES.....	6
1.5    TIMELINE AND MILESTONES.....	7
<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
2.3    HARDWARE AND SOFTWARE INTEGRATION.....	14
<b>CHAPTER 3.....</b>	<b>16</b>
<b>TESTING AND EVALUATION.....</b>	<b>16</b>
3.1    TESTING.....	16
3.2    RESULT.....	24
3.3    EVALUATION.....	32
<b>CHAPTER 4.....</b>	<b>33</b>
<b>CONCLUSION.....</b>	<b>33</b>
<b>REFERENCES.....</b>	<b>34</b>
<b>APPENDICES.....</b>	<b>35</b>

## **CHAPTER 1**

### **INTRODUCTION**

#### **1.1 PROBLEM STATEMENT**

Di tengah dinamika pendidikan yang terus berkembang, proses pengelolaan kehadiran atau absensi seringkali menjadi tantangan yang membutuhkan solusi efektif. Sistem manual yang digunakan untuk melakukan absensi seringkali rentan terhadap kesalahan dan kurangnya efisiensi dalam pencatatan kehadiran mahasiswa atau karyawan. Hal ini juga dibuktikan dengan banyaknya mahasiswa yang melakukan “titip absen”, yang membuat sistem kehadirannya menjadi tidak akurat.

Dalam konteks ini, penelitian ini dibentuk sebagai respons terhadap permasalahan yang umum terjadi dalam proses absensi di lingkungan akademik maupun perusahaan. Proses manual yang masih dominan sering kali menyebabkan ketidakakuratan data kehadiran, kesulitan dalam manajemen data, serta memakan waktu yang cukup besar baik bagi pengelola maupun pengguna sistem.

Dari keterbatasan yang terdapat dalam sistem manual, dapat mengakibatkan kurangnya transparansi dan keterkinian informasi kehadiran, yang pada gilirannya dapat menghambat pengambilan keputusan yang tepat, terutama bagi pihak pengelola, seperti dosen atau manajer, dalam menilai tingkat kehadiran dan keterlibatan.

Oleh karena itu, kita membangun sistem kehadiran ini dengan tujuan untuk mengatasi berbagai masalah yang dihadapi dalam proses absensi, dengan menerapkan teknologi IoT dan RFID. Diharapkan solusi ini dapat memberikan efisiensi, akurasi, dan kemudahan dalam pengelolaan kehadiran bagi mahasiswa.

#### **1.2 PROPOSED SOLUTION**

Untuk mengatasi berbagai permasalahan yang teridentifikasi sebelumnya, solusi yang diusulkan dalam proyek ini mengadopsi pendekatan teknologi yang efisien dan inovatif. Melalui penerapan Internet of Things (IoT) dengan perangkat ESP32 sebagai otak utama sistem, serta teknologi Radio-Frequency Identification (RFID), hal ini ditujukan untuk memberikan solusi terpadu dalam proses absensi yang lebih akurat, efisien, dan user-friendly.

Komponen utama yang digunakan, seperti ESP32 sebagai pusat pengendalian, RFID Reader dan Kartu untuk identifikasi pengguna, serta penggunaan *Interface Web/Aplikasi* untuk manajemen data kehadiran, dirancang secara sistematis untuk memastikan interaksi yang lancar antara perangkat keras dan pengguna.

Solusi ini juga menghadirkan fitur-fitur yang mendukung penggunaan yang mudah, seperti layar LCD untuk konfirmasi langsung untuk mengindikasikan keberhasilan absensi. Adapun penggunaan *Interface Web/Aplikasi* memberikan kemudahan akses dan manajemen data kehadiran bagi pihak terkait, seperti dosen atau manajer. Selain itu, terdapat pengiriman kode OTP, yang ditujukan untuk memastikan keakuratan kehadiran mahasiswa, agar dapat mencegah terjadinya “titip absen” terhadap sistem kehadiran ini.

Proses registrasi pengguna melalui tag RFID dan *database* merupakan salah satu elemen penting dalam solusi ini. Melalui langkah ini, pengguna akan terdaftar dengan lengkap dalam sistem sebelum dapat melakukan absensi. Dengan demikian, proses absensi selanjutnya akan lebih terstruktur dan terjamin keakuratannya.

Diharapkan bahwa solusi yang diusulkan ini tidak hanya akan mengatasi permasalahan yang ada, tetapi juga membawa manfaat signifikan dalam meningkatkan efisiensi, akurasi, serta keterkinian informasi kehadiran bagi para pemangku kepentingan.

### 1.3 ACCEPTANCE CRITERIA

Berikut ini adalah *acceptance criteria* dari proyek ini:

1. Sistem harus mampu melakukan registrasi pengguna dengan memungkinkan identifikasi tag RFID dan menyimpan data yang sesuai (nama dan NPM) ke dalam database secara akurat.
2. Sistem harus dapat melakukan proses absensi dengan memvalidasi tag pada kartu RFID terhadap database pengguna yang telah terdaftar, memberikan konfirmasi

kehadiran dengan menampilkan informasi waktu masuk/waktu keluar pada layar LCD.

3. Untuk memberikan verifikasi kebenaran atas kehadirannya, sistem harus dapat mengirimkan kode OTP kepada penggunanya.
4. Interface Web/Aplikasi harus dapat diakses dengan respons yang cepat dan menampilkan data kehadiran secara terperinci bagi dosen atau pengelola sistem. Hal ini harus mencakup kemampuan untuk menambah, mengubah, dan menghapus data kehadiran.
5. Sistem harus dapat mendukung operasional sehari-hari tanpa mengalami gangguan berarti, seperti kegagalan pembacaan tag RFID atau kesalahan dalam penanganan data kehadiran.
6. Keamanan sistem harus terjamin dengan baik, termasuk perlindungan terhadap akses yang tidak sah terhadap data kehadiran dan fitur otentikasi yang memadai dalam Interface Web/Aplikasi.

#### 1.4 ROLES AND RESPONSIBILITIES

The roles and responsibilities assigned to the group members are as follows:

Roles	Responsibilities	Person
Rangkaian	Bertanggung jawab untuk menyusun rangkaian dan melakukan penyolderan.	Satya Ananda Sulistio, Seno Pamungkas, Akmal Rabbani, Rian Abrar
Pembuatan kode Hardware	Bertanggung jawab untuk membuat kode untuk implementasi ESP32, RFID, LCD, serta koneksi ke server <i>database</i> .	Satya Ananda Sulistio, Akmal Rabbani
Pembuatan kode <i>Frontend</i>	Bertanggung jawab untuk membuat kode untuk <i>interface</i> aplikasi.	Seno Pamungkas

Pembuatan kode <i>Backend</i>	Bertanggung jawab untuk membuat kode <i>backend</i> aplikasi.	Seno Pamungkas
Pembuatan Blynk	Bertanggung jawab untuk membuat <i>interface</i> Blynk untuk memungkinkan pengontrolan.	Satya Ananda Sulistio
Pembuatan Laporan Proyek Akhir	Bertanggung jawab untuk membuat laporan akhir	Akmal Rabbani, Rian Abrar
Presentasi	Bertanggung jawab untuk membuat presentasi Proyek Akhir.	Satya Ananda Sulistio, Seno Pamungkas, Akmal Rabbani, Rian Abrar

Table 1. Roles and Responsibilities

## 1.5 TIMELINE AND MILESTONES

Month	November										December									
	21	22	23	24	25	26	27	28	29	30	1	2	3	4	5	6	7	8	9	10
Progress																				
Hardware Design Completion																				
Software Development																				
Integration and Testing of Hardware and Software																				
Final Product Assembly and Testing																				

Table 2.Gantt Chart

- Hardware Design Completion:

21-24 November: Mengembangkan ide terkait produk yang akan dibuat dan menyampaikannya kepada asisten lab yang bersangkutan.

25-26 November: membuat design dari produk

- Software Development:

28 November - 8 Desember: Merancang web application dan ESP32 yang akan digunakan.

- Integration and Testing of Hardware and Software:

4-8 Desember: Merangkai perangkat keras seperti ESP32, RFID, dan LED serta mengintegrasikannya dengan blynk dan web application yang telah dibuat.

- Final Product Assembly and Testing:

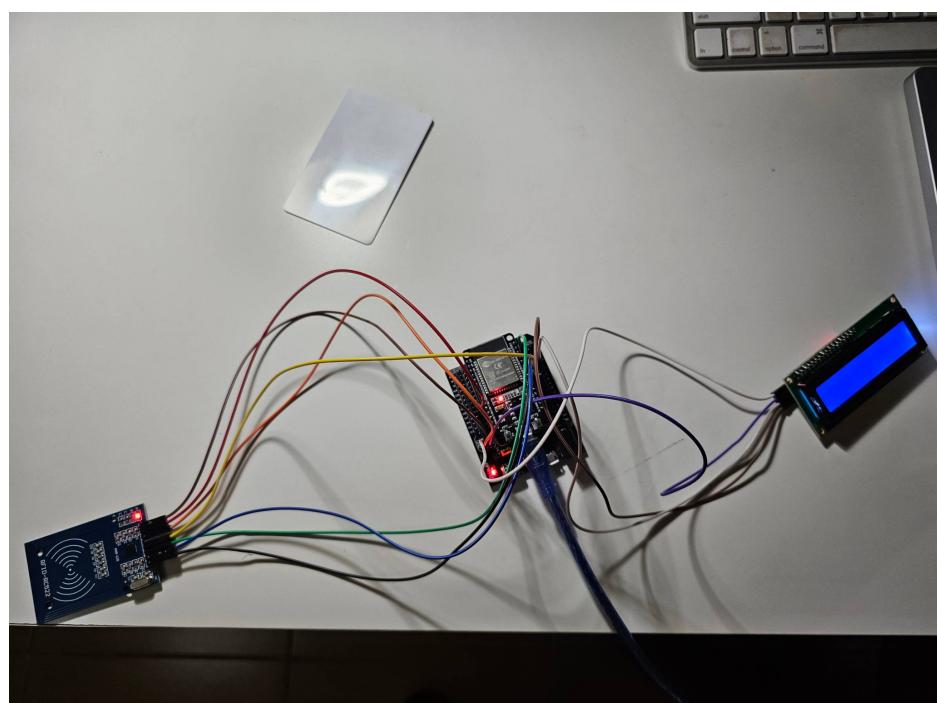
7-10 Desember: Menguji coba produk yang telah selesai dan membuat laporan.

## CHAPTER 2

### IMPLEMENTATION

#### 2.1 HARDWARE DESIGN AND SCHEMATIC

Perangkat keras di desain dengan menggunakan ESP32 sebagai otak utama dari proyek ini yang dirangkai dengan komponen-komponen pendukungnya. ESP32 ini digunakan karena kehandalannya dalam mengelola koneksi Wi-Fi, serta kapasitas yang cukup untuk memproses data dan mengendalikan modul lain dalam sistem. Selain itu, kami juga menggunakan RFID, yang fungsinya adalah untuk membaca informasi yang terdapat pada kartu RFID pengguna. Hal ini memungkinkan proses identifikasi yang cepat dan akurat. Selanjutnya, terdapat LCD yang berguna untuk menampilkan informasi kehadiran, seperti nama pengguna dan waktu masuk/keluar sebagai konfirmasi proses absensi.



Gambar 1. Rangkaian

Skematik sistem ini terdiri dari koneksi antara ESP32 dengan komponen pendukungnya. Skematik ini dibentuk secara terperinci untuk memastikan setiap koneksi dan pengaturan komponen sesuai dengan kebutuhan sistem, hal ini mencakup:

- Penyusunan rangkaian dengan menggunakan ESP32, RFID, LCD, dan komponen tambahan seperti kabel jumper.
- Koneksi fisik antara ESP32 dengan RFID Reader, dan LCD Display.
- Penyusunan *power supply* yang diperlukan untuk setiap komponen guna memastikan pasokan daya yang stabil dan sesuai.

## 2.2 SOFTWARE DEVELOPMENT

Pengembangan perangkat lunak dilakukan dengan fokus ke dalam implementasi untuk berbagai fitur yang diperlukan, seperti interaksi antar perangkat keras (ESP32 , RFID, dan LCD), hingga pengembangan perangkat lunak untuk web aplikasi yang dibuat.

```
#define BLYNK_TEMPLATE_ID "TMPL6rBMwxzs8"
#define BLYNK_TEMPLATE_NAME "AttendanceSystem"
#define BLYNK_AUTH_TOKEN "AJYj8CL2dzZCrxpF7EBER7ar7WnXXC1"
//*****BLYNK TOKENS*****
//*****libraries*****
//RFID-----
#include <SPI.h>
#include <MFRC522.h>
//ESP32-----
#include <WiFiManager.h>
#include <LiquidCrystal_I2C.h>
#include <PubSubClient.h>
#include <HTTPClient.h>
#include <BlynkSimpleEsp32.h>
#include <ArduinoJson.h>

//*****
#define SS_PIN 5 // GPIO5
#define RST_PIN 27 // GPIO27
//*****
MFRC522 RFC(SS_PIN, RST_PIN); // Create MFRC522 instance.
HTTPClient HTTP;
WiFiClient ESPClient;
PubSubClient ClientMQTT(ESPClient);

String getData, Link;
String OldCardID = "";
unsigned long previousMillis = 0;
static int lcdColumns = 18; // 18 columns LCD
static int lcdRows = 2; // 2 rows LCD
static LiquidCrystal_I2C LCD(0x27, lcdColumns, lcdRows); // LCD object
const char* ESP_AP_PASSWORD = "AP-ESP32"; // Customizable ESP AP password
const char* SSID = "ADJUST-SSID";
const char* PASSWORD = "ADJUST-PASSWORD";
const char* mqtt_server = "broker.hivemq.com";
const int mqtt_port = 1883;
const static String serverName = "https://absence-system.vercel.app/";
const static int interval = 15000;
```

Gambar 2. Potongan Kode untuk Perangkat Keras

Pada pengembangan *firmware* ESP32, kami menggunakan bahasa pemrograman C++ yang sesuai untuk diimplementasikan kedalam *platform* tersebut. Program ini bertanggung jawab atas logika kontrol sistem, seperti proses pembacaan data dari RFID, verifikasi kehadiran pada *database*, serta interaksi dengan layar LCD. Selain itu, program ini juga dapat mengelola koneksi ke server MQTT dan HTTP, yang memungkinkan untuk mengirim dan menerima data kehadiran atau informasi lainnya.

```
void mqtt_reconnect() {
    while (!ClientMQTT.connected()) {
        Serial.println("Attempting MQTT connection...");
        String connectionID = "ESP32" + String(random(0, 10000));
        if (ClientMQTT.connect(connectionID.c_str())) {
            Serial.println("connected");
            ClientMQTT.subscribe("esp32/otpEntered");
        } else {
            Serial.print("failed, rc=");
            Serial.print(ClientMQTT.state());
            Serial.println("Trying again in 5 seconds");
            delay(5000);
        }
    }
}

void vTaskMQTTConnection(void* params) {
    while (1) {
        if (!ClientMQTT.connected()) {
            mqtt_reconnect();
        }
        ClientMQTT.loop();
        vTaskDelay(50 / portTICK_PERIOD_MS);
    }
}

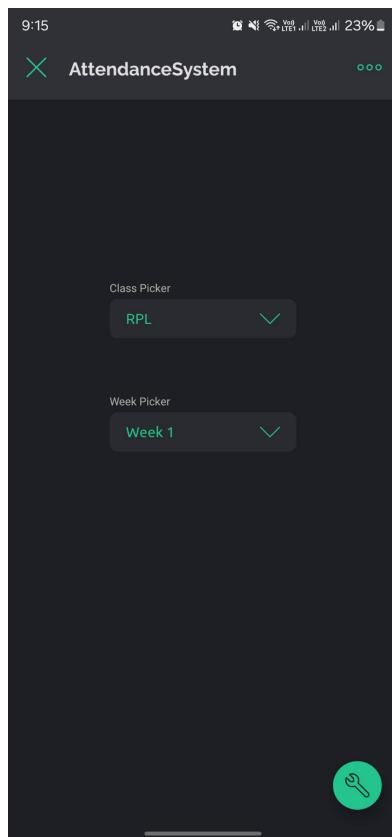
void vTaskWiFiConnection(void* params) {
    while (1) {
        unsigned long currentMillis = millis();
        // if WiFi is down, try reconnecting every 30 seconds
        if ((WiFi.status() != WL_CONNECTED) && (currentMillis - previousMillis >= interval)) {
            Serial.print(millis());
            Serial.println("Reconnecting to WiFi...");
            WiFi.disconnect();
            WiFi.reconnect();
            previousMillis = currentMillis;
        }
        vTaskDelay(100 / portTICK_PERIOD_MS);
    }
}
```

Gambar 3. Kode koneksi Server MQTT

Implementasi modul komunikasi dengan *database* dilakukan untuk menyimpan dan mengelola data kehadiran. Penggunaan protokol yang tepat seperti HTTP atau MQTT (Message Queuing Telemetry Transport) untuk mengirim dan menerima data ke dan dari server dilakukan secara aman dan efisien.

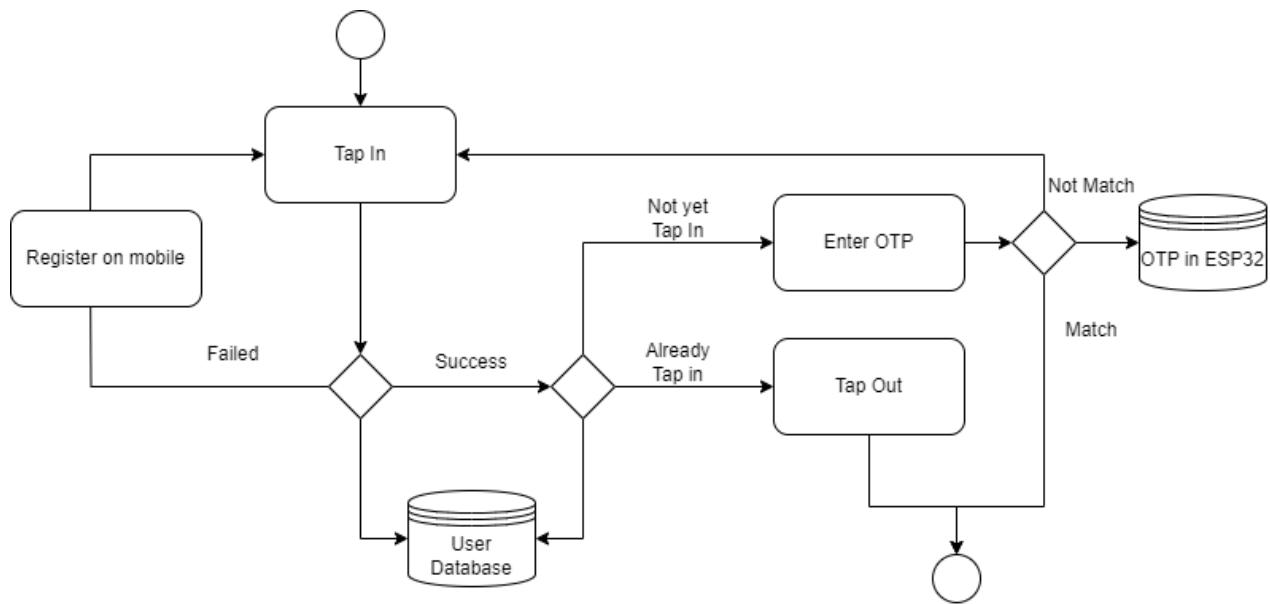
Untuk aplikasinya, dibentuk dengan *interface* yang responsif dan mudah digunakan oleh pengguna aplikasi, baik itu dosen yang mengakses data kehadiran maupun pengguna yang melakukan pendaftaran atau pengelolaan data pribadi. Selain itu, terdapat manajemen data kehadiran, yang mana terdapat fitur untuk menghapus data kehadiran dalam database melalui aplikasi web, yang dapat dilakukan oleh dosen.

Implementasi Blynk pada program ini dilakukan untuk mengatur kelas dan minggu pertemuan, sehingga pengaturan minggu pertemuan dan kelasnya dapat disesuaikan dengan baik. Berikut ini adalah tampilan *interface* dari implementasi aplikasi Blynk:



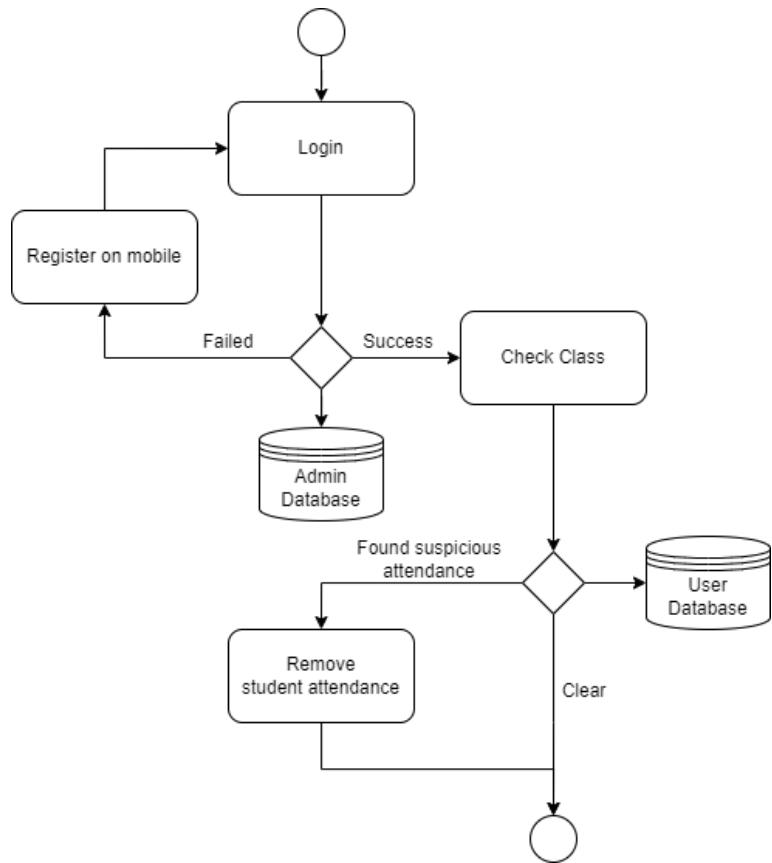
Gambar 5. Aplikasi Blynk

Pada saat mahasiswa melakukan tap in pada RFID, ESP32 akan menghubungkan dengan database apakah mahasiswa terdaftar pada kelas atau tidak. Jika belum terdaftar, maka mahasiswa harus melakukan registrasi pada web application. Jika sudah terdaftar, ESP32 akan memeriksa apakah mahasiswa sudah melakukan tap in sebelumnya. Jika belum melakukan tap in, maka tap akan dinyatakan sebagai tap in dan ESP32 akan menampilkan OTP pada LED yang sudah terhubung. Mahasiswa akan memasukan OTP pada web application dan absensi akan terekam pada log di web application. Jika sudah melakukan tap in, maka ESP32 akan menandai tap out dan akan menyimpan recordnya pada log di web application.



Gambar 6. Flowchart Mahasiswa

Dosen atau admin dapat mengakses database mahasiswa. Dosen atau admin akan login pada web application dan memeriksa kelas yang sudah ditempatkan. Apabila dosen atau admin belum memiliki account, maka dosen atau admin dapat melakukan registrasi terlebih dahulu. Dosen dan admin dapat memeriksa apabila terdapat mahasiswa yang melakukan absensi tetapi tidak menghadiri kelas. Apabila ditemukan hal seperti itu, Dosen atau admin dapat menghapus mahasiswa yang bersangkutan dari log absensi.



Gambar 7. Flowchart Dosen atau Admin

### 2.3 HARDWARE AND SOFTWARE INTEGRATION

Dalam integrasi *hardware* dan *software*, fokus utamanya adalah mengintegrasikan kode program yang telah dikembangkan, untuk diimplementasikan ke perangkat kerasnya (ESP32, RFID, dan komponen lainnya yang mendukung). Integrasi ini bertujuan untuk memastikan bahwa semua elemen bekerja bersama secara efektif dan sesuai dengan tujuan proyek.

Kode perangkat lunak dibuat untuk berinteraksi secara langsung dengan komponen perangkat keras seperti modul RFID, LCD, dan komponen lainnya. Hal ini dilakukan agar proses identifikasi pengguna, verifikasi, serta tampilan informasi kehadiran dapat berjalan secara terkoordinasi. Selain itu, program perangkat lunak ini memiliki tanggung jawab untuk

mengendalikan proses absensi, mulai dari pembacaan *tag* RFID hingga pengiriman data kehadiran ke server atau *interface* pengguna.

Setelah kode perangkat lunak diimplementasikan pada ESP32, dilakukan serangkaian uji coba untuk memastikan bahwa interaksi antara perangkat keras dan perangkat lunak berjalan sesuai yang diharapkan. Hal ini mencakup proses “*trial and error*”, sehingga kami dapat memodifikasi kode tersebut agar dapat memenuhi kriteria dari proyek ini.

## **CHAPTER 3**

### **TESTING AND EVALUATION**

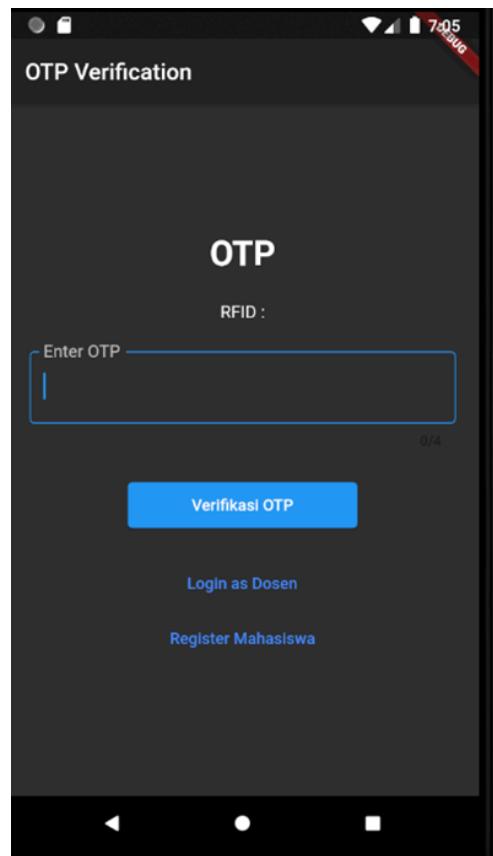
#### **3.1 TESTING**

Pada tahap pengujian sistem, kami melakukan serangkaian tes untuk memastikan bahwa program berjalan sesuai dengan kriteria yang sudah ditentukan sebelumnya. Sistem absensi yang dikembangkan ini harus berfungsi dengan baik dan sesuai dengan spesifikasi yang telah ditetapkan tersebut. Oleh karena itu, pengujian ini dilakukan dengan beberapa tahapan, seperti:

##### **3.1.1 Pengujian Fungsional**

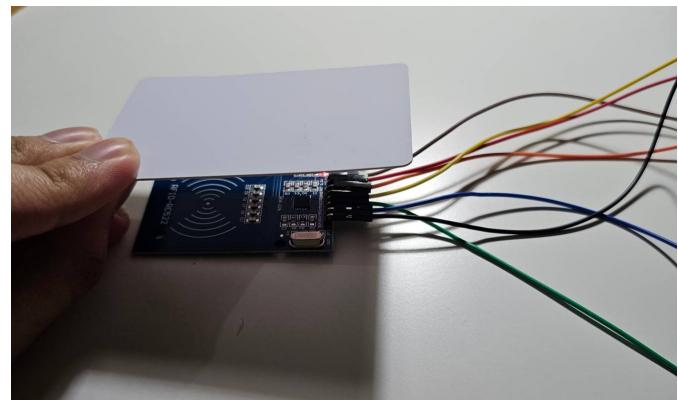
Pada tahapan ini, sistem diuji untuk mengidentifikasi pengguna berdasarkan *tag* RFID. Hal ini mencakup kelayakan pembacaan kartu dan pengenalan data yang diterima oleh ESP32 yang akan dievaluasi. Selanjutnya, dilakukan pengetesan terhadap data pengguna, untuk memverifikasi kebenaran data tersebut. Berikutnya, pengujian dilakukan untuk mengkonfirmasikan kehadiran yang tepat saat kartu RFID berhasil teridentifikasi. Ini termasuk kode OTP yang dikirimkan ke *interface* aplikasi untuk memastikan keakuratan kehadiran tersebut. Selain itu, pengujian tampilan data kehadiran pada LCD juga diuji, agar menampilkan data yang benar dan sesuai dengan mahasiswa yang bersangkutan. Berikut ini adalah tahapan pengujian dari fungsionalitas sistem absensi ini.

- Pengujian aplikasi untuk OTP



Gambar 4. *Interface* Aplikasi

- Pengujian “Tap in” kartu



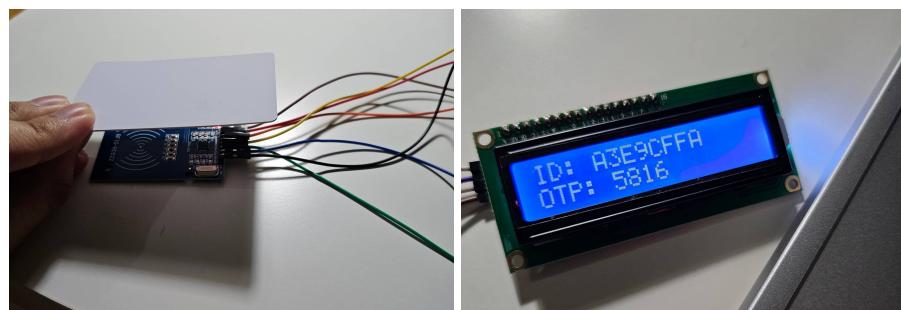
Gambar 5. “Tap in” Kartu

- Pengujian Tampilan LCD ketika melakukan Register dalam Aplikasi



Gambar 6. Tampilan LCD “Register in App”

- Pengujian “Tap in” kartu untuk menampilkan data mahasiswa



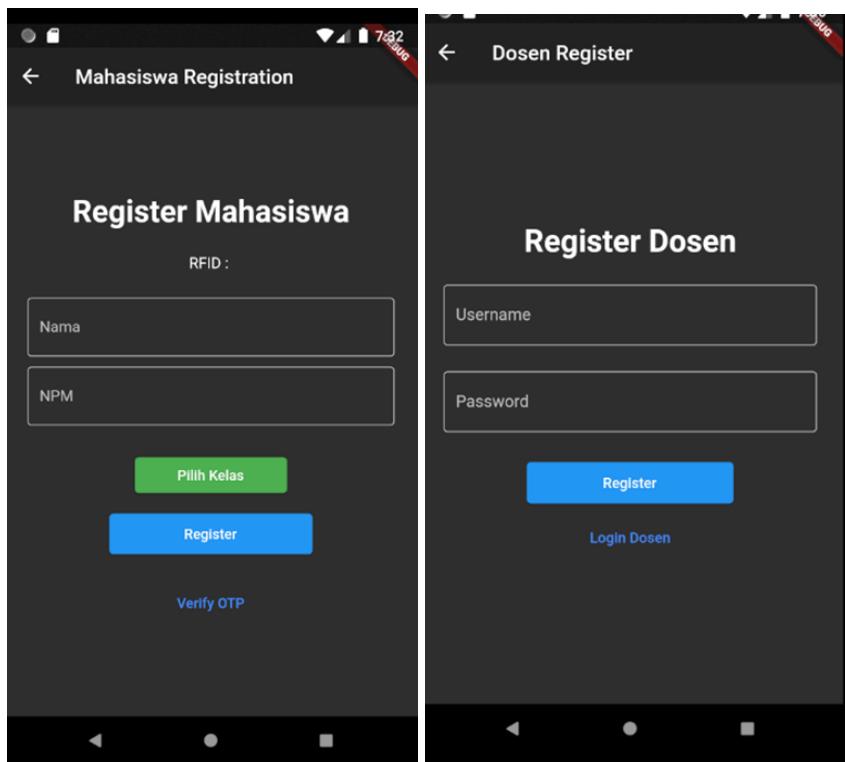
Gambar 7. Pengujian “Tap in” kartu mahasiswa

- Pengujian “Tap in” apabila mahasiswa tidak terdaftar pada kelas



Gambar 8. Hasil“Tap in” kartu mahasiswa

- Pengujian Register dan Login

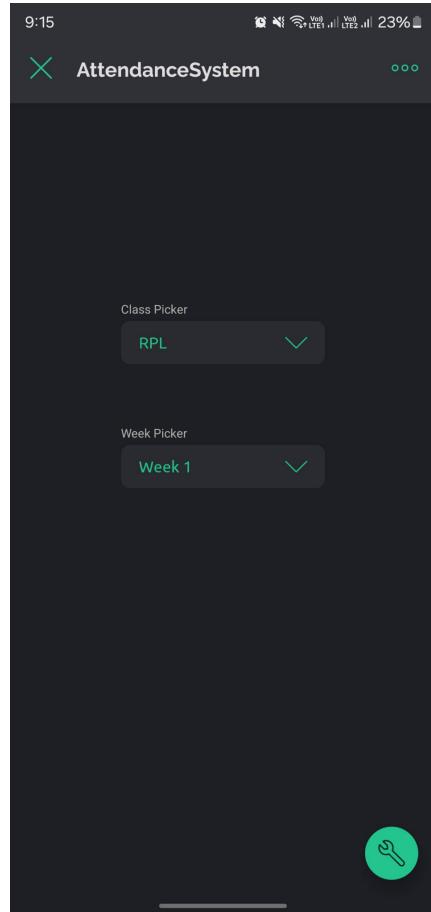


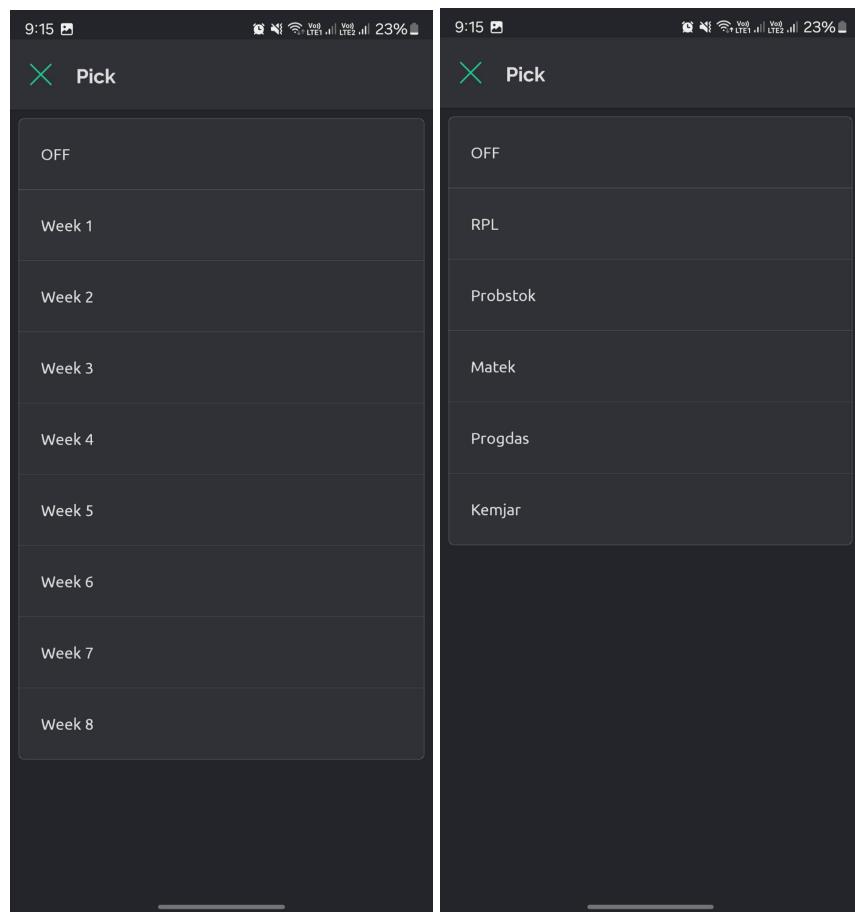
Gambar 9. Pengujian Register dan Login

### 3.1.2 Pengujian Interface Pengguna (UI/UX)

Pada tahapan ini, dilakukan pengujian terhadap *interface* aplikasi dan Blynk. Hal-hal yang diuji meliputi fungsionalitas antarmuka aplikasi (aplikasi yang dibuat dan aplikasi Blynk) dan kemampuan pengelolaan data kehadiran oleh pengguna. Fungsionalitas dan kemampuan dari aplikasi web yang dikembangkan dan aplikasi Blynk yang digunakan diuji, agar dapat bekerja dengan baik, sesuai dengan kriteria yang telah ditentukan.

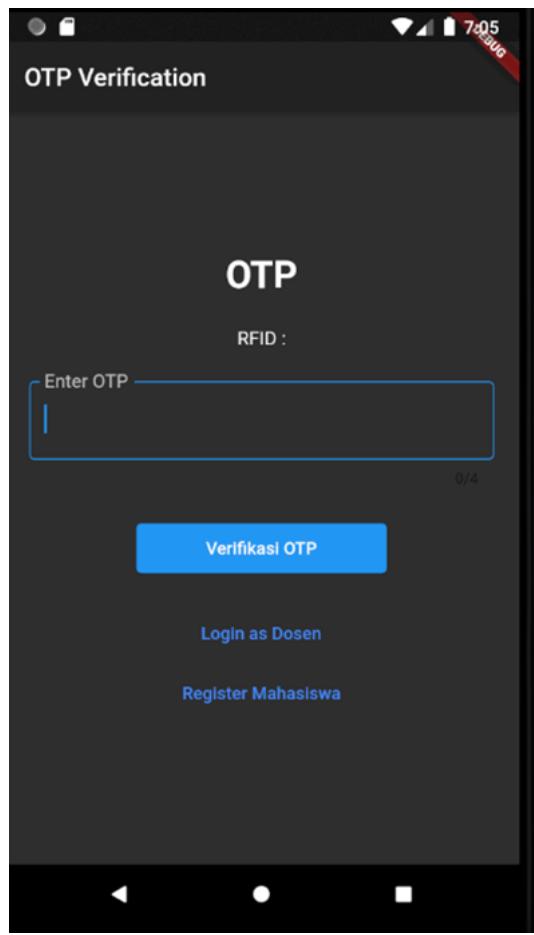
Berikut ini merupakan tampilan dari *interface* pada aplikasi Blynk:





Gambar 10. *Interface* aplikasi Blynk

Berikut ini merupakan tampilan dari interface pada aplikasi yang dikembangkan:



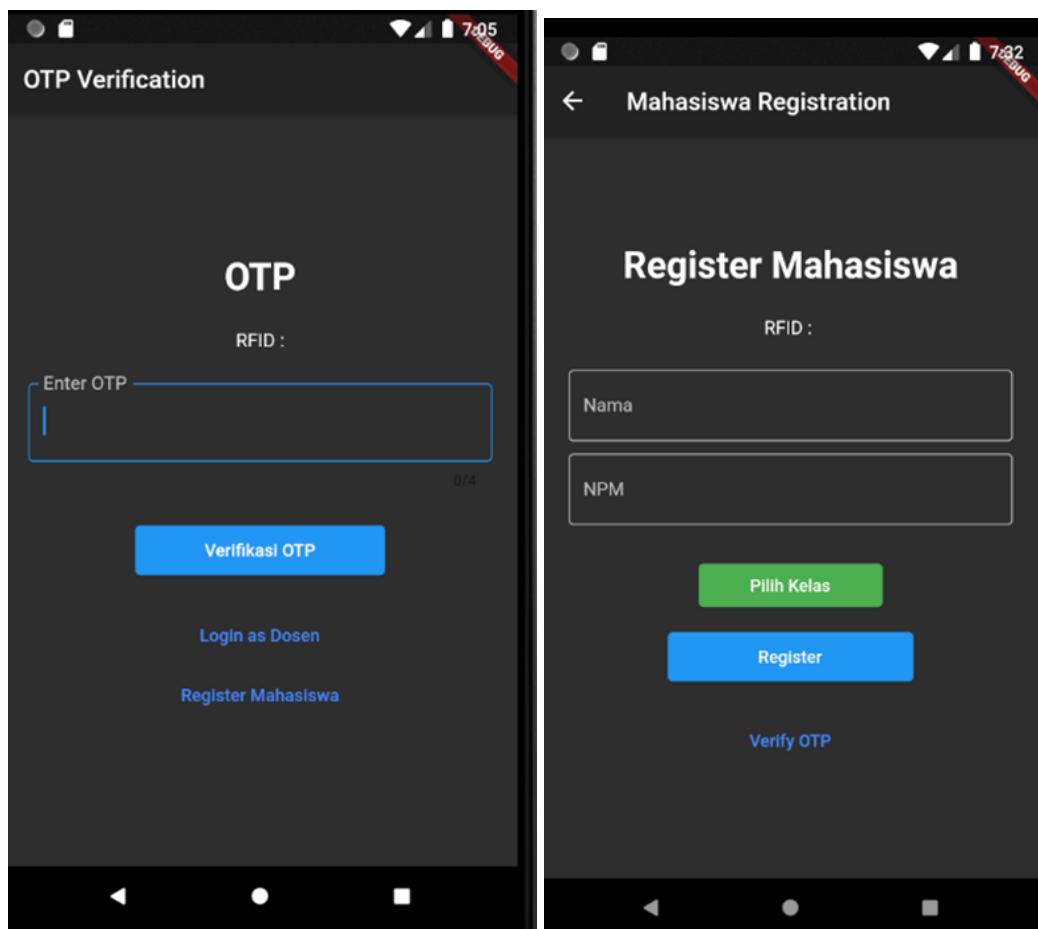
Gambar 11. *Interface* Aplikasi untuk OTP

Pada aplikasi Blynk, kami melakukan pengujian dengan menggantikan parameter yang terdapat dalam *interface* tersebut. Misal, kita menggantikan mata kuliah dengan minggu pertemuannya, maka hal ini akan mengatur pengiriman HTTP request yang dilakukan oleh ESP32 ke web server. Misalkan, pada aplikasi Blynk, kelas diatur menjadi Matematika Teknik maka ESP32 akan mulai mengirim data ke log absensi kelas Matematika Teknik pada database. Begitu pula untuk pengaturan pekan kuliah pada Blynk di mana setiap minggunya, administrator kelas dapat mengubah pekan kuliah melalui Blynk.

Pada aplikasi yang dikembangkan, kami melakukan pengujian terhadap login, register, dan melakukan login untuk dosen. Selain itu, aplikasi yang dikembangkan juga menampilkan bagian OTP, yang ditujukan untuk memverifikasi kehadiran dari mahasiswa. Hal ini dapat dilakukan dengan memasukkan kode OTP yang dikirimkan ke sistem absensinya, ke dalam bagian OTP pada aplikasi tersebut.

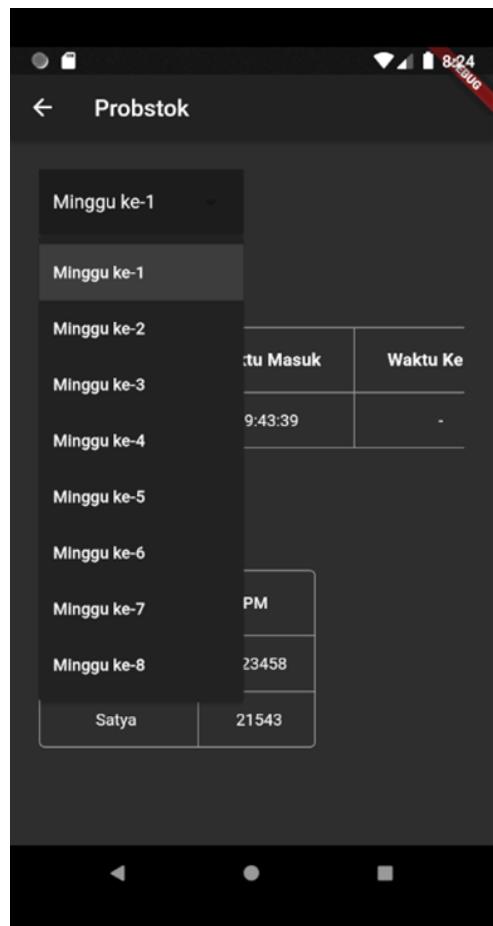
Dalam aplikasi yang dikembangkan ini, pengguna (dosen) dapat melakukan register dan login, tetapi berbeda dengan mahasiswa, yang tidak perlu untuk melakukan login karena

mahasiswa hanya perlu register nama dan NPM mereka agar tercantum ke dalam database aplikasinya, dan memasukkan kode OTP yang ditampilkan pada LCD, kemudian mahasiswa dapat melakukan pengisian kode OTP yang terpapar dalam tampilan aplikasi ini. Pengujian yang dilakukan terhadap aplikasi yang dikembangkan ini meliputi beberapa percobaan untuk memasukkan kode OTP ke dalam bagian “Enter OTP”, yang mana kami mencoba beberapa kemungkinan, seperti ketika memasukkan kode OTP yang tidak sesuai, dan ketika memasukkan kode OTP yang sesuai. Selain itu, pengujian ini juga dilakukan berdasarkan waktu, yang mana ketika penggunanya tidak memasukkan kode OTP dalam rentan waktu tertentu, sistem absensinya akan menjadi gagal. Berikut ini adalah contoh gambar pengujian yang dilakukan terhadap aplikasi yang dikembangkan.



Gambar 12. Pengujian Aplikasi untuk Menunjukkan OTP

Selain itu, dalam aplikasi ini, pihak administrator atau dosen dapat melihat data kehadiran mahasiswa, sehingga dosen dapat mengetahui kapan mahasiswa yang bersangkutan masuk dan keluar, serta siapa saja yang tidak hadir pada saat kelas berlangsung. Berikut ini merupakan tampilannya.



Gambar 13. *Interface* Data Kehadiran Mahasiswa

## 3.2 RESULT

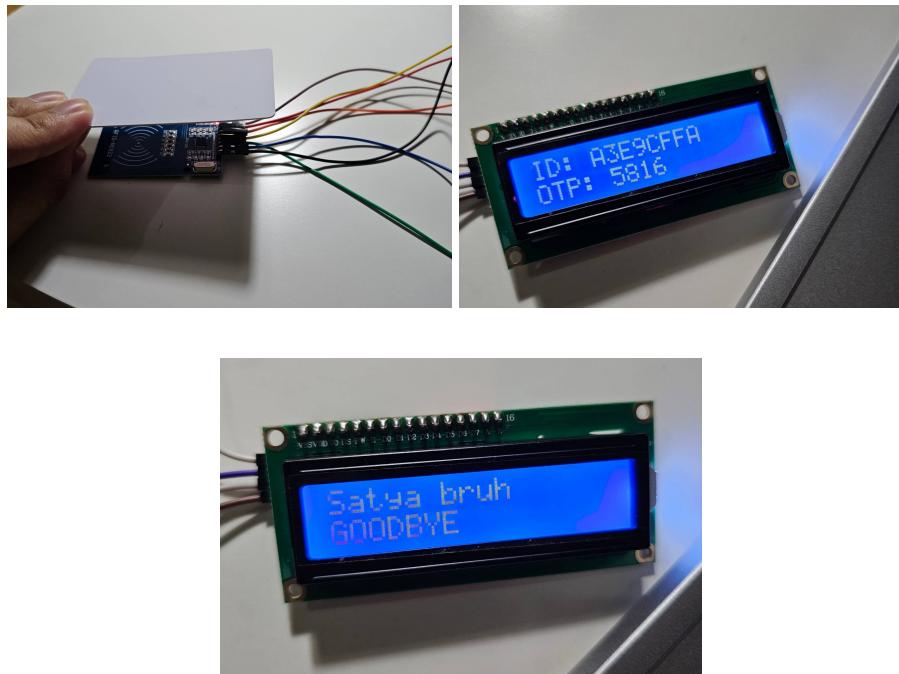
Hasil dari pengujian yang dilakukan menunjukkan bahwa kesuksesan pengembangan sistem absensi dengan menggunakan ESP32 dan RFID. Berikut ini adalah hasil yang diperoleh dari masing-masing pengujian yang dilakukan sebelumnya:

### 3.2.1 Hasil Pengujian Fungsional

Hasil pengujian fungsionalitas dari sistem absensi ini menunjukkan pencapaian yang signifikan dalam pengembangan sistem absensi berbasis ESP32 dan RFID. Berdasarkan serangkaian pengujian dan implementasi, hasil dari pengujian fungsionalitas dari keseluruhan sistem ini adalah sebagai berikut.

Pengujian fungsionalitas yang dilakukan pertama kali adalah dengan menjalankan program dan melakukan “Tap In” kartu, kemudian data mahasiswa akan muncul ke dalam LCD. Tampilan LCD akan menunjukkan nama mahasiswa dan kode

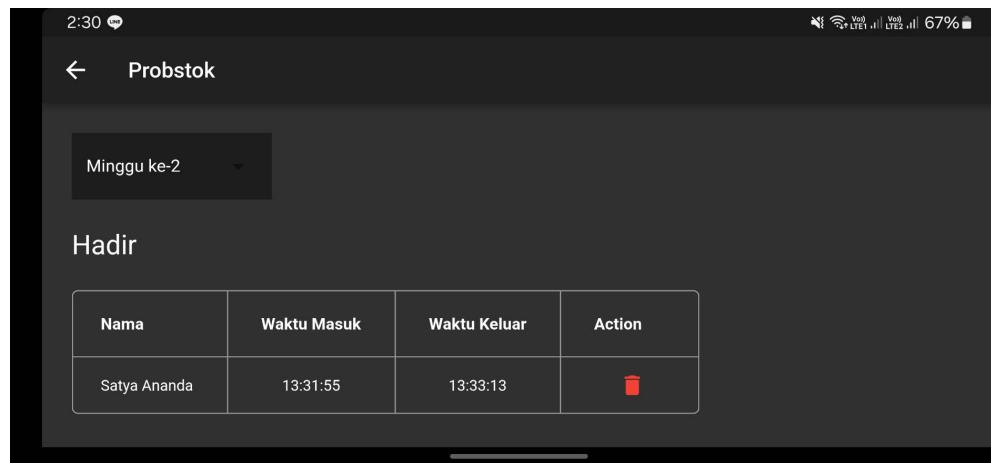
OTP, untuk dimasukkan ke dalam aplikasi yang dikembangkan. Selanjutnya, hasil data absen akan dicatat waktunya ke dalam aplikasi tersebut, sehingga pihak administrator dapat melihat waktu absen masuk dan keluar dari mahasiswa. Berikut ini adalah hasil pengujian fungsionalitas program pada LCD.



Gambar 14. Hasil Pengujian Fungsionalitas

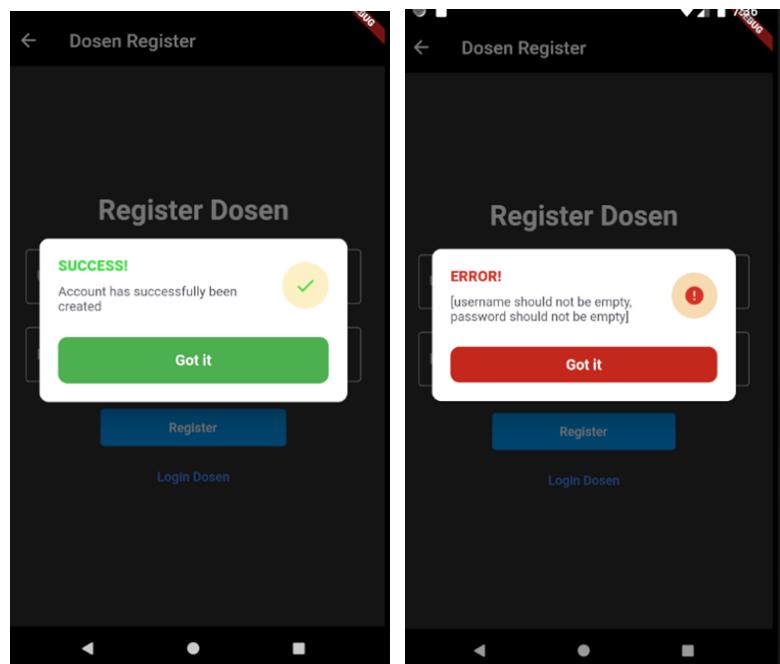
### 3.2.2 Hasil Pengujian Interface Pengguna (UI/UX)

Hasil dari pengujian *Interface* pengguna dalam aplikasi Blynk dan aplikasi OTP menunjukkan tampilan yang memudahkan pengguna dalam menggunakan aplikasinya. Pengujian yang dilakukan terhadap aplikasi Blynk menunjukkan bahwa pihak administrator dapat melakukan pengaturan kelas dan minggu pertemuan untuk menggunakan sistem absensi ini, sehingga kelas dan minggu pertemuannya dapat disesuaikan dengan baik. Selain itu, hasil absensi juga memiliki tingkat keakuratan yang tinggi, karena sesuai dengan data yang tercantum dalam *database*. Berikut ini adalah gambaran hasil pengujian terhadap aplikasi Blynk.

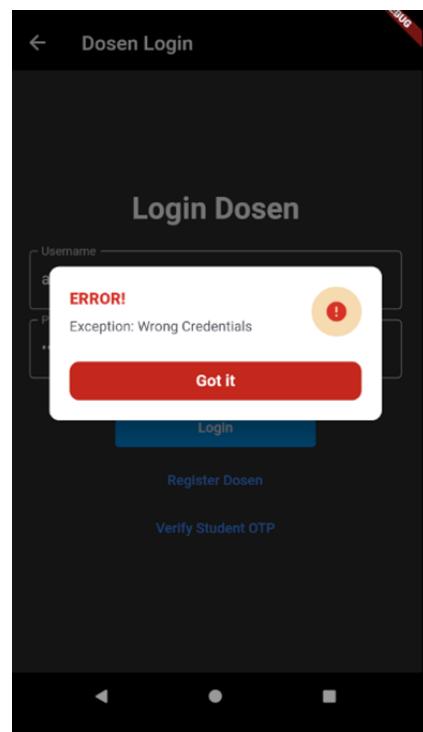
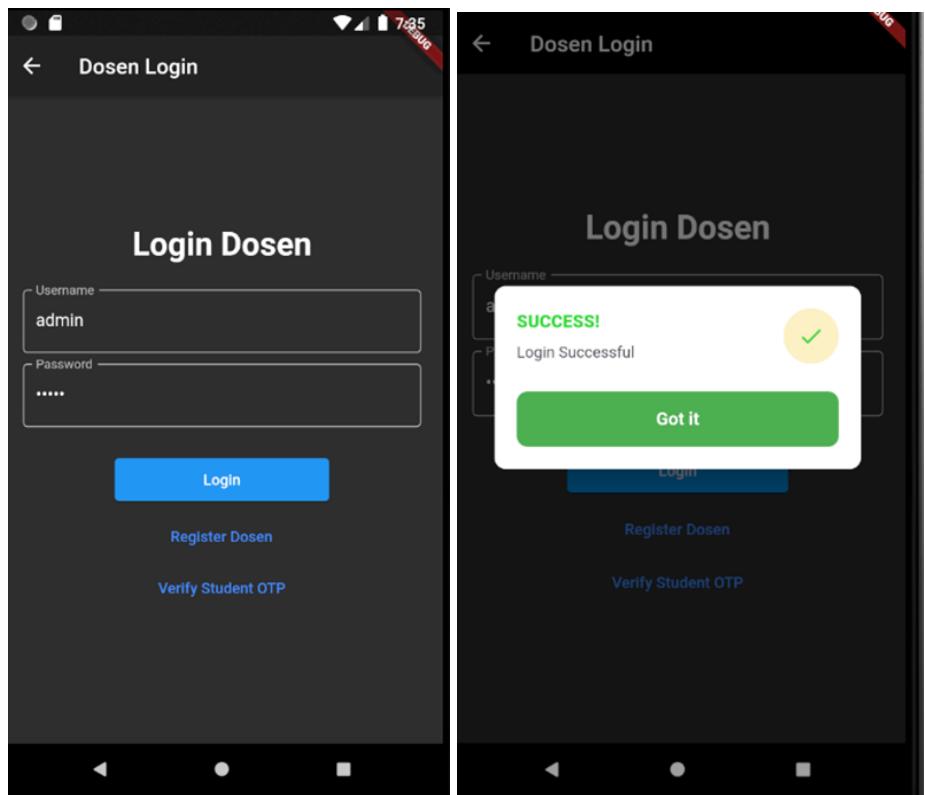


Gambar 15. Hasil Absensi pada Aplikasi

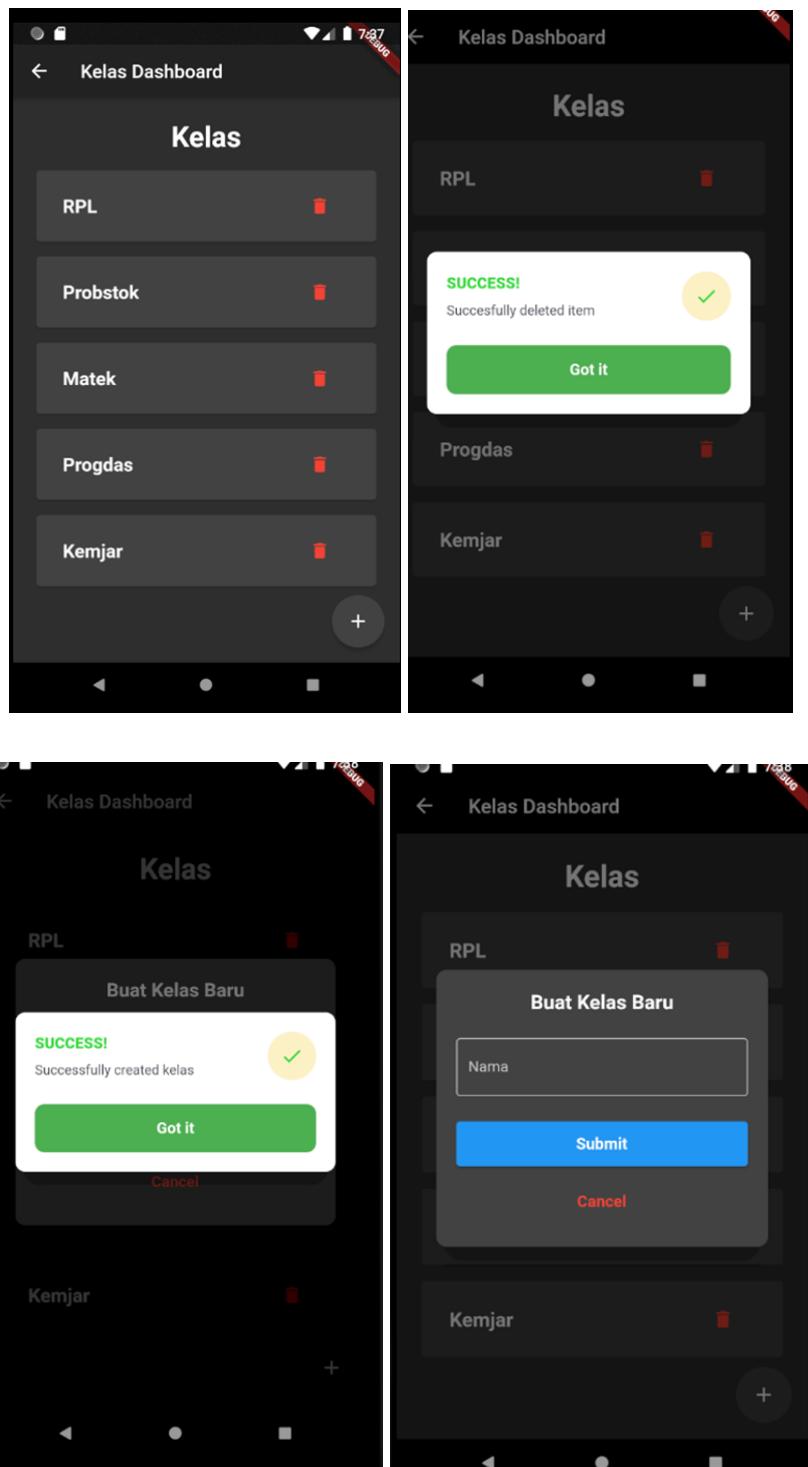
Pengujian yang dilakukan untuk login sebagai dosen dilakukan untuk mengetahui keberhasilan dari aplikasi yang dikembangkan. Dalam hal ini, pengujian membawakan hasil, yang mana ketika pihak dosen melakukan login, mereka dapat mengatur kelas, sehingga dosen dapat menambahkan kelas, mengurangi kelas, serta melihat tabel data kehadiran mahasiswa seperti pada gambar di atas. Pengujian yang dilakukan terhadap fitur ini juga dilakukan dengan menerapkan “*error handling*”. Berikut ini merupakan hasil dari pengujian fitur untuk dosen pada aplikasi yang kami kembangkan.



Gambar 16. Register Dosen



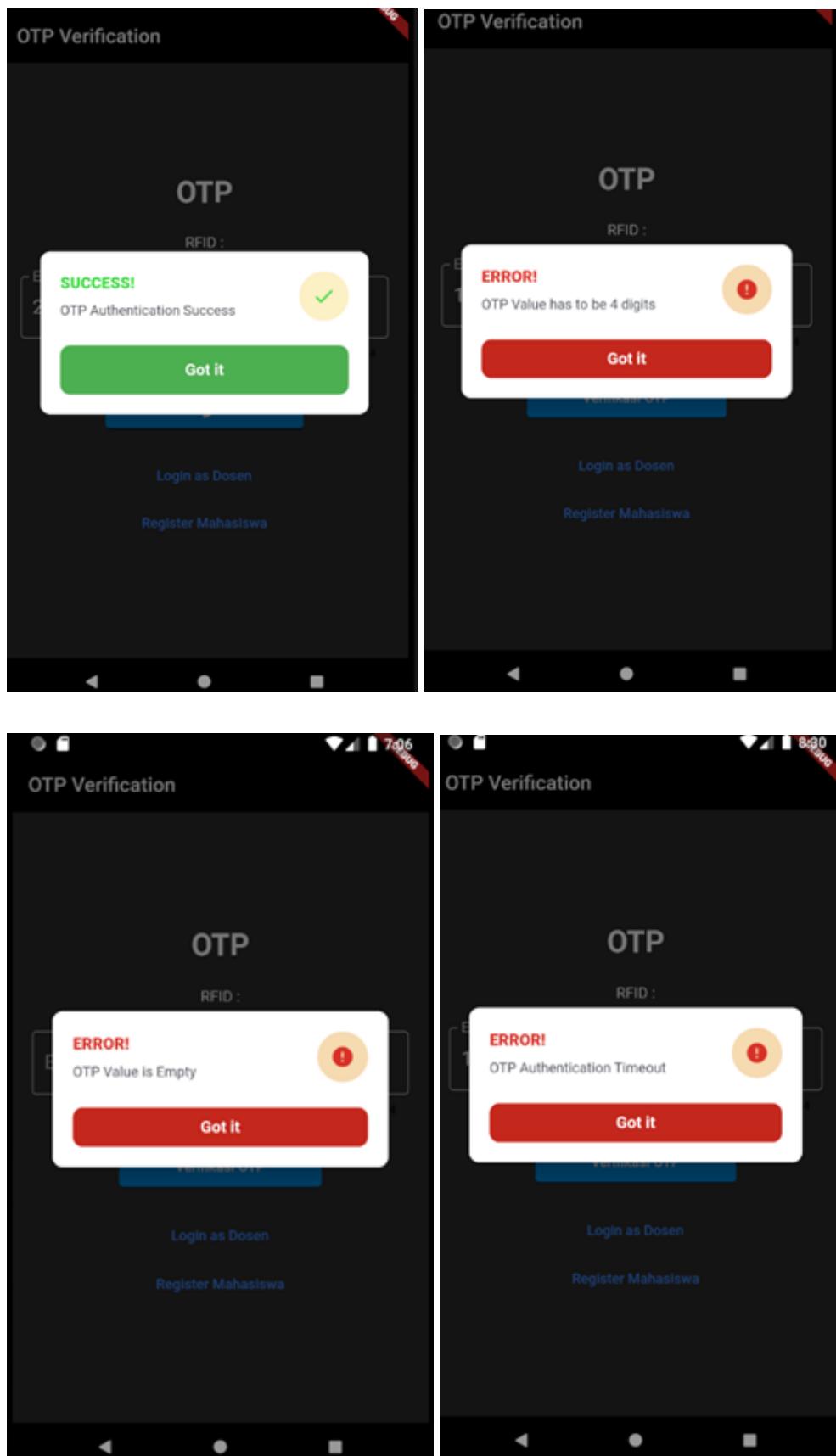
Gambar 17. Login Dosen



Gambar 18. *Dashboard Kelas*

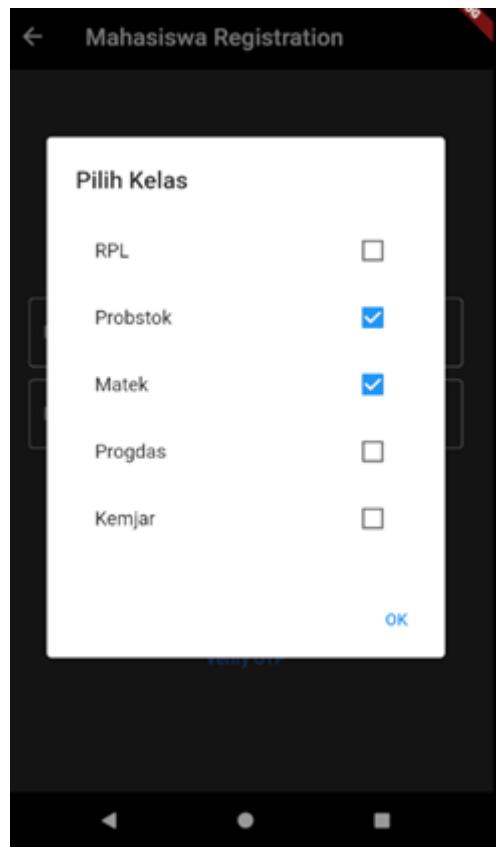
Selanjutnya, hasil dari pengujian terhadap aplikasi OTP yang dikembangkan, menunjukkan bahwa OTP dapat dijalankan dengan benar, sehingga penggunanya tidak dapat memasukkan angka yang tidak sesuai dengan kode OTP yang dihasilkan oleh sistem absensi ini. Selain itu, terdapat fitur rentang waktu, yang mana akan

muncul pesan *error*, sehingga pengguna tidak dapat memasukkan kode OTP ketika sudah melewati batas waktu yang ditetapkan. Berikut ini adalah hasil pengujianya.



Gambar 19. Hasil pengujian OTP pada Aplikasi

Hasil pengujian pemilihan kelas untuk *role* mahasiswa sudah sukses dilakukan, hal ini menunjukkan bahwa mahasiswa dapat melakukan absennya sesuai dengan kelas dan waktu kelas yang sedang diikutinya. Berikut ini adalah bukti dari hasil pengujian yang dilakukan.



Gambar 20. Hasil pengujian Pemilihan Kelas

Selain itu, pengujian yang dilakukan terhadap aplikasi yang dikembangkan juga membawakan hasil, di mana pengguna (dosen) dapat melihat tampilan data kehadiran mahasiswa, yang membuatnya dapat mengetahui siapa saja yang hadir dan waktu kehadiran dari masing-masing mahasiswa yang tersebut. Berikut ini adalah gambar hasil pengujianya.



Gambar 21. Hasil Pengujian Data Kehadiran Mahasiswa

Dari hasil-hasil yang diperoleh ini, dapat dikatakan bahwa sistem dapat mengidentifikasi pengguna dengan akurat, menggunakan *tag* RFID yang telah terdaftar dalam *database*-nya. Selanjutnya, konfirmasi kehadiran dari mahasiswa dapat dilihat dari LCD yang tertera pada gambar di atas. Selain itu, aplikasi Blynk yang digunakan dapat melakukan perubahan atas minggu pertemuan dan kelas yang terdapat pada *interface*-nya. Kemudian, integrasi komunikasi antara perangkat lunak dan perangkat keras dapat dilakukan secara mulus, sehingga komunikasinya dapat diandalkan. Kinerja sistem dalam hal kecepatan respons terhadap identifikasi kartu, pencatatan kehadiran, dan respons *interface* pengguna dinilai sudah memuaskan dan dapat diterima oleh penggunanya.

Hasil-hasil ini membuktikan bahwa proyek ini sudah memenuhi segala kriteria yang diperlukan untuk mencapai tujuan utamanya, yaitu untuk meningkatkan efisiensi dan akurasi proses absensi di lingkungan perkuliahan. Oleh karena itu, proyek ini dapat berjalan dengan baik, dan sesuai dengan kebutuhan dan ketentuannya.

### **3.3 EVALUATION**

Tahapan evaluasi merupakan tahapan yang sangat penting dalam penilaian keberhasilan, kekurangan, dan potensi perbaikan sistem yang telah dikembangkan. Evaluasi ini dilakukan dengan melihat keberhasilan dari tujuan utama proyek, yang mana menunjukkan bahwa fungsionalitas keseluruhan dari sistem ini dapat berjalan dengan baik. Selanjutnya, evaluasi dilakukan dengan mereferensikan *acceptance criteria* dari proyek ini, yang mana juga sudah terpenuhi seluruhnya, walaupun mungkin terdapat beberapa bagian yang memerlukan untuk dikembangkan secara lebih lanjut.

Dari tahapan evaluasi ini, kami menilai bahwa terdapat kesulitan dan tantangan yang dihadapi dalam proses pengembangannya. Contohnya, pada proses integrasi antara perangkat keras dan perangkat lunak yang memerlukan waktu cukup lama untuk diselesaikan. Dengan ini, diperlukan untuk meningkatkan manajemen waktu dan proses integrasinya. Berikutnya, meskipun keamanan sistem dapat diimplementasikan dengan baik, tetapi akan lebih baik lagi untuk memberikan lapisan keamanan yang lebih, terutama pada bagian otentikasi dan enkripsi data, agar sistem absensi ini dapat digunakan dengan aman.

Proyek ini dapat ditingkatkan lebih lanjut mengenai fungsionalitas sistem dengan menambahkan fitur-fitur tambahan, seperti pelaporan kehadiran yang lebih rinci, pemberitahuan real-time, atau integrasi dengan sistem manajemen lainnya. Evaluasi ini memberikan wawasan tentang bagian mana dari sistem yang memerlukan optimalisasi, termasuk respons sistem yang lebih cepat dan penanganan kesalahan yang lebih baik, sehingga sistem dapat berjalan dengan optimal.

## **CHAPTER 4**

### **CONCLUSION**

Proyek yang bertujuan untuk membantu proses absensi ini berhasil diimplementasikan dengan menggunakan ESP32 dan RFID. Keberhasilan ini diakibatkan oleh pengembangan perangkat keras dan perangkat lunak yang terintegrasi, sehingga sistem absensi ini dapat mencapai tujuan utamanya yaitu untuk meningkatkan efisiensi dan akurasi proses absensi di lingkungan perkuliahan.

Implementasi *software* dan *hardware* berhasil diintegrasikan, yang membuatnya dapat melakukan proses absensi menjadi lebih efisien, mengurangi waktu yang dibutuhkan untuk pencatatan kehadiran, dan menghindari kesalahan manual. Selain itu, sistem ini sudah dapat mengidentifikasi pengguna dengan akurat melalui teknologi RFID, serta memberikan konfirmasi yang jelas atas kehadiran atau ketidakhadiran, dengan menambahkan OTP untuk memverifikasi kehadiran tersebut. melalui *interface* web atau mobile, dosen atau pengelola dapat dengan mudah mengakses dan mengelola data kehadiran karyawan atau siswa.

Salah satu tantangan utama yang berhasil diatasi adalah pada tahap integrasi antara perangkat ESP32 dan perangkat lunaknya. Hal ini dilakukan dengan serangkaian uji coba dan penyesuaian, sehingga program dapat berjalan sesuai dengan tujuannya. Selain itu, mengelola koneksi WiFi, komunikasi dengan server, kemudian memberikan pesan ke LCD juga merupakan tantangan lainnya yang dalam proses pengembangan proyek ini, yang berhasil diatasi.

Hasil pengujian membuktikan keberhasilan program ini dalam mencapai tujuannya. Dengan ini, dapat dikatakan bahwa program telah berjalan dengan baik dan sesuai dengan kriteria yang diperlukan untuk proyek ini. Harapan kami, proyek ini dapat bermanfaat bagi sistem kehadiran pada lingkungan perkuliahan.

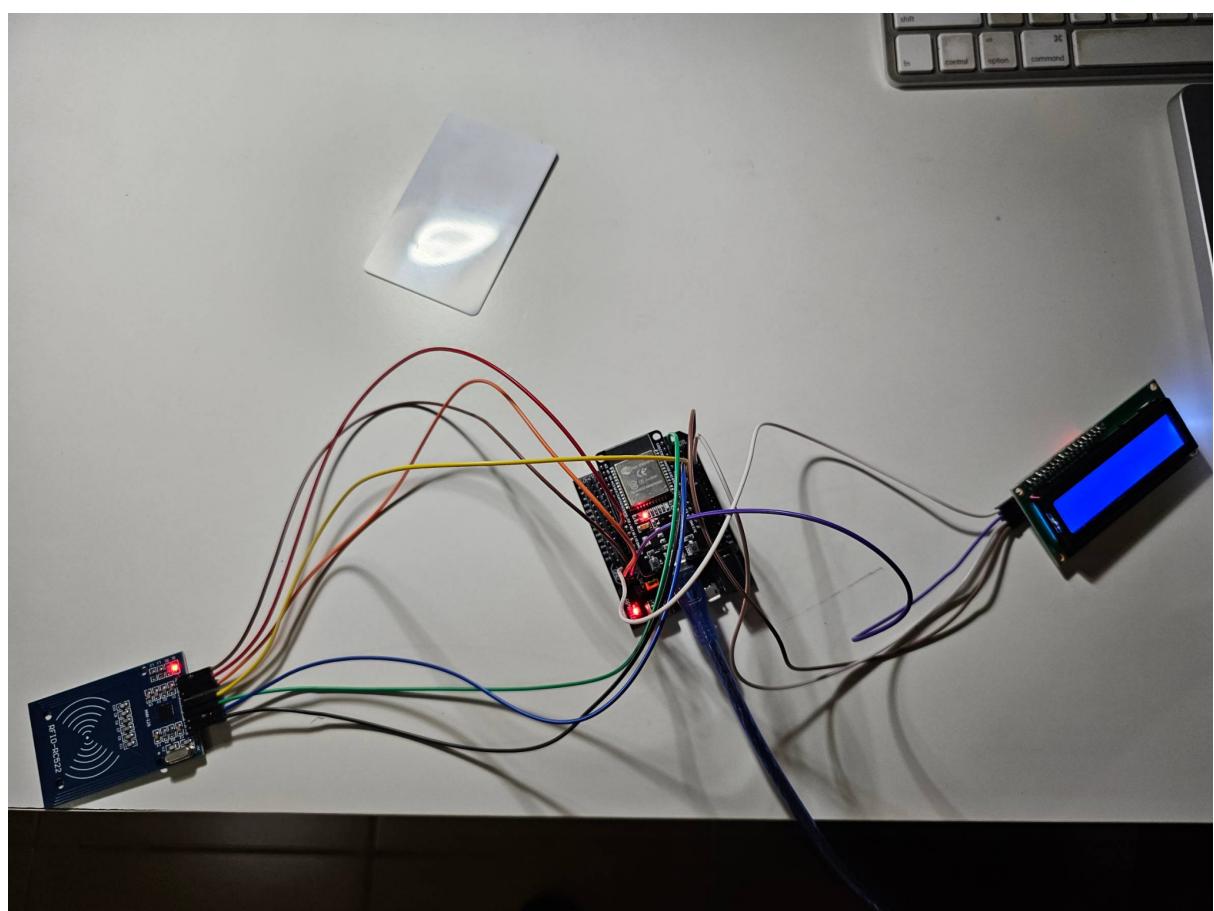
## REFERENCES

- [1] “Documentation | NestJS - A progressive Node.js framework,” *Documentation | NestJS - a Progressive Node.js Framework*. <https://docs.nestjs.com/> [Accessed: Des 5, 2023]
- [2] “Flutter documentation,” *Flutter*. <https://docs.flutter.dev/> [Accessed: Des 5, 2023]
- [3] R. Santos and R. Santos, “ESP32 HTTP GET and HTTP POST with Arduino IDE | Random Nerd Tutorials,” *Random Nerd Tutorials*, Oct. 27, 2022. <https://randomnerdtutorials.com/esp32-http-get-post-arduino/> [Accessed: Des 5, 2023]
- [4] Admin, “IoT Based RFID Attendance System using ESP32,” *IoT Projects Ideas*, Sep. 15, 2022. <https://iotprojectsideas.com/iot-rfid-attendance-system-esp32/> [Accessed: Des 5, 2023]
- [5] BaldGuyDIY, “ESP32: How to read RFID tags with a RFID Reader,” *YouTube*. Jun. 05, 2021. [Online]. Available: <https://www.youtube.com/watch?v=pJLjFm4Ipro> [Accessed: Des 5, 2023]

## APPENDICES

### Appendix A: Project Schematic

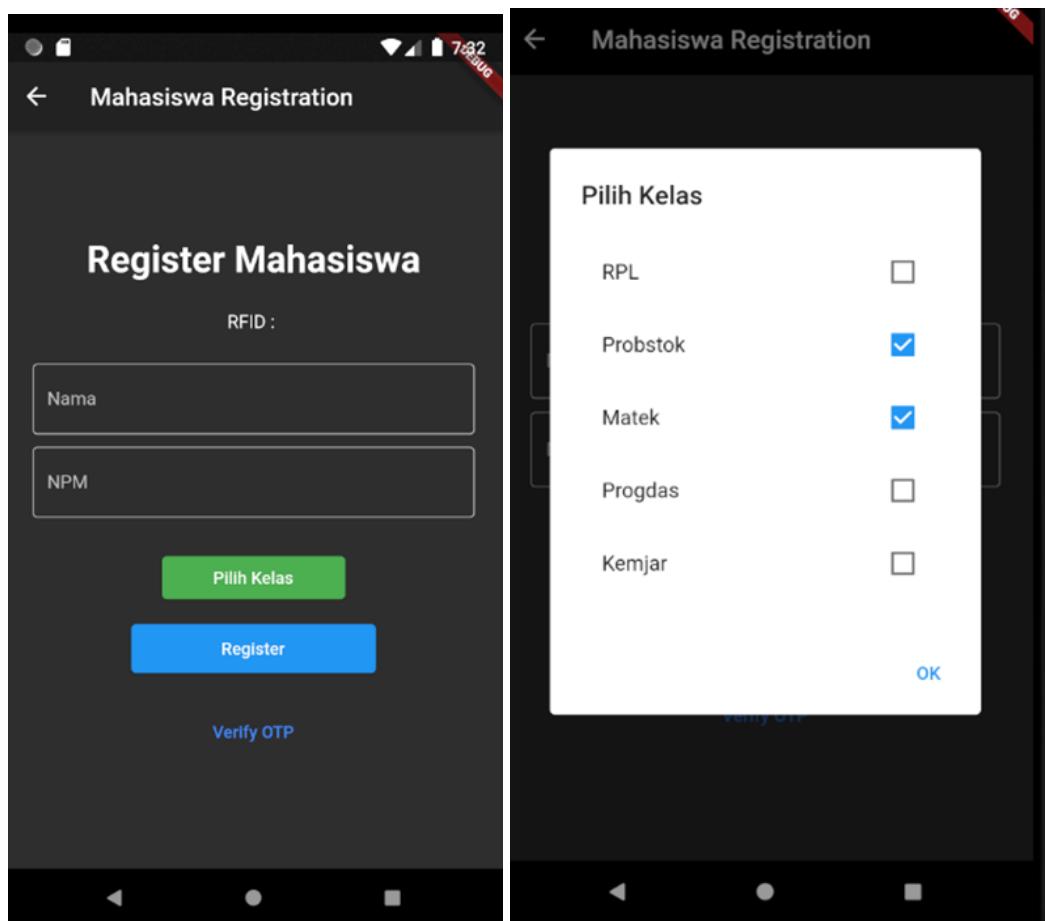
Berikut ini adalah gambar rangkaian dari proyek sistem absensi ini:



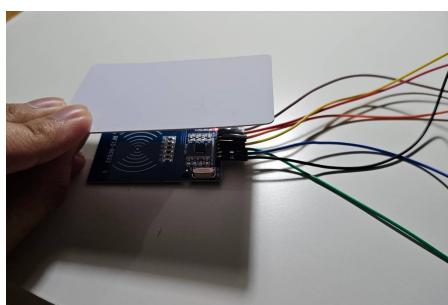
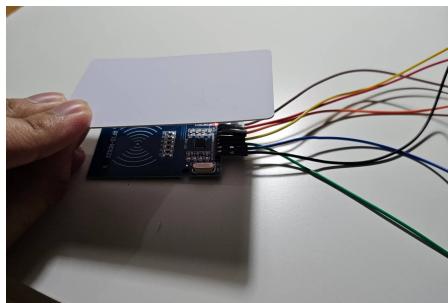
### Appendix B: Documentation

Berikut ini adalah dokumentasi terhadap sistem absensi yang dibuat:

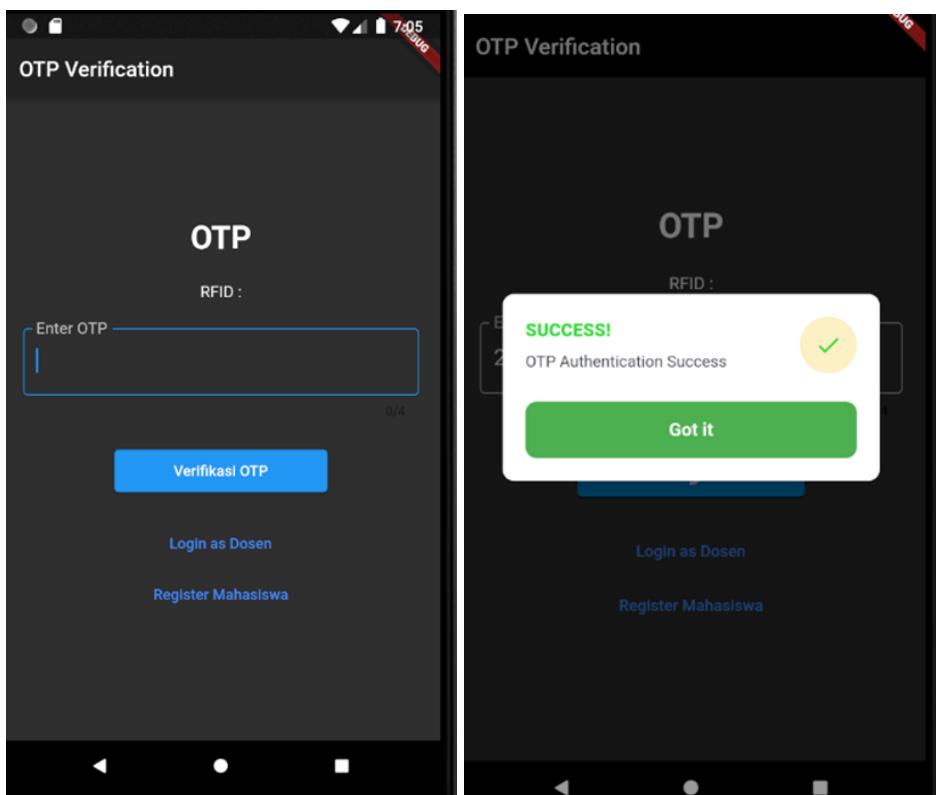
- Melakukan Register dan Memilih Kelas Mahasiswa



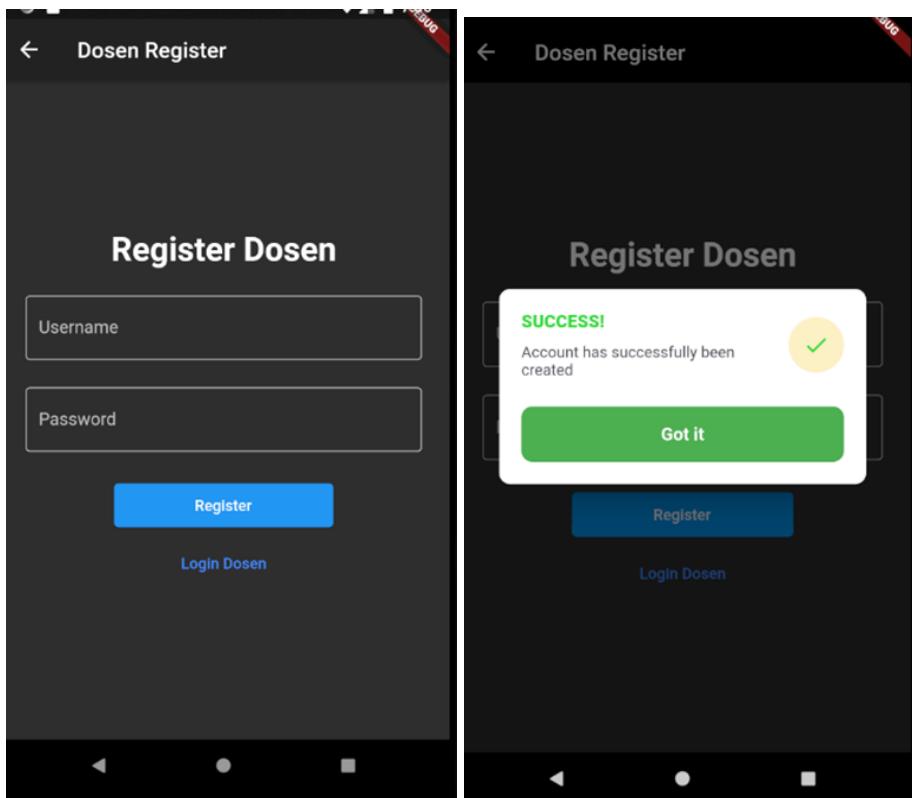
- Melakukan Tap in dan Out

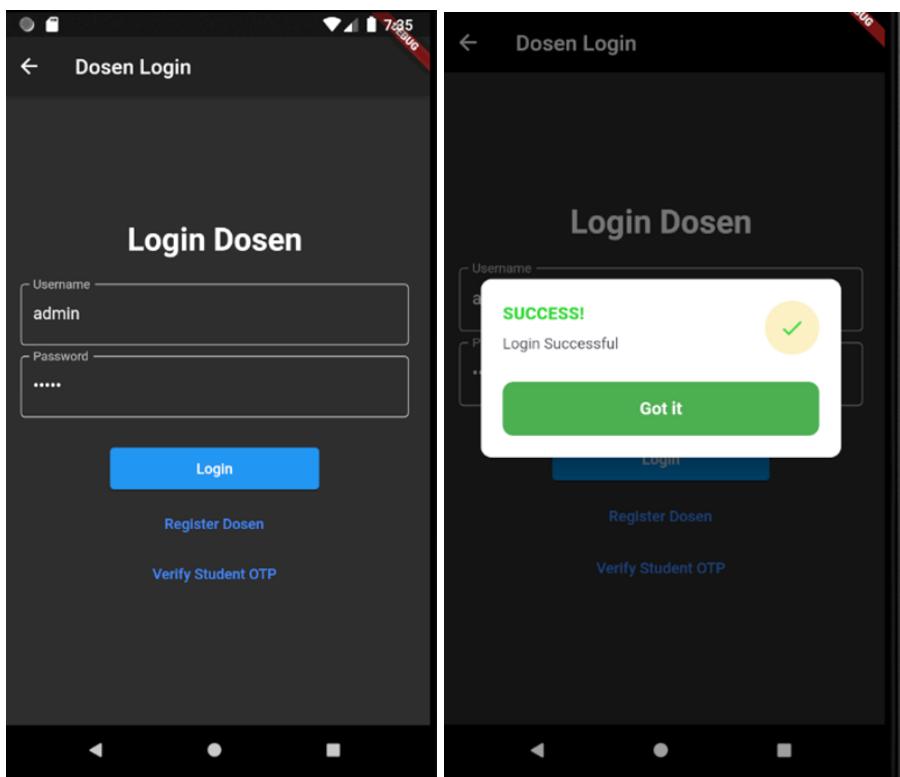


- Memasukkan kode OTP



- Melakukan Register Dosen dan Login

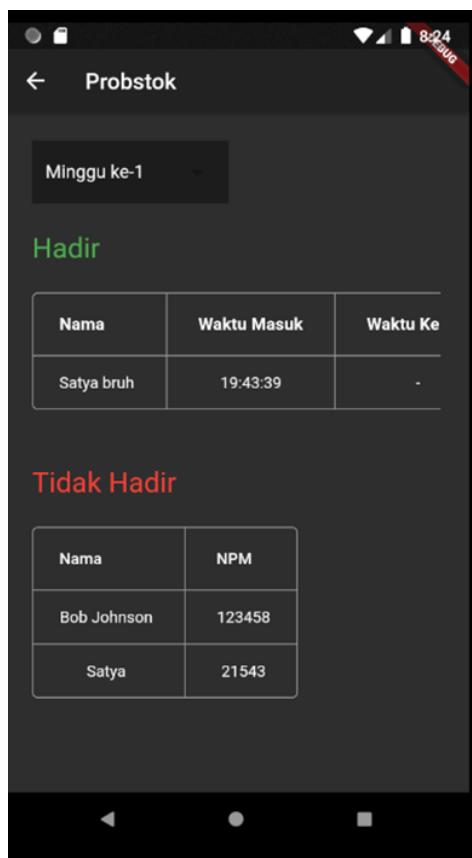




- Mengganti pertemuan mata kuliah

Minggu ke-1	Waktu Masuk	Waktu Ke
Minggu ke-1		
Minggu ke-2		
Minggu ke-3	9:43:39	-
Minggu ke-4		
Minggu ke-5		
Minggu ke-6		
Minggu ke-7	PM	
Minggu ke-8	23458	
Satya	21543	

- Melihat hasil waktu absensi



- Membuat kelas baru dan Menghapus Kelas

